electronics

MDPI

Article

THANet: Transferring Human Pose Estimation to Animal Pose Estimation

Jincheng Liao [1], Jianzhong Xu [2,*], Yunhang Shen [3] and Shaohui Lin [1,4,*]

[1] School of Computer Science and Technology, East China Normal University, Shanghai 200062, China; jcl.ecnu23@gmail.com
[2] Training and Simulation Center, Army Infantry Academy, Nanchang 330103, China
[3] Youtu Lab, Tencent, Shanghai 200123, China; shenyunhang01@gmail.com
[4] KLATASDS-MOE, Shanghai 200062, China
* Correspondence: xukezheng1978@163.com (J.X.); shlin@cs.ecnu.edu.cn (S.L.)

Abstract: Animal pose estimation (APE) boosts the understanding of animal behaviors. Recent vision-based APE has attracted extensive attention due to the advantages of contactless and sensorless applications. One of the main challenges in APE is the lack of high-quality keypoint annotations for different animal species since manually annotating the animal keypoints is very expensive and time-consuming. Existing works alleviate this problem by synthesizing APE data and generating pseudo-labels for unlabeled animal images. However, feature representations learned from synthetic images could not be directly transferred to real-world scenarios, and the generated pseudo-labels are usually noisy, which limits the model's performance. To address the above challenge, we propose a novel cross-domain vision transformer for APE to **T**ransfer **H**uman pose estimation to **A**nimal pose estimation, termed THANet, as humans share skeleton similarities with some animals. Inspired by the success of ViTPose in HPE, we design a unified vision transformer encoder to extract universal features for both animals and humans followed by two task-specific decoders. We further introduce a simple but effective cross-domain discriminator to bridge the domain gaps between the human pose and the animal pose. We evaluated the proposed THANet on the AP-10K and Animal-Pose benchmarks, and the extensive experiments show that our method achieves a promising performance. Specifically, the proposed vision transformer and cross-domain method significantly improve the model's accuracy and generalization ability for APE.

Keywords: animal pose estimation; cross-domain; vision transformer

check for updates

Citation: Liao, J.; Xu, J.; Shen, Y.; Lin S. THANet: Transferring Human Pose Estimation to Animal Pose Estimation. *Electronics* **2023**, *12*, 4210. https://doi.org/10.3390/electronics12204210

Academic Editor: Dah-Jye Lee

Received: 27 August 2023
Revised: 2 October 2023
Accepted: 4 October 2023
Published: 11 October 2023

Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Animal pose estimation (APE) aims to predict the keypoint position of animals' essential parts from images or videos. APE has received increasing attention from researchers for consecutive years. Parsing animal posture promotes the understanding of animal behaviors, which is the foundation of other disciplines such as biomechanics, neuroscience, and behavior.

The traditional APE methods make physical markings on the animal body. For instance, researchers place visible markers such as small light bulbs or chemical light-emitting agents on the animal body, which capture the behavior information of the animals [1,2]. However, one main issue in traditional APE methods is that the utilization of markers may frighten the animals and affect their behavior, which limits the generalization of these methods. Therefore, due to its contactless and sensorless qualities, the vision-based APE methods have significantly captured the attention of researchers. In recent years, it has achieved remarkable progress, owing to the fast-paced development of various deep learning models and the improvement of computing resources.

The existing works of vision-based APE are classified into two main categories. The first focuses on directly transferring the exiting human pose estimation (HPE) methods to

animals [3]. The performance of this kind of APE method is constrained by the difference in skeletal structure between humans and animals, so the results of APE are hardly comparable to the accuracy achieved by HPE. The second group of methods uses human data or synthetic animal data to learn the APE models [4–6]. In these methods, initial and weak models are built to generate the pseudo-labels for unlabeled images. Then, the synthetic animal data and the real animal images are incorporated into the training stage to learn the APE model. This kind of method is also called cross-domain APE.

Although these existing works have obtained significant results, there are still some limitations. First, unlike HPE, which only focuses on the single skeletal structure of humans, in APE, the skeletal structures of animals are diversified. For example, animals, such as *fish*, *birds*, *snakes*, and *dogs*, have completely different skeletal structures. It makes the APE task more challenging than HPE. It is also difficult to have a general pose estimation model that simultaneously solves the problems of pose estimation with these different structures. Second, the diversification of animals also makes the annotation task of pose estimation onerous. It requires annotating a sufficient amount of data for each animal species to train an effective APE model. How to improve the generalizability of the APE model for different animals is an urgent problem. Finally, the challenges of APE are also brought by the diversification of animal behaviors, as the location of joints is dramatically varied when the animals perform their behavior. Enhancing the model generalizability for different postures is also a challenge that this paper focuses on.

CNN has become the dominant method for vision tasks over the span of the last ten years [7,8]. However, with the advent of efficient structures and convergence of computer vision and natural language processing, the transformer-based neural network has become a new research direction for vision tasks to achieve superior performance [9,10]. Recently, ViTPose [11] uses non-hierarchical and plain ViT as backbones to extract represented features for regions containing person instances and an efficient decoder for HPE. In this paper, we find that there are some structural similarities between the human body and certain animals (such as *monkeys*), but humans and animals belong to different species with a domain gap. Therefore, we propose THANet to simultaneously estimate human and animal poses and learn universal features, thus achieving excellent pose estimation and species generalizability.

Conclusively, the main contributions of our work are as follows:

- We design a simple yet efficient multi-task encode–decoder transformer network, namely THANet, which simultaneously estimates human pose and animal pose.
- We propose a joint-learning strategy and cross-domain adaptation to transfer knowledge from the human domain to the animal domain to improve performance and generalization.
- The proposed THANet achieves promising results on the AP-10K and the Animal-Pose benchmarks, verifying the effectiveness of our proposed approach.

2. Related Work

In this section, we summarize the representative progress in human pose estimation and animal pose estimation in Table 1. Section 2.1 provides a detailed introduction to the previous work of human pose estimation, while Section 2.2 delves into animal pose estimation.

2.1. Human Pose Estimation

Human pose estimation (HPE) is a core task of predicting and tracking the locations of a person in computer vision, which is used in many applications, such as sports analysis, health care, and game production. One of the classical HPE approaches is the pictorial structure model [12–14], which models the spatial relationship among parts of the body as a tree-structured graph. However, this approach fails to capture correlations between invisible and deformable body parts. Recently, deep learning-based methods have achieved great performance in improving single-person pose estimation [15–18]. For multi-person estimation, the existing approaches are grouped into top–down [19–21] and bottom–up

methods [22–24]. The first category finds human regions by the object detectors and estimates skeletal joints. The second category estimates all skeletal joints directly and then clusters these joints to generate a distinctive pose based on the skeleton.

In the past few years, CNN-based models have improved in performance, but result in insufficient data with regard to the HPE [18,25,26]. Most of the models are variants of CNNs and continuously pursue high performance. Recently, Xu et al. [11] applied the vision transformer to HPE, which demonstrates its effectiveness in keypoint detection.

There are also some related tasks that are close to HPE, such as pose localization [27], head pose estimation [28], and head movement [29]. Their excellent works and HPE contribute to a better understanding and analysis of human behavior.

Table 1. The representative progress in human pose estimation and animal pose estimation within the last decade.

Method (Dataset)	Keywords	Year	Description
Animal Pose Estimation			
TigDog [30]	2D joint, horse, tiger	2015	A keypoint dataset for horse and tiger.
WS-CDA [6]	2D joint, weakly semi supervised, cross-domain	2019	Transfer knowledge from unlabeled animal data and provide a new Animal-Pose dataset.
CC-SSL [4]	Synthetic data, semi-supervised learning	2020	Consistency-constrained SSL framework for APE.
UDA [5]	Unsupervised learning, self-distillation	2021	Unsupervised learning pipeline for APE and online pseudo-label refinement strategy.
AP-10K [31]	2D joint, mammal	2021	Large-scale mammal animal pose dataset.
Human Pose Estimation			
Deeppose [15]	Single-person, regression	2014	The first HPE method via deep neural network.
CPM [17]	Top–down, sequential	2016	Sequential CNNs to learn spatial features in the end-to-end way.
SB [25]	Efficient, tracking	2018	Efficient pose estimation via CNN.
HRNet [26]	high-resolution representation	2019	Learning high-resolution representation from images.
DEKR [24]	Bottom–up, regression	2021	Directly regressing the keypoint position with bottom–up paradigm.
ViTPose [11]	Vision-transformer	2022	Effective pose estimation via vision transformer.

2.2. Animal Pose Estimation

Animal pose estimation (APE) is similar to HPE but comes with challenges such as biodiversity and data sparsity. Therefore, most existing datasets contain limited species because it is hard to define a unified pose. For instance, when dealing with non-quadruped animals such as fish and insects, their body structures and modes of locomotion differ significantly from those of humans and common quadruped animals. Consequently, defining their poses and collecting images of rare animals are both challenging. Additionally, capturing high-resolution images of animals is difficult due to their concealed habitat and quick movement. Some work from the literature collected and annotated additional data [31,32]. Yu et al. [31] built the first large-scale benchmark for mammal APE, i.e., AP-10K. In addition, there are other real animal datasets, including TigDog [30] and Animal-Pose [6], which all contain *horses*, *tigers*, and other quadrupeds. In terms of limited labeled animal data, Mu et al. [4] proposed a synthetic dataset by using CAD animal models and used the generated synthetic animal images to train the APE models.

Benefiting from the appearance of large-scale synthetic animal data and great success in the HPE task, transfer learning is applied to APE [5,6,33]. Most transfer-based works

are combined with the domain adaptive method, which learns the representation from other domains that contain rich data, such as synthetic animal data and human data, to boost the behavior of APE. Cao et al. [6] designed a weakly and semi-supervised APE framework to extract the features between humans and quadrupeds since humans are mammals and share a remarkably similar skeletal structure with the majority of quadruped animals. Mu et al. [33] synthesized APE data with more than ten animals to train the initial model and produce pseudo-labels for unlabeled real images, which are used to refine the initial model. For the initial models trained on synthetic datasets, the generated pseudo-labels for unlabeled images are usually noisy. To solve the problem of noisy pseudo-labels, Mu et al. [33] designed three consistency-check rules to discover noisy pseudo-labels. Li et al. [5] trained the APE model on a synthetic animal dataset with an unsupervised domain adaptation. In contrast to the method in [33], they built a teacher–student network and updated the pseudo-label in an online coarse-to-fine way.

However, it is still difficult to align feature representation on server domain shifts without extra information. We attempt to extract universal features from the domains of humans and real animals, which is similar to cross-domain APE. In this paper, we mainly focus on transferring the human domain to the animal domain.

3. Method

Recently, most vision tasks have experienced rapid development from convolutional neural networks to vision transformers. Transformers have been used in the human pose estimation (HPE) [11] with competitive behavior. In this paper, we use the vision transformer as our basic structure, and then build a baseline model for animal pose estimation (APE). Owing to the absence of a rich and diverse animal keypoint detection dataset and the availability of substantial labeled human pose data, we leverage human and animal pose datasets together to train a strong representational encoder backbone. Then, we design two decoders to estimate the keypoint for humans and animals because the keypoint definition of human and animal is different. However, there exists a domain shift between the animals and the humans, we further introduce a simple but efficient cross-domain discriminator to bridge the domain gap. The pipeline overview of our THANet is shown in Figure 1. In the following section, we will describe our approach concretely. Section 3.1 introduces the formulaic definition of HPE and APE. Section 3.2 illustrates the basic ViT architecture we used. Section 3.3 explains the cross-domain module in our approach. Section 3.4 presents the loss function.

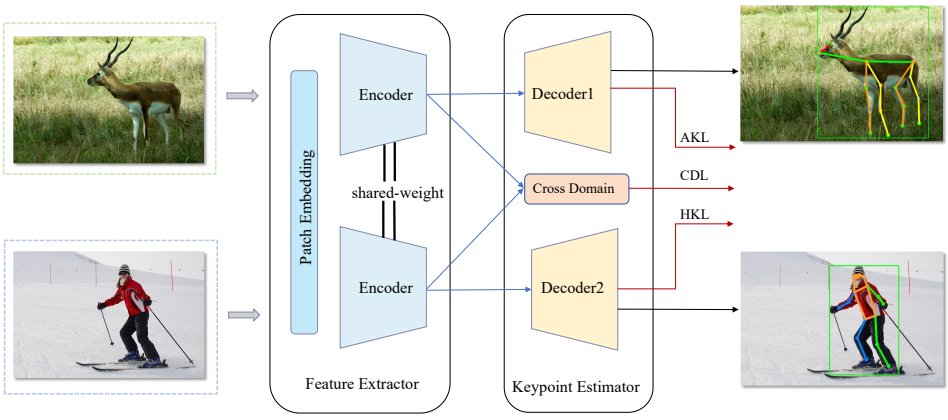

Figure 1. The pipeline of the proposed THANet. The input animal or human images are first divided into patches. The shared-weight encoder extracts universal features via a cross-domain module, i.e., domain discriminator. Two independent decoders are used to compute the keypoint of animals and humans. The points generated by decoders are shown in color. THANet has three losses, including cross-domain loss (CDL), human keypoint loss (HKL), and animal keypoint loss (AKL), respectively.

3.1. Problem Definition

Human and Animal Pose Estimation

Human and animal pose estimation is a classic vision task that seeks to identify and track the keypoint position in images or videos that contain our target object (human or animal). According to different types of input and output, there are 2D pose estimation, 3D pose estimation, and others. In this paper, we focus our attention on 2D animal pose estimation, which tries to detect the keypoints of animals from sRGB images. We use the symbol notation $h = (h_1^i, h_2^i, ..., h_k^i)$ to represent keypoint, where $h_k^i = (x_k, y_k)$ denotes the 2D location of the k-th object's joint in the i-th image. The keypoint can also be represented through heatmap H_k^i. In a heatmap, the keypoint position is typically represented by high-intensity pixel values, while the surrounding pixels gradually decrease in intensity. This allows for a visual representation of the location and confidence level of each keypoint in the image.

In deep learning-based methods, the 2D image I_i is fed into a well-manually designed deep neural network (CNNs or other DNNs) f, and the network forward step generates the output heatmap $\hat{H}_k^i \in R^{K \times H \times W}$ for the keypoints:

$$\hat{H}_k^i = f(I_i), \tag{1}$$

where K represents the total count of keypoints in each sample, and $\hat{H}_k^i$ represents the predicted heatmap for the k-th predicted keypoint in the i-th sample. To be more specific, the highest score position of k-th heatmap can be converted into coordinates (x_k, y_k), and the predicted scores around this are lower than the maximal score. The calculation of the keypoint loss is the distance between the output heatmaps and the ground-truth labels:

$$\mathcal{L} = \frac{1}{N \times K} \sum_{i=1}^{N} \sum_{k=1}^{K} ||H_k^i - \hat{H}_k^i||^2, \tag{2}$$

where N represents the number of samples, and H_k^i represents the k-th keypoint's ground-truth heatmap in the i-th sample. By minimizing $\mathcal{L}$, we train the DNNs to improve the accuracy of pose estimation in a deep learning manner.

3.2. Vision Transformer Architecture

Inspired by the impressive work of ViT in HPE [11], we design a unified encoder based on the vision transformer [9] to extract universal features of the images from animals and humans. For the keypoint decoder part, we apply two versions of the decoder by following the method in [11], i.e., classical and simple, to generate output keypoints.

3.2.1. Encoder

Our encoder is shown in Figure 2a, and the encoder block used for feature extraction is shown in Figure 2b.

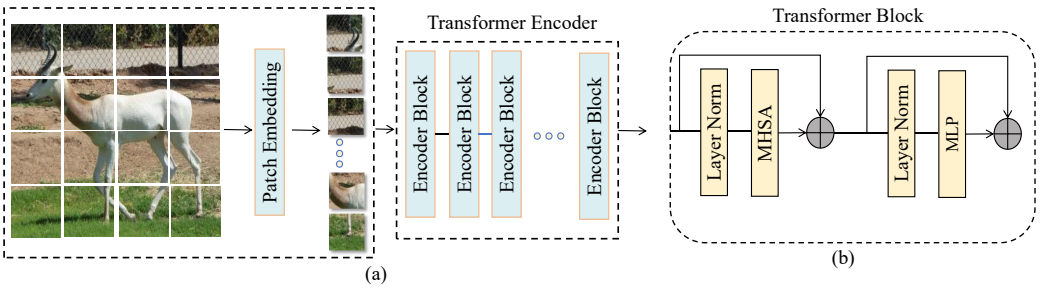

Figure 2. The encoder of THANet. (**a**) The encoder architecture. (**b**) The encoder block.

First, the image I is fed into the patch embedding layer to be divided into different tokens, which are then input into encoder blocks for feature extraction. The transformer

block includes layer normalization [34], multi-head self-attention (MHSA) [35], skip connection [8], and a multi-layer perceptron (MLP):

$$F_0 = \mathrm{PEL}(I_i), \tag{3}$$

$$F_j^{\mathrm{inter}} = F_{j-1} + \mathrm{MHSA}\left(\mathrm{LayerNorm}\left(F_{j-1}\right)\right), \tag{4}$$

$$F_j = F_j^{\mathrm{inter}} + \mathrm{MLP}\left(\mathrm{LayerNorm}\left(F_j^{\mathrm{inter}}\right)\right), \tag{5}$$

where F_j represents the feature extracted by the j-th transformer encoder block. Before inputting into the transformer encoder, we process the sRGB image to obtain the initial tokens by using Equation (3). PEL denotes the patch embedding layer. As shown in Figure 2a of the paper, PEL takes an image as input and outputs a series of small patches or tokens. LayerNorm normalizes features for each sample. MHSA calculates the attention weight map to focus on important features. MLP applies a linear transformation of features. According to the original design of the vision transformer, we retain its property that the feature map size is unchanged after passing each block, so the final output feature of the THANet encoder is expressed as $F_{\mathrm{out}} \in \mathcal{R}^{\frac{H}{d} \times \frac{W}{d} \times C}$. And d is the square root of the total token count in image I_i.

3.2.2. Decoder

As to the decoder, to minimize time consumption, we use two efficient decoders to decode the universal features extracted by the shared-weight encoder network and process the predicted keypoints by following the post-process procedure.

As shown in Figure 3, the classic decoder contains batch normalization [36], ReLU [37], and deconvolution blocks. The design of Upsample originates from the previous HPE method [38], which applies the Deconvolution operator to upsample the feature maps by 2 times. Then, we use a 1×1 convolution layer to obtain heatmaps $\hat{H}^i$ with K channels. Thus, we formulate the classic decoder as:

$$\mathrm{Upsample}(F_{\mathrm{out}}) = \mathrm{ReLU}(\mathrm{BatchNorm}(\mathrm{Deconv}(F_{\mathrm{out}}))), \tag{6}$$

$$\hat{H}^i = \mathrm{Conv}_{1\times1}(\mathrm{Upsample}(\mathrm{Upsample}(F_{\mathrm{out}}))). \tag{7}$$

Apart from this, as shown in Figure 3, we also use another simpler decoder in our methods. Considering the computing cost and strong feature extraction capacity of the ViT encoder, such a decoder is also effective. Concretely, we use the bilinear interpolation method, which upsamples the feature map size to 4 times compared to the original size, with a ReLU activation function and a 3×3 convolution operator to generate the final predicted heatmaps $\hat{H}^i$. The simpler decoder is formulated as:

$$\hat{H}^i = \mathrm{Conv}_{3\times3}(\ \mathrm{Bilinear}\ (\mathrm{ReLU}(F_{\mathrm{out}}))). \tag{8}$$

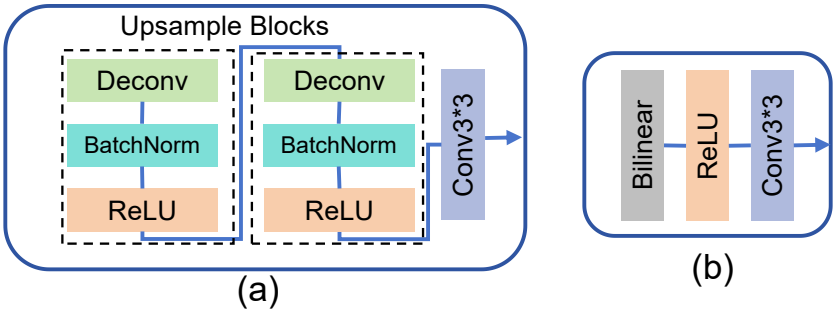

Figure 3. The decoder of THANet. (**a**) The classic decoder. (**b**) The simple decoder.

3.2.3. Joint Learning

The training of THANet is achieved by a joint-learning strategy, which trains the network on two or more kinds of datasets from different domains. Here, we train our THANet on both human and animal pose data. We first uniformly sample images from the human and animal pose datasets. Then, we feed these images from a different domain into the shared-weight encoder of THANet to attain universal features. Finally, we deliver the universal features to the corresponding decoder and obtain the detected joints. Note that the HPE decoder in THANet is removed in the testing stage.

Since we need to achieve APE and HPE simultaneously, the pose estimation loss consists of two losses, animal keypoint loss (AKL) and human keypoint loss (HKL), and AKL and HPL are both the mean-squared error (MSE). The overall loss function for pose estimation $\mathcal{L}_{\text{PEL}}$ is formulated as follows:

$$\mathcal{L}_{\text{PEL}} = \frac{1}{N_a}\sum_{i=1}^{N_a}(\mathcal{L}_{\text{AKL}}(f(I_i^a), H^a)) + \frac{1}{N_h}\sum_{i=1}^{N_h}\Big(\mathcal{L}_{\text{HKL}}(f(I_i^h), H^h)\Big), \tag{9}$$

where N_a and N_b represent the total images of humans and animals, respectively. f is our proposed network THANet. I_i^a and I_i^h denote the input image of animals and humans, respectively. H_a^i and H_h^i represent the ground-truth heatmaps of I_i^a and I_i^h, respectively.

3.3. Cross-Domain Adaptation

Based on the joint-learning strategy, we further introduce a simple but effective cross-domain module. Domain shift is an unavoidable issue in transfer learning. This issue also exists in our work, simply joint-training two datasets from different domains negatively affects the behavior of THANet. As shown in Figure 1, to narrow the domain gap between the human and animal domains and enhance the universal features extracted by the shared-weight encoder, the output of the encoder steps forward to two parallel branches, which are the keypoint estimation branch and the cross-domain branch. In this branch, we add a domain discriminator g to classify the obtained features, which is a fully convolutional network (FCN). The domain discriminator attempts to classify animals and humans, and the parameters of the cross-domain module are optimized based on cross-entropy loss to construct the cross-domain loss (CDL). Unlike the well-trained classifier, we expect that the domain discriminator does not have the ability to classify the domain correctly. CDL is formulated as follows:

$$\mathcal{L}_{\text{CDL}} = -\frac{1}{N_a + N_b}\sum_{i=1}^{N_a+N_b}(z_i \log(g(F_{\text{out}})) + (1 - z_i)\log(1 - g(F_{\text{out}}))), \tag{10}$$

where z_i represents the domain ground-truth label of image I_i ($z_i = 0$ for humans and $z_i = 1$ for animals); $g(F_{\text{out}})$ is the binary probability distribution of the domain discriminator g.

3.4. Overall Loss Function

Our proposed model includes three parts of loss calculation, where the human pose estimation loss and animal pose estimation loss are merged into a single pose loss and the pre-advise discriminator for reducing domain discrepancy as the domain discriminator loss. In order to achieve this, we expect to maximize the auxiliary cross-domain module loss $\mathcal{L}_{\text{CDL}}$ and minimize the pose estimation loss $\mathcal{L}_{\text{PEL}}$. Our THANet is expected to achieve better behavior by leveraging universal features extracted from two domains by this design. The total loss of the THANet is formulated as:

$$\mathcal{L} = \alpha\mathcal{L}_{\text{CDL}} + \beta\mathcal{L}_{\text{PEL}}. \tag{11}$$

Since the training procedures of the keypoint detection branch and cross-domain branch are very similar to adversarial training, we need to set $\alpha \times \beta < 0$, which encourages domain

confusion and boosts APE behavior simultaneously. In the experiments, we set $\alpha = -1$, $\beta = 1$ for the training of THANet.

4. Experimental Section

4.1. Dataset

We use three datasets: AP-10K [31], MS COCO Keypoint Detection [39], and Animal-Pose [6]. The AP-10K and MS COCO are used for the main training and testing, while we conduct our generalization experiments on Animal-Pose. The distribution of Animal-Pose is not the same with AP-10K, so we conducted experiments on Animal-Pose to verify the generalizability of THANet.

AP-10K is the large-scale benchmark for mammal animal pose estimation [31], which consists of 10 K high-quality keypoint annotated images collected from 54 species. The COCO dataset contains over 200 K images [39]. We randomly sample 10 K samples from the MS COCO Human Keypoint Detection for training with the 10 K animal images from AP-10K together. Animal-Pose [6] is collected from the internet or other dataset, which contains five common quadruped categories: *cow*, *horse*, *sheep*, *cat*, and *dog*. The scale of Animal-Pose is relatively small compared with AP-10K, which limits the application. There are 3 K images in total, and 5 K instances are annotated with 20 keypoints.

4.2. Implementation Details and Evaluation Metrics

4.2.1. Experimental Settings

We implement some representative methods based on the MMPose codebase [40] in a top–down manner. Two A100 GPUs with 40 GB memory are used during both the training and testing for all the experiments. Before the training procedure begins, the backbones are always initialized with pre-trained weights of MAE [41]. The input image resolution is 256×256 for all experiments. The mini-batch size is 64 for each GPU. For stable training, we use AdamW [42] to optimize the network for 210 epochs. The initial learning rate is set to 5×10^{-4} decay by a factor of 10 at 170-th and the 200-th epoch. The hyperparameters of ViT are the same with [9], and we use the classical decoder as our main decoder in our experiments for fair comparison with well-designed CNNs. HPE post-processing method UDP [43] is applied in our experiment, following the previous work [31].

4.2.2. Metrics

In the basic experiment part, we use the standard pose estimation performance evaluation metric average precision (AP) and average recall (AR), which is further calculated via object keypoint similarity (OKS). The details of OKS are formulated as follows:

$$\mathrm{OKS}(\hat{H}^i, H^i) = \frac{\sum_k^K \exp(-d(\hat{H}_k^i, H_k^i)^2 / 2s_k^2 W_k^2)\delta(v_k > 0)}{\sum_{k=1}^K \delta(v_k > 0)}, \tag{12}$$

where d calculates the pixel–domain distance between the predictions and the ground-truths, v_k represents the visibility of the keypoint (0 if lack of annotation, and 1 otherwise), s_k represents the sigma of the k-th keypoint, and W_k is a per-keypoint weight. δ represents the indicator function (the output is equal to 0 if $v_k = 0$ and 1 otherwise). W_k is set to 1 for k = 1...K. The value of s_k is set to [0.025, 0.025, 0.035, 0.079, 0.072, 0.062, 0.079, 0.072, 0.062, 0.107, 0.087, 0.089, 0.107, 0.087, 0.089] by following [31]. OKS measures the similarity between predicted keypoints and ground-truth keypoints. First, we calculate the distance for each prediction and ground-truth pair. Then, the exp function maps the distance into values ranging from 0 to 1. Finally, the average values of all annotated keypoints are obtained; higher OKS values represent better keypoint estimation results. AP and AR are based on OKS, which are formulated as follows:

$$\mathrm{AP}^t = \frac{\sum_{k=1}^K \delta(\max(\hat{H}_k^i) \geq T_{\text{score}}) \times \delta(\mathrm{OKS}(\hat{H}_k^i, H_k^i) > (t/100.0)) \times \delta(v_k > 0)}{\sum_{k=1}^K \max(\hat{H}_k^i) \geq T_{\text{score}}}, \tag{13}$$

$$\mathrm{AR}^t = \frac{\sum_{k=1}^{K} \delta(\max(\hat{H}_k^i) \geq T_{\text{score}}) \times \delta(\mathrm{OKS}(\hat{H}_k^i, H_k^i) > (t/100.0)) \times \delta(v_k > 0)}{\sum_{k=1}^{K} \delta(v_k > 0)}, \quad (14)$$

where K represents the total number of defined keypoints; T_{score} represents score threshold for retaining high-confidence keypoint. t represents the threshold for AP or AR, which controls the strictness of evaluation. The max obtains the maximal predicted scores of the heatmap. Ten thresholds with a step of 5 ranging from 50 to 95 ($t = 50, 55, \ldots, 95$) are used to attain AP^t and AR^t. AP^{50} and AR^{50} are results in the condition of $t = 50$. The larger the AP^t and AR^t, the more accurate the result, which is like OKS.

To align with previous works [4,5], we also apply the percentage of correct keypoint (PCK) as the evaluation metric in the generalization testing experiment. PCK calculates the accuracy by measuring the distance between the predicted joint location and the ground-truth joint location. In other words, first, we obtain the distance between predictions and ground-truth labels, then normalize the distance by the heatmap size. Then, we compare the normalized distance and t, which represents whether this prediction is correct or not. Finally, we count the number of correct keypoints, which is divided by the total number of annotated keypoints. The details of PCK are formulated as follows:

$$\mathrm{PCK}^t = \frac{\sum_{k=1}^{K} \delta\left(\frac{d(\hat{H}_k^i, H_k^i)}{d_p^{\text{def}}} \leq t\right)}{\sum_{k=1}^{K} \delta(v_k > 0)}, \quad (15)$$

where k represents the k-th keypoint in the i-th image, t represents threshold (range from 0 to 1), p represents the p-th species, and d_p^{def} represents the scale factor of the p-th species. In this work, t is 0.05 and d_p^{def} is 256 (the size of the heatmap), following [4,5]. Note that the metrics formulated in this paper are calculated on a single image. To obtain the final results, we accumulate and average metric scores for all images.

4.3. Quantitative Analysis

4.3.1. Comparison with the Methods Based on CNNs

We compare the HRNet [26] and SB [25] on AP-10K, as shown in Table 2. Our method outperforms the previous convolutional methods by a large margin at a 256×256 image resolution, which achieves 77.12% AP and 80.32% AR, with a margin of 4.17% AP over the CNN-based HRNet-w48. For metric AP^{50} and AR^{50}, our THANet is also higher than the CNN-based method. We also compare the complexity of these methods. For training, the average cost time of THANet for one epoch is less than that of HRNet but more than SB. This is because HRNet keeps the large resolution all the time and acquires more convolutions on high-resolution features. The main operations in transformers are matrix multiplications, which make batch processing easier to accelerate on GPUs. For testing, our method achieves gains of 4.17% AP, 2.04% AP^{50}, 4.04% AR, and 2.08% AR^{50} over HRNet-w48, with little but worthwhile increase of inference time cost (+1.74 ms).

4.3.2. Ablation Study

We conduct the ablation study to verify the contribution of our proposed joint-learning and cross-domain module. In Table 3, JL expresses that we use the proposed joint-learning strategy for training on the COCO and AP-10K datasets. The AP and AR scores decreased by 1.76% and 1.51%, respectively. The decrease in JL in AP and AR scores is due to the domain gap.

Although they share some common features, humans and animals are from two different domains with a certain domain gap. Simply training on the two datasets from different domains may hurt the performance. Compared with the plain ViT network, our approach with the JL strategy and CD module improved AP and AR scores, 1.14% and 1.06%, respectively, which verified the effectiveness of our cross-domain module. Note

that the CD module is built on JL, which is necessary to receive features from different modalities. Therefore, we cannot achieve the results with CD but without JL.

Table 2. Comparison with HRNet, SB, and our approach on AP-10K. We also compare the calculation time during the training and testing stages. For training, we conduct experiments on two A100 GPUs with 64 samples per GPU; then, we measure the calculation time for each epoch and calculate the average time of 210 epochs. For testing, we measure the inference speed for each sample. We run this three times and take the average of inference time.

Model	Backbone	Params (M)	Training Time (s)	Inference Time (ms)	Resolution	AP	AP^{50}	AR	AR^{50}
HRNet	HRNet32	28.54	90.09	8.13	256×256	72.46	94.24	75.81	94.95
HRNet	HRNet48	63.59	91.59	10.33	256×256	72.95	94.28	76.28	95.04
SB	ResNet50	33.99	47.60	6.88	256×256	67.96	91.92	71.68	92.88
SB	ResNet101	52.99	49.08	7.28	256×256	68.25	92.01	71.78	92.95
Ours	ViT-B	104.08	71.93	12.07	256×256	**77.12**	**96.24**	**80.32**	**97.12**

Table 3. Ablation study of joint-learning (JL) strategy and cross-domain (CD) module conducted on AP-10K and COCO datasets. AP and AR results on AP-10K benchmark. '✓' represents that we add this module or strategy while conducting the experiment.

ViT	JL	CD	AP	AP^{50}	AR	AR^{50}
✓			75.98	95.42	79.26	96.11
✓	✓		74.22	94.72	77.71	95.31
✓	✓	✓	**77.12**	**96.24**	**80.32**	**97.12**

4.3.3. Generalize to Animal-Pose Dataset

Table 4 shows that other cross-domain methods perform poorly on unseen species like *dogs*, *cats*, and *sheep*. Our method performs the best in the five categories, which indicates that our method has great generalizability. This is because our method is trained on COCO and the AP-10K, another reason is that our backbone network has a better feature extraction capability than theirs.

Table 4. PCK@0.05 accuracy for the generalization compared with CC-SSL and UDA on Animal-Pose benchmark.

Method	Horse	Dog	Cat	Sheep	Cow	Mean
CC-SSL [33]	65.35	30.27	15.05	52.39	63.71	47.6
UDA [5]	72.84	42.48	27.65	59.51	71.31	56.77
Ours	**86.86**	**77.01**	**69.36**	**79.11**	**88.26**	**81.53**

4.4. Qualitative Analysis

Figure 4 shows some competitive results on the AP-10K (the red dots and lines in Figure 4c represent our predictions). It shows that the results of our method are accurate and have good generalization, which accurately predicts the keypoints for a variety of animals with different poses. Moreover, the proposed THANet accurately generalizes to some keypoints that are not present in the ground-truth annotations. While HRNet and SB excel in keypoint detection and other vision tasks, THANet provides a more comprehensive and precise localization and recognition results.

Similarly, we visualize the results on Animal-Pose to illustrate the generalizability of THANet. As shown in Figure 5, compared to other cross-domain animal pose estimation methods, our proposed THANet exhibits a more robust species generalization, capable of generalizing to a broader range of species with minimal prediction errors on their body parts.

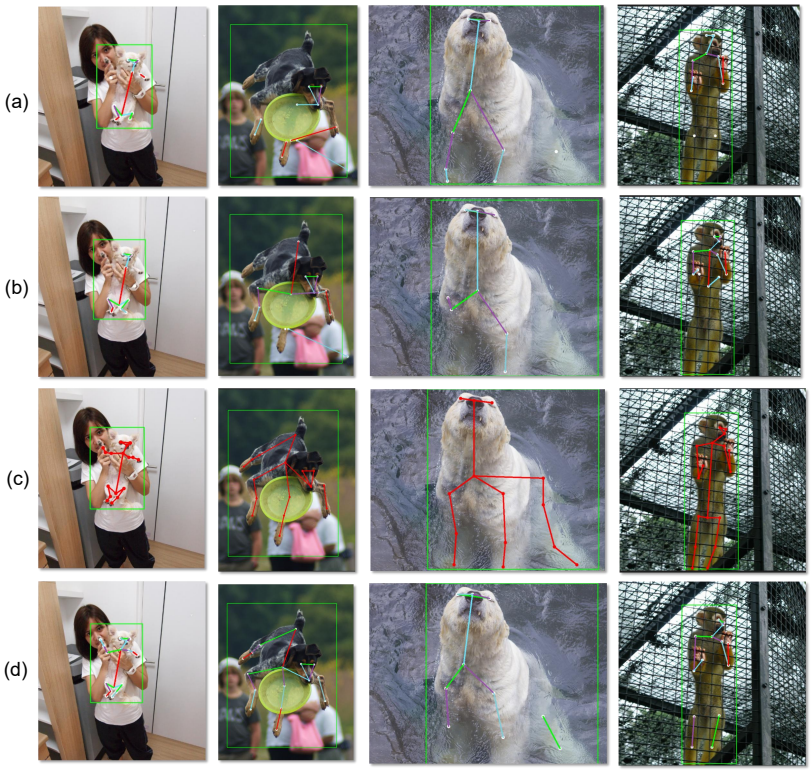

Figure 4. Visual comparison between different methods on the AP-10K. (**a**) SB-resnet50, (**b**) HRNet-width32, (**c**) THANet, (**d**) ground-truth. We highlight the pose predicted by THANet with red lines.



Figure 5. Visualization of generalization on Animal-Pose. (**a**) CC-SSL, (**b**) UDA, (**c**) THANet, (**d**) ground-truths. We highlight the pose predicted by THANet with red lines.

5. Discussion

In this section, we summarize our findings on APE and compare them with previous studies. Then, we will analyze the limit of the proposed THANet and present future research directions about APE.

We focus on transferring knowledge from human to animal and on achieving satisfactory results on AP-10K and Animal-Pose. The main basic network of THANet is the vision transformer, which reveals the same pattern as that of previous ViT-based work in other vision tasks and has strong feature extraction ability in computer vision. The commonalities in skeletons between humans and animals are also worth noting. We can leverage human pose data to alleviate the problem of insufficient animal pose data and improve generalizability.

Although we have achieved remarkable results on APE, for animals that have huge vision differences in the skeleton compared to humans, the effectiveness of the proposed approach is limited. Thus, how to fully use human pose data to train a cross-domain APE model is still to be explored.

6. Conclusions

In this paper, we propose an approach for cross-domain animal pose estimation based on the vision transformer. We improve the backbone network for extracting features from the traditional convolutional network to a visual transformer, which achieves excellent performance. In addition, to alleviate the problem of insufficient animal pose datasets and the poor generalizability to other species, we propose THANet for the joint training of multiple datasets to perform mixed training on human pose data and animal pose data. At the same time, we introduce a domain classifier to narrow the gap between the human domain and the animal domain, resulting in the model being insensitive to species, with an improved generalizability and robustness. In the future, we will further explore different domain generalization methods on large datasets to improve the model's performance and generalizability.

Author Contributions: Conceptualization, J.L. and S.L.; Methodology, J.L.; Software, J.X.; Validation, J.L., Y.S. and S.L.; Formal analysis, J.X.; Investigation, J.L.; Resources, S.L.; Data curation, J.X.; Writing—original draft, J.L.; Writing—review & editing, Y.S. and S.L.; Visualization, J.L.; Supervision, J.X.; Project administration, S.L.; Funding acquisition, S.L. All authors have read and agreed to the published version of the manuscript.

Funding: This work is supported by the National Natural Science Foundation of China (NO. 62102151), Shanghai Sailing Program (21YF1411200), CCF-Tencent Open Research Fund, the Open Research Fund of Key Laboratory of Advanced Theory and Application in Statistics and Data Science, Ministry of Education (KLATASDS2305), the Fundamental Research Funds for the Central Universities.

Data Availability Statement: Data available in a publicly accessible repository that does not issue DOIs. Publicly available datasets were analyzed in this study. (1) AP-10K can be found here: https://github.com/AlexTheBad/AP-10K. (2) COCO can be found here: https://cocodataset.org. (3) Animal Pose can be found here: https://sites.google.com/view/animal-pose.

Conflicts of Interest: The authors have no competing interests to declare that are relevant to the content of this article.

Abbreviations

The following abbreviations are used in this manuscript:

APE	Animal Pose Estimation
HPE	Human Pose Estimation
AKL	Animal Keypoint Loss
HKL	Human Keypoint Loss
CDL	Cross-Domain Loss

JL	Joint Learning
CD	Cross-Domain
AP	Average Precision
AR	Average Recall
OKS	Object Keypoint Similarity
PCK	Percentage of Correct Keypoints

References

1. Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; Bengio, Y. Decoding complete reach and grasp actions from local primary motor cortex populations. *J. Neurosci.* **2010**, *30*, 9659–9669.
2. Wenger, N.; Moraud, E.M.; Raspopovic, S.; Bonizzato, M.; DiGiovanna, J.; Musienko, P.; Morari, M.; Micera, S.; Courtine, G. Closed-loop neuromodulation of spinal sensorimotor circuits controls refined locomotion after complete spinal cord injury. *Sci. Transl. Med.* **2014**, *6*, 255ra133. [CrossRef] [PubMed]
3. 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]
4. Mu, J.; Qiu, W.; Hager, G.D.; Yuille, A.L. Learning from synthetic animals. In Proceedings of the CVPR, Seattle, WA, USA, 14–19 June 2020; pp. 12386–12395.
5. Li, C.; Lee, G.H. From synthetic to real: Unsupervised domain adaptation for animal pose estimation. In Proceedings of the CVPR, Nashville, TN, USA, 19–25 June 2021; pp. 1482–1491.
6. Cao, J.; Tang, H.; Fang, H.S.; Shen, X.; Lu, C.; Tai, Y.W. Cross-Domain Adaptation for Animal Pose Estimation. In Proceedings of the ICCV, Seoul, Republic of Korea, 27 October–2 November 2019.
7. Simonyan, K.; Zisserman, A. Very Deep Convolutional Networks for Large-Scale Image Recognition. In Proceedings of the ICLR, San Diego, CA, USA, 7–9 May 2015.
8. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 25 June–1 July 2016; pp. 770–778.
9. 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 16 $\times$ 16 words: Transformers for image recognition at scale. *arXiv* **2020**, arXiv:2010.11929.
10. 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 ICCV, Montreal, BC, Canada, 11–17 October 2021; pp. 10012–10022.
11. Xu, Y.; Zhang, J.; Zhang, Q.; Tao, D. ViTPose: Simple Vision Transformer Baselines for Human Pose Estimation. In Proceedings of the Advances in Neural Information Processing Systems, New Orleans, LA, USA, 28 November–9 December 2022.
12. Andriluka, M.; Roth, S.; Schiele, B. Pictorial structures revisited: People detection and articulated pose estimation. In Proceedings of the CVPR, IEEE, Miami, FL, USA, 20–25 June 2009; pp. 1014–1021.
13. Sapp, B.; Jordan, C.; Taskar, B. Adaptive pose priors for pictorial structures. In Proceedings of the CVPR, IEEE, San Francisco, CA, USA, 13–18 June 2010; pp. 422–429. [CrossRef]
14. Dantone, M.; Gall, J.; Leistner, C.; Van Gool, L. Human pose estimation using body parts dependent joint regressors. In Proceedings of the CVPR, Portland, OR, USA, 23–28 June 2013; pp. 3041–3048.
15. Toshev, A.; Szegedy, C. Deeppose: Human pose estimation via deep neural networks. In Proceedings of the CVPR, Columbus, OH, USA, 23–28 June 2014; pp. 1653–1660.
16. Tompson, J.; Goroshin, R.; Jain, A.; LeCun, Y.; Bregler, C. Efficient object localization using convolutional networks. In Proceedings of the CVPR, Boston, MA, USA, 7–12 June 2015; pp. 648–656.
17. Wei, S.E.; Ramakrishna, V.; Kanade, T.; Sheikh, Y. Convolutional pose machines. In Proceedings of the CVPR, Las Vegas, NV, USA, 27–30 June 2016; pp. 4724–4732.
18. Newell, A.; Yang, K.; Deng, J. Stacked hourglass networks for human pose estimation. In Proceedings of the ECCV, Amsterdam, The Netherlands, 11–14 October 2016; pp. 483–499.
19. Chen, Y.; Wang, Z.; Peng, Y.; Zhang, Z.; Yu, G.; Sun, J. Cascaded Pyramid Network for Multi-Person Pose Estimation. In Proceedings of the CVPR, Salt Lake City, UT, USA, 18–22 June 2018.
20. Wang, J.; Long, X.; Gao, Y.; Ding, E.; Wen, S. Graph-PCNN: Two Stage Human Pose Estimation with Graph Pose Refinement. In Proceedings of the ECCV, Glasgow, UK, 23–28 August 2020; Vedaldi, A., Bischof, H., Brox, T., Frahm, J.M., Eds.; Springer: Cham, Switzerland, 2020; pp. 492–508.
21. Li, K.; Wang, S.; Zhang, X.; Xu, Y.; Xu, W.; Tu, Z. Pose Recognition With Cascade Transformers. In Proceedings of the CVPR, Nashville, TN, USA, 20–25 June 2021; pp. 1944–1953.
22. Nie, X.; Feng, J.; Zhang, J.; Yan, S. Single-Stage Multi-Person Pose Machines. In Proceedings of the ICCV, Seoul, Republic of Korea, 27 October–2 November 2019.
23. Kreiss, S.; Bertoni, L.; Alahi, A. PifPaf: Composite Fields for Human Pose Estimation. In Proceedings of the CVPR, Long Beach, CA, USA, 15–20 June 2019.
24. Geng, Z.; Sun, K.; Xiao, B.; Zhang, Z.; Wang, J. Bottom-Up Human Pose Estimation via Disentangled Keypoint Regression. In Proceedings of the CVPR, Nashville, TN, USA, 20–25 June 2021; pp. 14676–14686.

25. Xiao, B.; Wu, H.; Wei, Y. Simple baselines for human pose estimation and tracking. In Proceedings of the ECCV, Munich, Germany, 8–14 September 2018; pp. 466–481.
26. Sun, K.; Xiao, B.; Liu, D.; Wang, J. Deep high-resolution representation learning for human pose estimation. In Proceedings of the CVPR, Long Beach, CA, USA, 15–20 June 2019; pp. 5693–5703.
27. Kendall, A.; Grimes, M.; Cipolla, R. PoseNet: A Convolutional Network for Real-Time 6-DOF Camera Relocalization. In Proceedings of the ICCV, Santiago, Chile, 7–13 December 2015.
28. Rieger, I.; Hauenstein, T.; Hettenkofer, S.; Garbas, J.U. Towards Real-Time Head Pose Estimation: Exploring Parameter-Reduced Residual Networks on In-the-wild Datasets. In Proceedings of the Advances and Trends in Artificial Intelligence. From Theory to Practice, Graz, Austria, 9–11 July 2019; Wotawa, F., Friedrich, G., Pill, I., Koitz-Hristov, R., Ali, M., Eds.; Springer: Cham, Switzerland, 2019; pp. 123–134.
29. Bruno, A.; Moore, M.; Zhang, J.; Lancette, S.; Ward, V.P.; Chang, J. Toward a head movement-based system for multilayer digital content exploration. *Comput. Animat. Virtual Worlds* **2021**, *32*, e1980. [CrossRef]
30. Del Pero, L.; Ricco, S.; Sukthankar, R.; Ferrari, V. Articulated motion discovery using pairs of trajectories. In Proceedings of the CVPR, Boston, MA, USA, 7–12 June 2015; pp. 2151–2160.
31. Yu, H.; Xu, Y.; Zhang, J.; Zhao, W.; Guan, Z.; Tao, D. AP-10K: A Benchmark for Animal Pose Estimation in the Wild. In Proceedings of the Thirty-Fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2), Virtual, 6–14 December 2021.
32. Ng, X.L.; Ong, K.E.; Zheng, Q.; Ni, Y.; Yeo, S.Y.; Liu, J. Animal Kingdom: A Large and Diverse Dataset for Animal Behavior Understanding. In Proceedings of the CVPR, New Orleans, LA, USA, 19–24 June 2022; pp. 19023–19034.
33. Ma, Q.; Yang, J.; Ranjan, A.; Pujades, S.; Pons-Moll, G.; Tang, S.; Black, M.J. Learning to dress 3d people in generative clothing. In Proceedings of the CVPR, Seattle, WA, USA, 13–19 June 2020; pp. 6469–6478.
34. Ba, J.L.; Kiros, J.R.; Hinton, G.E. Layer normalization. *arXiv* **2016**, arXiv:1607.06450.
35. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. *Adv. Neural Inf. Process. Syst.* **2017**, *30*, 6000–6010.
36. Ioffe, S.; Szegedy, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In Proceedings of the International Conference on Machine Learning, PMLR, Lille, France, 7–9 July 2015; pp. 448–456.
37. Agarap, A.F. Deep learning using rectified linear units (relu). *arXiv* **2018**, arXiv:1803.08375.
38. Zhang, J.; Chen, Z.; Tao, D. Towards high performance human keypoint detection. *Int. J. Comput. Vis.* **2021**, *129*, 2639–2662.
39. 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 ECCV, Zurich, Switzerland, 6–12 September 2014; pp. 740–755.
40. Openmmlab Pose Estimation Toolbox and Benchmark. 2020. Available online: https://github.com/open-mmlab/mmpose (accessed on 1 October 2023).
41. He, K.; Chen, X.; Xie, S.; Li, Y.; Dollár, P.; Girshick, R. Masked autoencoders are scalable vision learners. In Proceedings of the CVPR, New Orleans, LA, USA, 19–24 June 2022; pp. 16000–16009.
42. Reddi, S.J.; Kale, S.; Kumar, S. On the convergence of adam and beyond. *arXiv* **2019**, arXiv:1904.09237.
43. 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 CVPR, Seattle, WA, USA, 13–19 June 2020; pp. 5700–5709.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.