Unified 2D Human Pose Estimation

Dhavalkumar Bharatkumar Limbachiya[1] and Muhammad Saif Ullah Khan[2]

[1] sut23dil@rptu.de
[2] muhammad_saif_ullah.khan@dfki.de

Abstract. The key objective of human pose estimation is to detect the set of keypoints on the human body for understanding the orientation and position of humans. It has many downstream applications in fitness, autonomous driving, surveillance, etc. The research on human pose estimation has progressed and produced numerous state-of-the-art model architectures with the availability of multiple pose estimation datasets. However, each neural network architecture is trained on a particular dataset, which performs adversely when tried on different datasets. As a result, a network that performs well on the majority of datasets is uncommon. In this work, we use the concept of knowledge distillation to train a network architecture capable of efficiently estimating poses across datasets.

Keywords: Human Pose Estimation, Knowledge Distillation, 2D

1 Introduction

Human Pose Estimation (HPE) is a computer vision-based task that involves detecting and tracking the orientation and position of the human body from photos or videos [16]. A pose estimation model takes an image as input and returns a set of detected keypoints with confidence scores. A keypoint indicated by a unique ID represents a body joint. The confidence score indicates the likelihood that a keypoint is present at that position, a value between 0 and 1 [4]. HPE can be achieved through two approaches: top-down and bottom-up [10]. In the top-down approach, the model initially recognizes the subject of the image before detecting the essential keypoints on the body. On the other hand, in the Bottom-Up approach, the model first detects the essential keypoints, followed by grouping together the keypoints of the subject in image [10].

HPE has made significant progress in the last few years with advancements in deep neural networks and the availability of pose estimation datasets. It has various applications in autonomous driving, fitness tracking, security surveillance, etc. HPE can be broadly categorized into 2D and 3D estimations. In this work, we will focus on 2D HPE. The keypoints in 2D HPE are represented using the X and Y axes. Some well known datasets are MPII [5], COCO [14], AIC [17] etc. The typical body keypoints from COCO and MPII are shown in Fig 1. Each dataset has its unique set of annotated body joints. Generally, each neural network is trained on a particular dataset. This approach often results in networks that excel on their specific dataset but struggle when tested on others. Thus, a network that can perform well on most of the datasets rarely exists. Therefore, we aim to create a network capable of efficiently estimating poses across datasets.

The following sections of the paper are organized as follows: Section 2 provides an overview that includes existing methodologies for HPE, a summary of the dataset and evaluation metrics used in these different methodologies, followed by problems with the existing approaches. Section 3 introduces our approach to solving the problem and a detailed description of workflow, followed by experimental results in Section 4 and conclusion in Section 5.

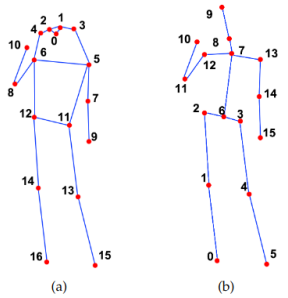

Fig. 1: keypoints in COCO and MPII Datasets

2 Overview

2.1 Existing Approaches

2D heatmap Based A 2D heatmap representation [6][7] is a widely used approach for localizing keypoints on a person's body in an image for human pose estimation. In this approach, the model is provided with an image as an input, which is preprocessed to generate a set of heatmaps for each key point (e.g., shoulder, elbow, wrist). Imagine a grid overlaid on the image, each cell representing a pixel. The intensity value of a specific pixel in a heatmap shows the likelihood that the associated keypoint is at that location. However, this approach has few limitations explained in [13] and [12]. The accuracy of the model can significantly decrease with low-quality images. High-resolution heatmaps require more computational resources to process. Lastly, an additional step might be needed to refine the key point locations due to inherent limitations of the heatmap representation [13].

Coordinate Classification Based To address above mentioned problems, a coordinate classification approach [13][12] was introduced. In the coordinate classification approach, the model predicts 2D coordinates for each key point in an image. It treats this task as two separate classification tasks, one for predicting the x-coordinate and another for predicting the y-coordinate. Research suggests that the coordinate classification approach outperforms heatmap-based approaches in low-resolution settings. Moreover, by removing the need for post-processing steps in heatmap-based methods, coordinate classification offers a more efficient and computationally lightweight approach [13]. For instance - SimCC [12], a Simple Coordinate Classification(SimCC) method for human pose estimation. A clear difference between heatmap-based approaches and coordinate classification-based approaches like SimCC is shown in Fig 2. In this project, we have used MMPose [2] based Real-

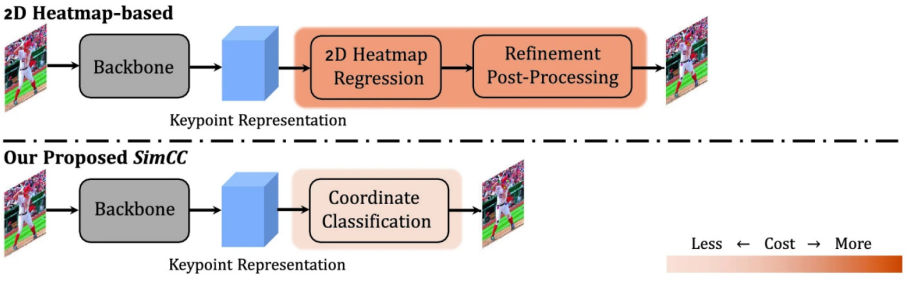

Fig. 2: Comparison between 2D heatmap based method and SimCC from [12]

Time Models for Pose Estimation (RTMPose), a model architecture with SimCC-based algorithm that treats keypoint localization as a classification task [11]. It uses a top-down approach for pose

estimation with CSPNeXt [15] as the backbone for object detection and SimCC [12] as the prediction head of architecture. The overall architecture of RTMPose is depicted in Fig 3, which consist of backbone, processing layer for refining keypoint representation and lastly with SimCC head for predicting X and Y coordinate of keypoints.

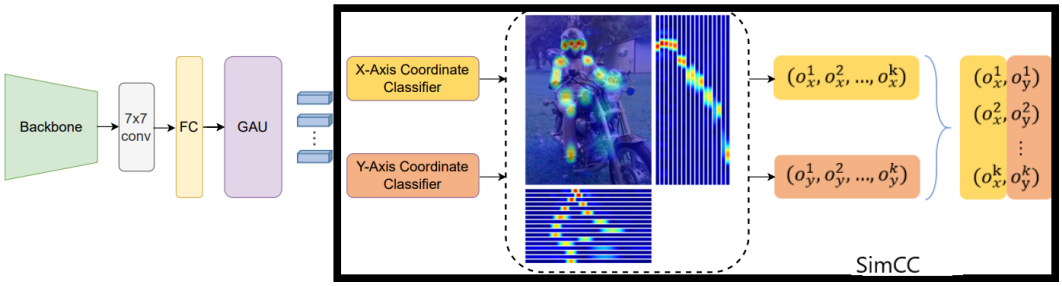

Fig. 3: Architecture of RTMPose from [11]

RTMPose has achieved a good balance between processing speed and accuracy, making it suitable for real-time downstream tasks. Moreover, it performs better than well-established libraries like OpenPose [7] in terms of accuracy and downstream tasks. More information about the performance of RTMPose on the different datasets is discussed in Section 4.

2.2 Datasets

Table 1 contains a brief summary of the datasets used in this work:

Table 1: Datasets

Dataset	Num. Keypoints	Splits	Problem	Figure
COCO	17	Train : 118k Validation: 5k Test: 41k	Does not include any annotations for the spine, which means complex poses which involve a curved spine (e.g., while dancing or working out) cannot be accurately represented.	Fig 1(left)
MPII	16	Train : 15k Validation: 3k Test: 7k	Does not include keypoints for the nose, eyes, and ears. These key points can be important to track the head pose for understanding the sense of direction.	Fig 1(right)

2.3 Evaluation Metric

To evaluate pose estimation models, the following metrics were used in this work:

1. **Average Precision(AP)** Precision defines how accurately the model detects the keypoints. AP is calculated by averaging the precision at multiple Intersection over Union (IoU) thresholds.
2. **Average Recall(AR)** Recall is the portion of annotated object instances for which an estimated correct pose is available [9]. Average recall is the average of the recall rates calculated for multiple threshold settings.
3. **Percentage of Correct keypoints(PCK)** [3] Represents the percentage of correct keypoints detected by the model. It will consider a key point as correct only if the distance between the true and predicted points is within a certain threshold.

2.4 Problem

There are many datasets for 2D human pose estimation, each with its unique set of annotated body joints. Typically, a network is trained for each dataset separately. This approach often results in networks that excel on their specific dataset but struggle when tested on others. Thus, a network capable of efficiently estimating poses across datasets are rarely available.

3 Our approach

3.1 Idea

Using the concept of knowledge distillation introduced in [8], we have used a group of pre-trained teacher networks to help create a single, adaptable student network capable of estimating poses across datasets. Knowledge distillation is a technique used for transferring knowledge from complex, large models (teacher networks) to small models (student networks) [8]. This technique helps the student network achieve similar or even better performance than teacher networks. Thus, we have two teacher models–the COCO teacher model and the MPII teacher model–that help a smaller student model estimate poses in our defined dataset.

3.2 Our Dataset

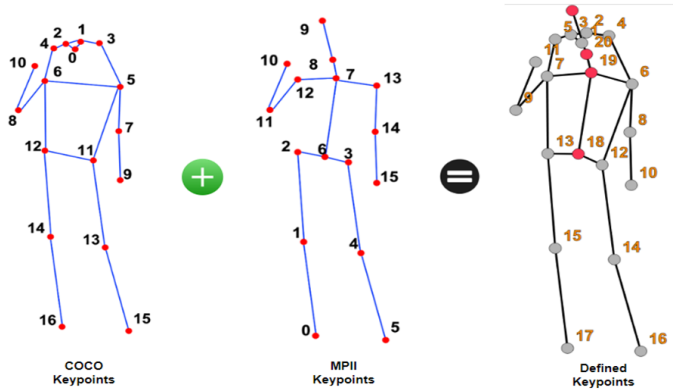

Fig. 4: Defined dataset keypoints as the union of COCO and MPII

We define an extended skeleton, the union of the COCO and MPII keypoints depicted in 4. This skeleton has 21 keypoints, including 17 original COCO and four new keypoints from the MPII dataset to represent the spine.

Pelvis: In pose estimation, the "pelvis" keypoint is usually considered to be roughly in the middle of the left and right hip keypoints. It's the point that, in a frontal view, would appear roughly midway along the line connecting the two hip keypoints. This point references the base of the spinal cord.

Thorax: The thorax, or chest, is a region between the neck and abdomen. In pose estimation, the "thorax" keypoint can be considered the midpoint between left and right shoulder keypoints. It is located at the visible dip between the neck and collarbone at the base of the throat.

Upper Neck: In pose estimation, the "upper neck" keypoint can be considered the point at the base of the neck where the face joins the torso.

Head Top: The "head top" keypoint in pose estimation is the highest point on the head.

Here, we are using images from the COCO and MPII datasets. In the training and inference process, the model uses a dataset configuration file that defines the new dataset keypoints. Thus,

no new data collection and annotation process was performed as we use COCO and MPII images and annotations.

3.3 Pipeline

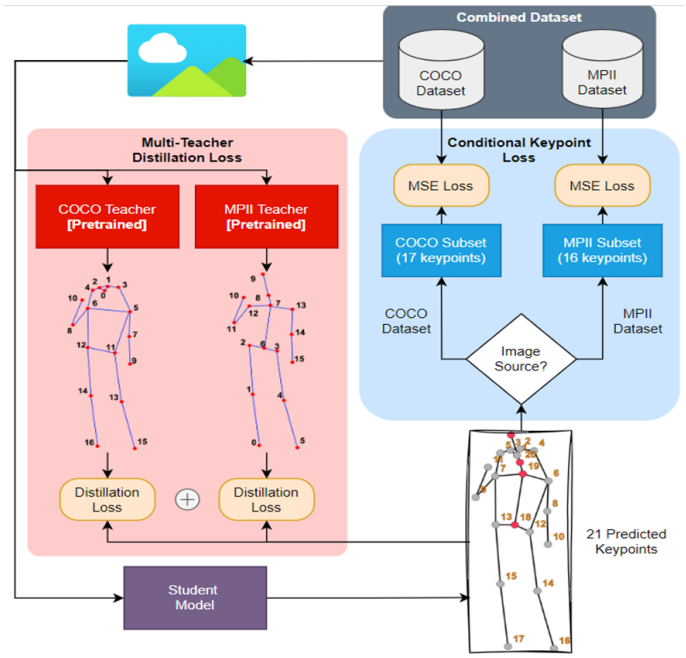

Fig. 5: Pipeline for training student network using COCO and MPII teachers

As mentioned in Section 3.1, we use knowledge distillation to train a student network capable of estimating poses across COCO and MPII data using two teacher models - COCO teacher and MPII teacher models. We use more than one teacher network, it's called multi-teacher distillation. Moreover, we are performing offline knowledge distillation processes, i.e., using pre-trained teacher networks for training the student network [1].

RTMPose-m architecture is used as COCO teacher network and MPII teacher network. As a student network, we have experimented on RTMPose-m, RTMPose-s, and RTMPose-t, which were trained using RTMPose-m-based teacher networks. Detailed results on performance of different student networks are shared in Section 4.

In this work, we are focusing on response-based knowledge distillation. It focuses on the final output layer of the teacher model. The hypothesis is that the student model will learn to mimic the predictions of the COCO and MPII teacher models. This can be achieved by using a loss function, termed the distillation loss, that captures the difference between the logits of the student and the teacher models respectively [1]. As this loss is minimized over training, the student model will improve at making the same predictions as the teachers [1]. Fig 5 depicts the process of response-based knowledge distillation performed using pre-trained COCO and MPII teachers. Here, we calculate two losses - Distillation loss and Conditional keypoint Loss. Distillation loss is calculated using logits of student networks and COCO/MPII teacher networks. Conditional keypoint Loss is calculated between ground truth labels of data and prediction labels of student models. As our student network is predicting COCO and MPII keypoints, we developed a logic to separate COCO prediction label and MPII prediction labels to compare it with respective ground truth labels for conditional keypoint loss calculation.

In the process of model training, we derive a single loss value by calculating weighted loss average of distillation loss and conditional keypoints loss. For given distillation loss for COCO dataset β, distillation loss for MPII dataset γ and conditional keypoint loss α, loss is calculated as

$$Loss(\mathcal{L}) = (w1 * \alpha) + (w2 * \beta) + (w3 * \gamma) \tag{1}$$

where w1, w2 and w3 are respective custom weights for distillation loss and conditional keypoint loss.

4 Experiment and Results

In our experiment, we used two teacher models for RTMPose-m architecture: i) a pre-trained model provided by MMPose and ii) a self-trained model on COCO and MPII data, respectively. Here is the performance of both sets of teacher models for reference.

Table 2: Teacher model - I: RTMPose-m (Pretrained models by MMPose)

Dataset	**Epoch**	**MPII/ PCK**	**MPII/ PCK 0.1**	**COCO/ AP**	**COCO/ AP@50**	**COCO/ AP@75**	**COCO/ AR**	**COCO/ AR@50**	**COCO/ AR@75**
COCO	420	-	-	0.746	0.899	0.817	0.795	0.935	-
MPII	210	0.907	0.348	-	-	-	-	-	-

Table 3: Teacher model - II: RTMPose-m (Custom trained model)

Dataset	**Epoch**	**MPII/ PCK**	**MPII/ PCK 0.1**	**COCO/ AP**	**COCO/ AP@50**	**COCO/ AP@75**	**COCO/ AR**	**COCO/ AR@50**	**COCO/ AR@75**
COCO	100	-	-	0.699	0.906	0.775	0.728	0.915	0.796
MPII	100	89	32.256	-	-	-	-	-	-

As mentioned in Section 3.3, a student model is optimized over a weighted loss function where custom weights are assigned for distillation loss for COCO dataset, distillation loss for MPII dataset and conditional keypoint loss. As MPII dataset is smaller is size, its distillation loss is assigned with 0.45 as weight, while COCO distillation loss is assigned with 0.25 and conditional keypoint loss with 0.45 as weight. Below mentioned are results from RTMPose-t and RTMPose-s as student models. "†" represents the model that is trained using knowledge distillation. For comparison purposes, performance of student model on COCO and MPII is provided along with its performance on combined dataset of COCO and MPII with and without knowledge distillation.

Table 4: Results with RTMPose-t as student model

Dataset	**Epoch**	**MPII/ PCK**	**MPII/ PCK 0.1**	**COCO/ AP**	**COCO/ AP@50**	**COCO/ AP@75**	**COCO/ AR**	**COCO/ AR@50**	**COCO/ AR@75**	**Teacher Model**
COCO	420	-	-	0.682	0.883	0.759	0.736	0.920	-	-
MPII	-	-	-	-	-	-	-	-	-	-
Combined	420	80.205	19.510	0.607	0.852	0.677	0.670	0.898	0.738	-
Combined †	420	79.952	19.583	0.620	0.856	0.694	0.681	0.900	0.751	I
COCO	50	-	-	0.572	0.84	0.63	0.607	0.852	0.67	-
MPII	-	-	-	-	-	-	-	-	-	-
Combined	50	78.113	16.674	0.565	0.826	0.625	0.631	0.877	0.692	-
Combined †	50	78.261	18.750	0.612	0.849	0.684	0.672	0.894	0.742	II

Table 5: Results with RTMPose-s as student model

Dataset	Epoch	MPII/ PCK	MPII/ PCK 0.1	COCO/ AP	COCO/ AP@50	COCO/ AP@75	COCO/ AR	COCO/ AR@50	COCO/ AR@75	Teacher Model
COCO	420	-	-	0.716	0.892	0.789	0.768	0.929	-	-
MPII	-	-	-	-	-	-	-	-	-	-
Combined	420	85.461	25.329	0.670	0.874	0.746	0.729	0.910	0.797	-
Combined †	420	84.155	24.657	0.686	0.883	0.757	0.742	0.920	0.807	I
COCO	50	-	-	0.661	0.885	0.733	0.693	0.89	0.758	-
MPII	-	-	-	-	-	-	-	-	-	-
Combined	50	83.039	22.227	0.635	0.860	0.712	0.695	0.904	0.767	-
Combined †	50	81.324	21.626	0.649	0.864	0.721	0.706	0.908	0.772	II

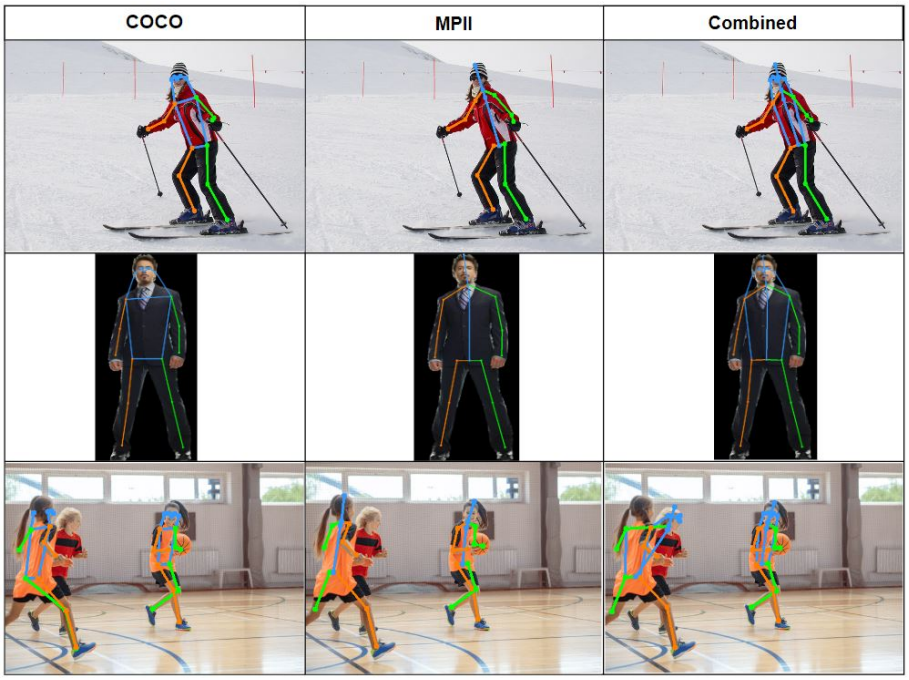

Fig. 6: Visualizations results in COCO keypoints, MPII keypoints and Combined keypoints

Here are the top findings from the above experiments on RTMPose-s and RTMPose-t:

1. As shown in table 5, RTMPose-s trained using knowledge distillation with pre-trained teacher models provided by MMPose performs the best with 84.155 PCK on MPII data and 0.686 AP on COCO data.
2. RTMPose-s and RTMPose-t architecture are not available for MPII dataset, but using our approach we obtained RTMPose-s and RTMPose-t models performing well on MPII as well as for COCO dataset.

5 Conclusion

In this work, we explored state-of-the-art human pose estimation approaches and it's available architectures. Further, we addressed the problem of unavailability of network architecture that is capable of performing well on multiple datasets. Thus, proposed a knowledge distillation based approach to train student model using set of teacher networks. Upon experimentation with RTMPose

models, we achieved student model with significant performance on MPII as well as on COCO dataset. Thus, obtaining a model architecture that is capable of performing on COCO and MPII. For further studies, this approach can be applied on other datasets to observe models efficacy.

References

1. Knowledge distillation : An overview. https://neptune.ai/blog/knowledge-distillation.
2. Mmpose:openmmlab pose estimation toolbox and benchmark. https://github.com/open-mmlab/mmpose.
3. Percentage of correct keypoints. https://oecd.ai/en/catalogue/metrics/percentage-of-correct-keypoints-%28pck%29.
4. Tensorflow ducumentation - pose estimation. https://www.tensorflow.org/lite/examples/pose_estimation/overview#:~:text=The%20pose%20estimation%20models%20takes,score%20between%200.0%20and%201.0.
5. Mykhaylo Andriluka, Leonid Pishchulin, Peter Gehler, and Bernt Schiele. 2d human pose estimation: New benchmark and state of the art analysis. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2014.
6. Yuanhao Cai, Zhicheng Wang, Zhengxiong Luo, Binyi Yin, Angang Du, Haoqian Wang, Xiangyu Zhang, Xinyu Zhou, Erjin Zhou, and Jian Sun. Learning delicate local representations for multi-person pose estimation. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part III 16*, pages 455–472. Springer, 2020.
7. Zhe Cao, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Realtime multi-person 2d pose estimation using part affinity fields. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7291–7299, 2017.
8. Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.
9. Tomáš Hodaň, Martin Sundermeyer, Bertram Drost, Yann Labbé, Eric Brachmann, Frank Michel, Carsten Rother, and Jiří Matas. Bop challenge 2020 on 6d object localization. In *Computer Vision–ECCV 2020 Workshops: Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16*, pages 577–594. Springer, 2020.
10. Xiaodong Ji, Qiaoning Yang, Xiuhui Yang, Jiahao Zheng, and Mengyan Gong. Human pose estimation: Multi-stage network based on hrnet. In *Journal of Physics: Conference Series*, volume 2400, page 012034. IOP Publishing, 2022.
11. Tao Jiang, Peng Lu, Li Zhang, Ningsheng Ma, Rui Han, Chengqi Lyu, Yining Li, and Kai Chen. Rtmpose: Real-time multi-person pose estimation based on mmpose. *arXiv preprint arXiv:2303.07399*, 2023.
12. Yanjie Li, Sen Yang, Peidong Liu, Shoukui Zhang, Yunxiao Wang, Zhicheng Wang, Wankou Yang, and Shu-Tao Xia. Simcc: A simple coordinate classification perspective for human pose estimation. In *European Conference on Computer Vision*, pages 89–106. Springer, 2022.
13. Yanjie Li, Sen Yang, Shoukui Zhang, Zhicheng Wang, Wankou Yang, Shu-Tao Xia, and Erjin Zhou. Is 2d heatmap representation even necessary for human pose estimation. *arXiv preprint arXiv:2107.03332*, 4, 2021.
14. Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pages 740–755. Springer, 2014.
15. Chengqi Lyu, Wenwei Zhang, Haian Huang, Yue Zhou, Yudong Wang, Yanyi Liu, Shilong Zhang, and Kai Chen. Rtmdet: An empirical study of designing real-time object detectors. *arXiv preprint arXiv:2212.07784*, 2022.
16. Esraa Samkari, Muhammad Arif, Manal Alghamdi, and Mohammed A Al Ghamdi. Human pose estimation using deep learning: A systematic literature review. *Machine Learning and Knowledge Extraction*, 5(4):1612–1659, 2023.
17. Jiahong Wu, He Zheng, Bo Zhao, Yixin Li, Baoming Yan, Rui Liang, Wenjia Wang, Shipei Zhou, Guosen Lin, Yanwei Fu, et al. Ai challenger: A large-scale dataset for going deeper in image understanding. *arXiv preprint arXiv:1711.06475*, 2017.