machine learning & knowledge extraction

MDPI

Systematic Review

Human Pose Estimation Using Deep Learning: A Systematic Literature Review

Esraa Samkari, Muhammad Arif *, Manal Alghamdi and Mohammed A. Al Ghamdi

College of Computer and Information Systems, Umm Al-Qura University, Makkah 21955, Saudi Arabia; s44380084@st.uqu.edu.sa (E.S.); maalghamdi@uqu.edu.sa (M.A.); maeghamdi@uqu.edu.sa (M.A.A.G.)
* Correspondence: mahamid@uqu.edu.sa

Abstract: Human Pose Estimation (HPE) is the task that aims to predict the location of human joints from images and videos. This task is used in many applications, such as sports analysis and surveillance systems. Recently, several studies have embraced deep learning to enhance the performance of HPE tasks. However, building an efficient HPE model is difficult; many challenges, like crowded scenes and occlusion, must be handled. This paper followed a systematic procedure to review different HPE models comprehensively. About 100 articles published since 2014 on HPE using deep learning were selected using several selection criteria. Both image and video data types of methods were investigated. Furthermore, both single and multiple HPE methods were reviewed. In addition, the available datasets, different loss functions used in HPE, and pretrained feature extraction models were all covered. Our analysis revealed that Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs) are the most used in HPE. Moreover, occlusion and crowd scenes remain the main problems affecting models' performance. Therefore, the paper presented various solutions to address these issues. Finally, this paper highlighted the potential opportunities for future work in this task.

Keywords: human pose estimation; 2D person pose estimation; systematic review; deep learning

Citation: Samkari, E.; Arif, M.; Alghamdi, M.; Al Ghamdi, M.A. Human Pose Estimation Using Deep Learning: A Systematic Literature Review. *Mach. Learn. Knowl. Extr.* **2023**, *5*, 1612–1659. https://doi.org/10.3390/make5040081

Academic Editor: Andreas Holzinger

Received: 23 September 2023
Revised: 4 November 2023
Accepted: 8 November 2023
Published: 13 November 2023

1. Introduction

Human Pose Estimation (HPE) is a task in computer vision that aims to locate body joints, such as the head, knees, wrists, and elbows, in an image or video frame. The goal is to connect these joints to form a skeleton shape [1,2]. Estimating human pose is important because it enables machines to interact with the human world by understanding human body poses, movements, and behaviors. HPE has a wide range of applications in many different fields. For example, in Hajj and Umrah, human poses are estimated to detect abnormal behavior when people throw pebbles or perform Tawaf [3]. Other applications of HPE include interacting with virtual worlds [4], animating game characters [5,6], tracking patient movements [7], and analyzing sports performance [8].

Depending on the requirements of the application, such as understanding human behavior or controlling robotics, the output of estimating human joints can be represented in either 2D (two-dimensional) or 3D (three-dimensional) form [9]. When given an image as input, 2D HPE locates all human joints (key points) in their X and Y coordinates. Similarly, 3D HPE performs the same task as 2D HPE but also considers the *Z*-axis (or depth information). Both 2D and 3D HPE face challenges [10]. Nevertheless, the accuracy of 3D HPE heavily relies on 2D HPE [11], as obtaining a 2D location for each human joint is the first step in 3D HPE. In addition, there are a lot of 2D HPE methods that aim to propose an accurate way to estimate the person's pose with less computational complexity. This review will focus only on 2D HPE to comprehensively cover and analyze these methods.

Developing an accurate and robust 2D HPE technique is computationally complex due to the challenges posed by pose estimation [9]. When a method is proposed to estimate

human pose in an image or video, it must account for variations in body size. For example, some bodies may be farther from the camera, making certain body parts almost invisible and difficult to estimate. Additionally, a 2D HPE technique must identify which body parts belong to which person when people overlap in an image. Other factors that increase the challenges of 2D HPE include different environments with varying light contrast, interaction with objects, and an unknown number of poses in an image [12].

Two approaches for estimating human pose are traditional computer vision-based and deep learning-based methods [2], as shown in Figure 1. Traditional methods involve designing handcrafted features and models to detect and locate body joints [13]. The features in these conventional methods are often manually crafted and require experts to develop features that handle diverse human poses. Figure 2 shows some of these features. The techniques used in traditional methods, such as pictorial structure [14], incorporate prior knowledge about the human body and its spatial relationships between body parts. These techniques can estimate the human body with lower computational requirements. However, they fail to capture complex variations in pose or occluded, nonvisible, or overlapped joints. Additionally, estimating more than one pose using the traditional approach was slow and challenging [15].

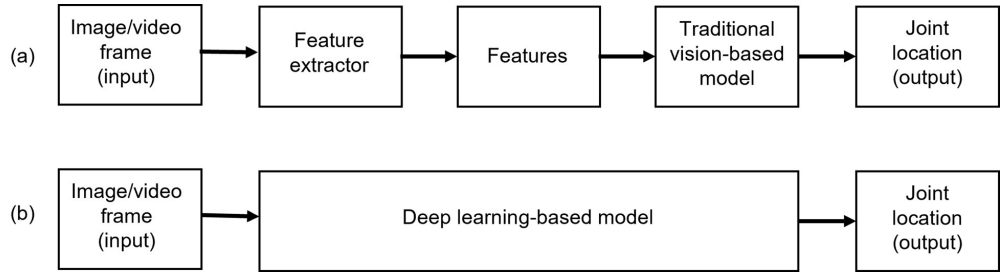


Figure 1. Two flows to estimate the human pose. (**a**) The traditional vision-based flow uses handcraft features, and then the vision-based model estimates the human parts. (**b**) The deep learning flow learns the features automatically, and then the model predicts the human pose.

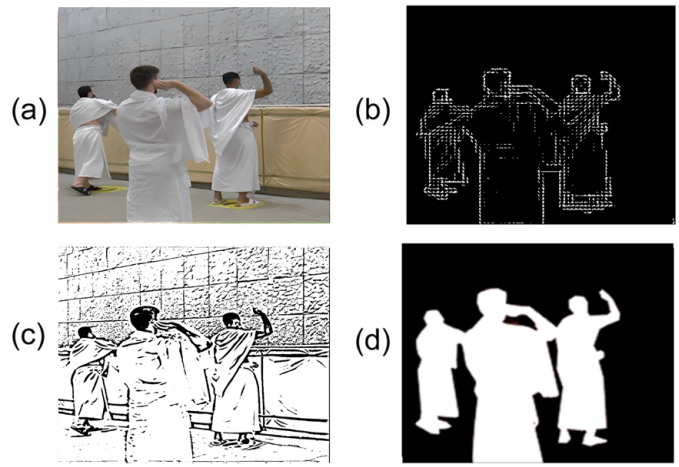


Figure 2. Examples of handcrafted features used in traditional vision-based methods for human pose estimation are (**a**) the original image, (**b**) the HOG (Histogram of Oriented Gradients) feature, (**c**) the edges feature, and (**d**) the silhouettes feature.

In contrast, deep learning-based methods have made significant progress in HPE due to their ability to extract features from data implicitly [13,16]. In other words, there is no need to extract specific features as in traditional methods. Additionally, the availability of large, annotated datasets, developments in computational hardware, and, most notably, the use of deep convolutional networks (ConvNets) in current HPE algorithms have led to a significant boost in performance [9]. Deep learning-based methods can handle most

HPE challenges, unlike vision-based methods, such as estimating nonvisible key points. The first study to convert the HPE process from traditional to deep learning-based was the DeepPose model [17]. This model uses several ConvNets layers to regress the body joint location from a single image.

Recently, most deep learning-based methods [18–20] use pretrained models that implicitly extract human body features while focusing on designing network structures that predict keypoint coordinates. Different techniques (e.g., multistages and branches) and different feature levels (global and local) are used in building the model. As a result, the accuracy of the model has increased. However, the model's size also increases, leading to an increase in computational complexity. Therefore, some methods attempt to build a network structure that balances the model's accuracy and size. More detailed information about deep learning-based methods is covered in this review.

Extracting human shape data from an image or video required significant effort, as most methods were vision-based. Some surveys [21,22] covered pose estimation and motion capture tracking methods based on traditional computer vision techniques, while other surveys [15,23] discussed more advanced topics of conventional methods. These surveys discussed three main components: features extraction (e.g., silhouette, SIFT, and HOG), appearance models (models used to detect human parts, such as SVM and AdaBoost classifiers), and structure models (models used to create a relationship between human parts, such as tree structure model). In addition, they also provided the limitations of traditional methods in estimating the human pose.

With the emergence of neural networks, surveys [2,24] also included deep learning-based techniques with vision-based methods. Gong et al. [2] comprehensively summarized HPE from monocular images, including traditional and early deep-learning works. Their survey covered feature extraction, the type of human body model used, and pose estimation methods. Additionally, they provided a summary of existing datasets and performance metrics used. However, only a few early deep-learning methods were covered. In contrast, a more recent survey [24] covered more indepth works that have significant contributions in estimating two or more persons' poses in an image, such as OpenPose [25], DeepCut [26], and Mask R-CNN [27], while highlighting their challenges and providing a brief description of traditional methodologies.

As deep learning techniques improved, many 2D HPE approaches were proposed. Several surveys collected and summarized these approaches [28–30]. Dang et al. [28] collected papers on 2D pose estimation and classified them based on the number of people in the image: single or multiple. They also included available datasets and a summary of metrics used. Similarly, Song et al. [29] followed the same taxonomies. However, they did not mention evaluation metrics.

Rather than grouping different 2D HPE methodologies based on the number of people in the image, Munea et al. [30] explained the architectures of 2D human pose estimation models. For each architecture, they provided details such as the number of layers, the types of layers used, their order, the techniques used, and the names of datasets used for training. In addition to covering datasets and metrics, they also discussed other main components of 2D pose estimation, including loss functions and pretrained feature extraction models.

Recently, the number of published surveys concerned about HPE has increased [9,31–35]. Like other surveys, the taxonomies for HPE in these surveys are the same. Methodologies are classified based on the number of people in the input data [9,31,34]. Each classification has a subcategory: regression and heatmap for single-pose estimation and top-down and bottom-up for multipose estimation. Most works that have made significant progress in HPE are included under each subcategory. However, the surveys by Toshpulatov et al. [32] and Liu et al. [33] grouped these works based on similar tasks performed by deep learning models. These tasks include structural modeling, refinement, multistage processing, and multitasking.

The above surveys focus on images as input data, while only a few are interested in video processing [33–35]. To our knowledge, only two surveys [33,36] have covered

multipose video-based estimation. Their classification of methodologies differs, and ref. [36] summarized fewer than ten works. Therefore, in addition to covering pose estimation using a single image, we will also cover single and multipose video-based methodologies in depth. Furthermore, since many surveys do not cover essential sections such as pretrained feature extraction models and loss functions, we will include them here.

Although most existing surveys list the datasets used for HPE, some do not provide sample data. Therefore, we will provide sample images for each dataset and represent their keypoint annotations graphically. Additionally, only a few surveys follow a systematic review procedure but have a limited scope, focusing on physical exercise [37,38] or gait identification [39]. Collecting papers on 2D HPE and performing a systematic literature review produces valuable contributions. Table 1 summarizes the differences between each survey mentioned and our review paper.

Table 1. Summary of existing surveys of human pose estimation. The * symbol indicates that the section is partially covered, and FE means the existing pretrain feature extraction models. "Loss" shows the types of loss functions used for posing estimation tasks, and "Metric" shows the kinds of metrics used to measure the performance of the pose estimation model.

Ref	Year	Journal	Dataset	Loss	Metric	FE	Single Pose		Multi Pose		Conventional/ Systematic
							Image	Video	Image	Video	
[2]	2016	Sensors	✔	✘	✔	✘	✔	*	✔	✘	Conventional
[28]	2019	TST	✔	✘	✔	✘	✔	✘	✔	✘	Conventional
[31]	2020	CVIU	✔	✘	✔	✘	✔	*	✔	✘	Conventional
[30]	2020	IEEE access	✔	✔	✔	✔	✔	✘	✔	✘	Conventional
[37]	2021	Sensors	✔	✘	✔	✘	*	✘	*	✘	Systematic
[38]	2021	ACM Com. Surv.	✘	✘	✘	✘	*	*	*	✘	Systematic
[9]	2021	IVC	✔	✘	✔	✘	✔	*	✔	✘	Conventional
[29]	2021	JVCIR	✔	✘	✘	✘	✔	✘	✔	✘	Conventional
[36]	2021	Patt. Recog.	✔	✘	✔	✘	✘	✘	✔	✔	Conventional
[32]	2022	JS	✔	✘	✔	✘	✔	✘	✔	✘	Conventional
[39]	2022	ACM Com. Surv.	✔	✘	✔	✘	✔	✘	✔	✘	Systematic
[33]	2022	ACM Com. Surv.	✔	✘	✔	✘	✔	✔	✔	✔	Conventional
[35]	2023	IEEE THMS	✔	✘	✔	✘	✔	✔	✔	*	Conventional
[24]	2023	MMS	✔	✘	✔	✘	✔	✘	✔	✘	Conventional
[34]	2023	ACM Com. Surv.	✔	✘	✔	✘	✔	✔	✔	✘	Conventional
our	2023	MAKE	✔	✔	✔	✔	✔	✔	✔	✔	Systematic

Many surveys have focused on collecting, presenting, and analyzing the structures of different models. Similarly, we discuss how skeletons are generated from images and video frames to provide readers interested in HPE with the latest advancements in pose estimation. Formulating research questions will help us gather relevant articles and information about HPE. This information may include the available datasets, loss functions, and evaluation metrics used to measure the model performance and the existing methodologies. Research questions with their purposes are shown in Table 2.

Due to the significant role that 2D plays in affecting the performance of 3D HPE, we have limited our focus to 2D HPE methodologies. This survey follows a systematic search procedure to gather these methods. Hence, our contributions are as follows:

- We have systematically collected up-to-date human pose estimation models for image- and video-based input from 2014 to 2023;
- We classified pose estimation methods based on input type (image-based or video-based) and the number of people (single or multiple);

- We also provide an overview of existing datasets for estimating human poses and a list of loss functions, evaluation metrics, and commonly used feature extraction models.

From now on, the remainder of the paper is organized as follows: The next section describes the methodology of our research protocol, and the results of the search are in Section 3. Sections 4–6 illustrate the existing datasets, the loss functions with evaluation metrics, and available pretrained feature extraction models, respectively, that are used as the main component of any human pose estimation model. Section 7 presents different single-pose and multipose estimation techniques in images and videos. The discussion of our survey is presented in Section 8, while future directions are discussed in Section 9. Finally, Section 10 provides the conclusion of our review.

Table 2. Research questions of our review.

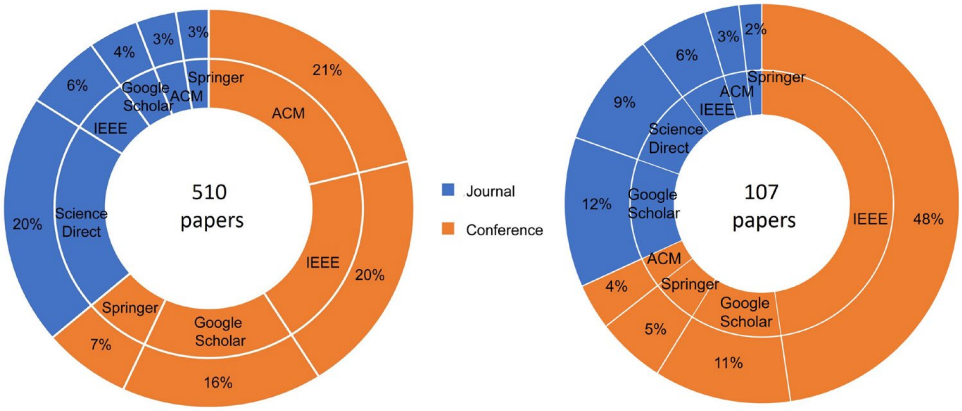
	Question	Purpose
RQ_1	Which datasets are used to analyze the performance of the deep learning methods?	Discover the quality and other criteria of the datasets used to train the model of human pose estimation.
RQ_2	Which loss functions and evaluation criteria are used to measure the performance of deep learning methods in human pose estimation?	Discover the loss functions and evaluation metrics used in human pose estimation to measure model performance in training and testing mode.
RQ_3	What pretrained models are used for extracting the features of the human pose?	Compares the available models utilized to extract the features and knows the criteria for selecting one of these models.
RQ_4	What are the existing deep learning methods applied for 2D human pose estimation?	Review different methods that estimate the human pose in images and videos to know the new trend methods in this field.

2. Methodology

Our review seeks to collect and organize 2D HPE papers that use deep learning techniques. Following systematic search procedures, 2D HPE-related papers were collected and filtered based on selecting search terms and databases, establishing exclusion criteria, and evaluating the papers' content quality. Each of these stages will be described in detail in this section.

2.1. Search Process

We chose several databases to gather papers from: IEEE Explore, Science Direct, Google Scholar, Springer, and ACM Digital Library. For each database, we entered the following search query to collect papers: [("human pose estimation" OR "pose estimation") AND (2D AND NOT 3D) AND "deep learning"]. The search date was set between January 2014 and October 2023. The articles collected from this step were added to the pool of candidate papers, which reached a total of 510. As shown in Figure 3, most of these articles were from conferences, accounting for 64% of the total, while the remaining initial articles were from journals, accounting for 36%. The following sections will show the criteria by which these candidate papers were reduced from 510 to 107, where the proportion of conference publications is 68% and the journals 32%.



(a) (b)

Figure 3. Distribution of publications of journals and conferences, where (**a**) is the distribution of the initial search result, and (**b**) is the distribution of the papers after applying the filtering process.

2.2. Exclusion Criteria

We used several criteria to exclude papers from the candidate set. The exclusion criteria (EC) are as follows:

- EC1: Studies must be peer-reviewed articles published in English;
- EC2: We do not include books, notes, theses, letters, or patents;
- EC3: Only papers that focus on applying deep learning methods to the problem of 2D human pose estimation are included;
- EC4: Unique contributions are considered for inclusion; repeated studies are not included;
- EC5: Papers that estimate only a part of the human pose, such as the head or hand, are omitted;
- EC6: Articles with multiple versions are not included; only one version will be included;
- EC7: Papers found in more than one database (e.g., in both Google Scholar and IEEE) are not included; only one will be included.

2.3. Quality Assessment

Defining quality assessment criteria will help ensure that the selection of candidate papers is fully transparent and objective, without bias. Table 3 shows the quality assessment criteria. This table has three columns: criteria description, potential value, and maximum weight for each assessment criterion. The description column lists the quality criteria for filtering the initial papers. The abstract, methodology, evaluation, result, and conclusion sections have their quality criteria, and each paper in the candidate set must pass this assessment. Each section has a weight value classified according to the type of answer. If the answer requires a yes or no response, the value type is considered binary, and the weight is either 0 or 1. However, assigning a binary value to some sections that contain details, such as methodology, evaluation, and results, would be unfair. Therefore, the value assigned to these sections ranges from 0 (no details mentioned) to the maximum value (all aspects of the section are covered without missing any information).

Table 3. Article quality assessment. The parenthesis () in the "Value" column means the value could be 0 or 1, whereas [] square brackets indicate the value takes a float number. The results are out of 7.

	Description	Value	Max Weight
QA_1	Presenting a transparent and fair explanation of the problem, the approach, and the results in the abstract.	(0,1)	1
QA_2	Presence of a visual representation of the proposed method. In addition, the steps of the process must be described in detail.	[0–2]	2
QA_3	Gives information about the dataset and measures utilized in the model's evaluation.	[0–1]	1
QA_4	Interpreting the findings cautiously, considering the study's aims, limitations, the number of analyses conducted, and related research findings.	[0–2]	2
QA_5	Mentioning the study's drawbacks and the limitations in the conclusion section.	(0,1)	1

3. Search Results

The initial search results, the selection criteria, and the qualitative evaluation of the articles chosen for inclusion are discussed here. Figure 4 summarizes the number of articles that passed the filtering process and shows how many were eliminated in each phase. Answering and analyzing our review questions will be in the discussion section, Section 8.

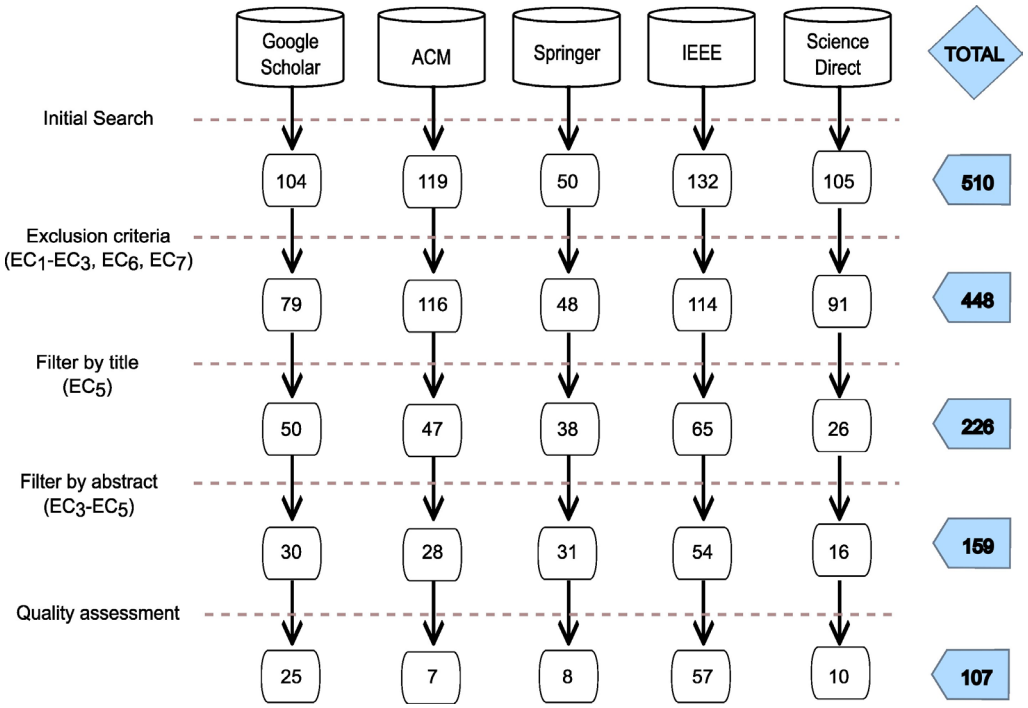


Figure 4. Database-filtered articles. EC means the exclusion criteria (see Section 2.2).

3.1. Exclusion of Articles from the Initial Search

The paper selection protocol was covered in Section 2, including everything from the initial search through the content analysis. Figure 4 displays the sum of all 510 articles collected across various databases during the initial search phase. After that, we eliminated duplicate articles, works not papers, e.g., books and theses, articles not written in English, and articles that used techniques other than deep learning, which left 448 unique articles. The nonrelated articles, such as hand-foot pose estimation, were excluded, resulting in 226, then reduced to 159 articles for each filter phase. Before applying the last quality assessment

filter, we distributed these articles interested in human pose estimation, as Figure 5 shows, to see the trend of HPE based on the input type. Lastly, the quality assessments, which will be discussed next, helped select the high-quality articles, resulting in 107 remaining articles.

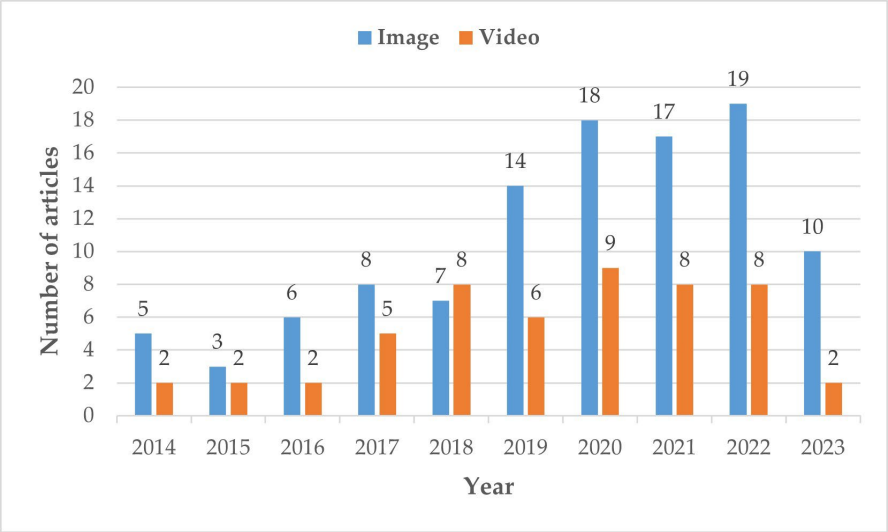


Figure 5. The distribution shows 159 articles that pass the exclusion filter. These articles were distributed according to the type of input data: image and video.

3.2. Result of Quality Assessments

This section covers the quality assessment outcome to choose relevant articles. Based on quality criteria, we defined them in Section 2.3. Failure to provide information according to quality assessment criteria results in a lower paper evaluation score. Papers that scored 4.5 or less out of 7 are eliminated from the candidate set. Therefore, only papers of high quality are considered. Figure 6 shows the papers that failed and passed these criteria. Only 107 out of 159 articles passed the quality assessment criteria and Figure 7 shows the distribution of these articles that are included in this review.

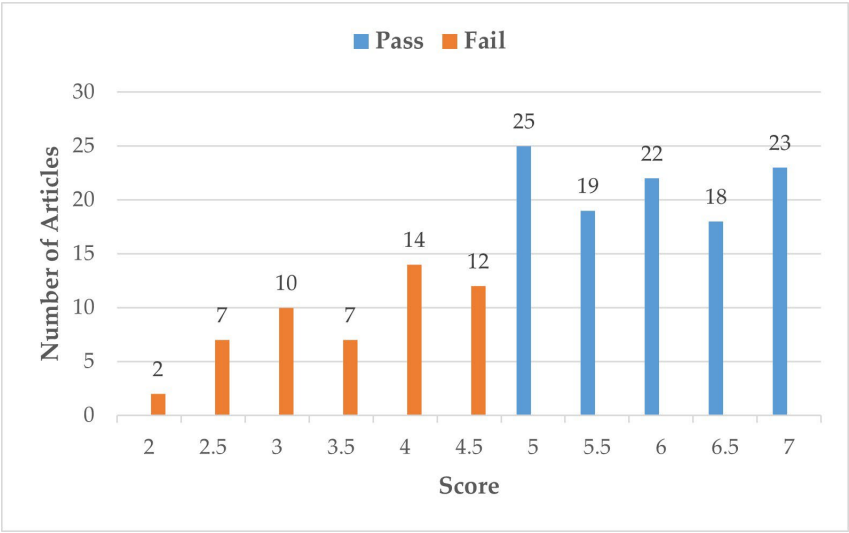


Figure 6. Plot of the quality assessment criteria. A total of 107 out of 159 articles passed the quality assessment filter.

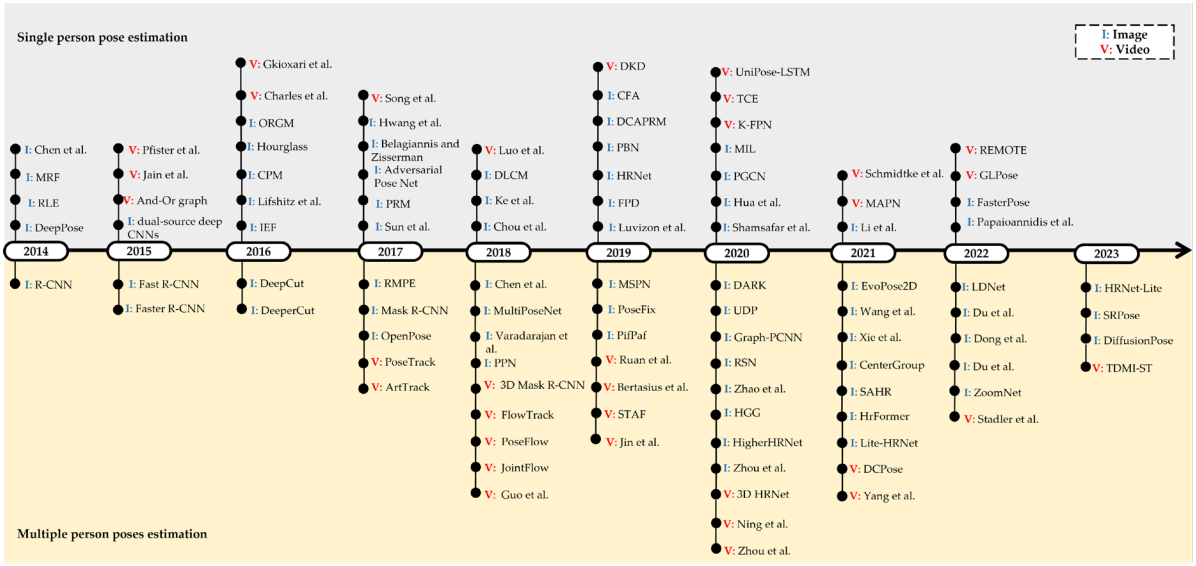


Figure 7. The distribution of 2D human pose estimation methods that are included in this review.

There are several reasons why some papers fail to pass in the quality assessment phase. One of them is that some studies did not mention the results in the abstract section. As a result, the assessment score is impacted. The other reason is the poor analysis, in which some studies compared their proposed methodology with old or fewer methods. Although most of the studies thoroughly explained their methodology, some did not mention the drawbacks and limitations of the methods, which also affected the assessment score.

4. Datasets of Human Pose Estimation

Deep learning models require large amounts of data to perform specific tasks accurately. Human Pose Estimation (HPE) models, in particular, require diverse data to handle challenges such as varying backgrounds, illumination, and clothing. Fortunately, existing datasets address these challenges by offering a diverse recording environment. Some datasets also provide different pose activities [40,41], such as daily life activities (e.g., standing, walking, sitting) and sports activities that contain complex poses and occlusion problems.

Datasets play a crucial role in training and testing 2D HPE models. HPE datasets typically consist of images, videos, or both that capture human subjects in various poses and positions. They are annotated with keypoint locations (i.e., the positions of body joints). In addition to keypoint locations, other annotations may also be provided, such as pose ID, joint visibility, and activity name. These annotations help address HPE challenges such as occlusion, tracking multiple poses, and handling complex poses.

The most well-known datasets used for 2D pose estimation that are available for access are LSP, FLIC, MPII, COCO, and CrowdPose. These datasets are commonly used for estimating single/multiple poses in images, while PennAction, JHMDB, and PoseTrack are used to estimate poses from videos. All of these datasets will be discussed in this section. Table 4 shows a comparison between the mentioned datasets.

Table 4. Publicly available datasets of 2D human pose estimation. The letter "i" indicates images, and the letter "v" means videos. Joints, size, and person instance indicate the number of annotated joints for each person, the amount of data in the dataset, and the number of people poses with annotation, respectively.

Dataset	Year	Single Pose		Multi Pose		Joints	Size	Person Instance	Source
		Image	Video	Image	Video				
LSP [42]	2010	✔	✘	✘	✘	14	2 K	-	Flickr
LSPE [43]	2011	✔	✘	✘	✘	14	10 K	-	Flickr
FLIC [44]	2013	✔	✘	✔	✘	10	5 K	-	Movies
PennAction [45]	2013	✘	✔	✘	✘	13	2.3 K	-	-
JHMDB [46]	2013	✘	✔	✘	✘	15	900	-	Internet
MPII [40]	2014	✔	✔	✔	✔	16	i = 25 K v = 5.5 K	40 K	YouTube
COCO [41]	2017	✔	✘	✔	✘	17	200 K	250 K	Internet
PoseTrack17 [47]	2017	✘	✔	✘	✔	15	550	80 K	Internet
PoseTrack18 [47]	2018	✘	✔	✘	✔	15	1 K	144 K	Internet
CrowdPose [48]	2019	✔	✘	✔	✘	14	20 K	80 K	Three benchmarks
PoseTrack21 [49]	2022	✘	✔	✘	✔	15	1 K	177 K	Internet

4.1. Leeds Sports Pose (LSP) and LSP Extended (LSPE)

The LSP dataset [42] is used for single-pose estimation. It has 1000 training images and 1000 testing images. LSP built this collection of images based on sports-related tags on Flickr, such as athletics, baseball, soccer, and tennis. As a result, most body poses in the images have complex poses, as shown in Figure 8. The dataset labels have 14 key points. It is worth noting that the data have been expanded [43] to include 10,000 training images, known as the LSP-Extended dataset.

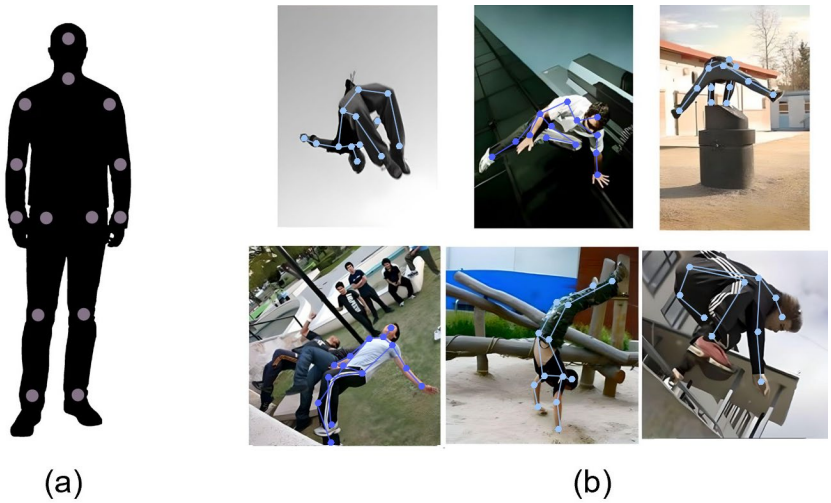


Figure 8. The LSP dataset, where (**a**) is the key points annotated of the dataset and (**b**) are some data examples.

4.2. Frames Labeled in Cinema (FLIC)

The FLIC dataset [44] is used for single and multipose estimation. It is split into training and testing datasets containing around 4000 and 1000 images. Annotations include ten key points. Images were collected from movies [30] depicting a distinct pose and clothing. Figure 9 shows examples from the FLIC dataset.

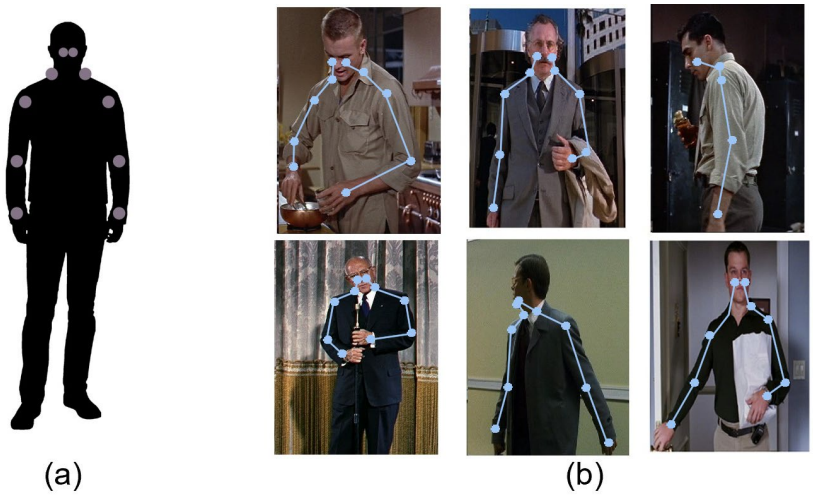


Figure 9. The FLIC dataset, where (**a**) is the key points annotated of the dataset and (**b**) are some data examples.

4.3. PennAction

The PennAction dataset [45] contains over 2000 videos with annotated frames, totaling 330,000 frames with an average of 70 frames per video. All video frames are in RGB color. There are 13 annotated key points, as shown in Figure 10. Around 1200 videos are used for training and 1000 for testing. The Penn dataset includes 15 actions such as pushups, strumming guitars, baseball, jumping rope, and tennis serve. Keypoint visibility, the four directions from which the point of view can be seen, and the bounding box of the person are also labeled.

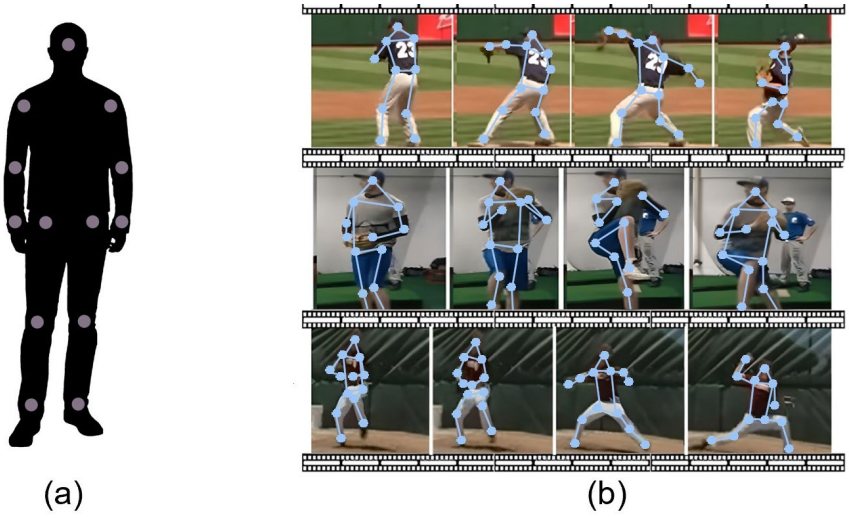


Figure 10. The PennAction dataset, where (**a**) is the key points annotated of the dataset and (**b**) are some data examples.

4.4. Joint Human Motion DataBase (JHMDB)

The JHMDB dataset contains complete annotations of human actions [46], such as brushing hair, catching, clapping, and climbing stairs. It is used for human action recognition and human pose estimation. Each person in this subset has 15 annotated joints and around 900 video clips. This dataset annotates human actions in each frame, including scale, segmentation, pose, and optical flow. Figure 11 shows samples of the dataset and keypoint annotations.

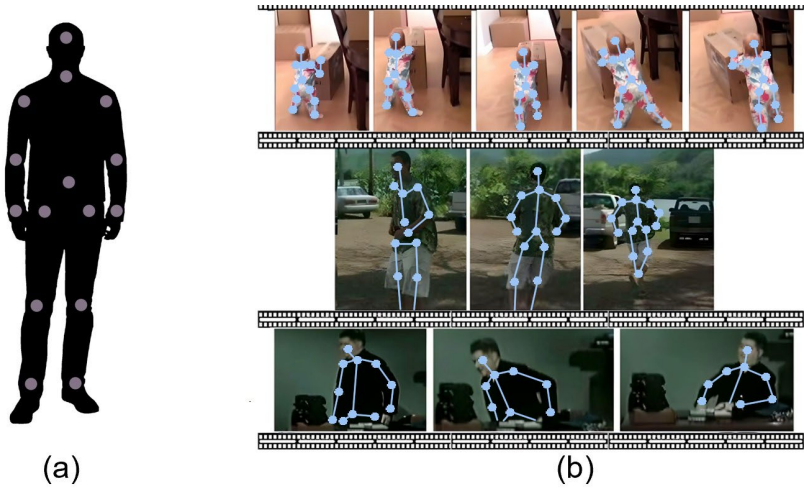


Figure 11. The JHMDB dataset, where (**a**) is the key points annotated of the dataset and (**b**) are some data examples.

4.5. Max Planck Institute for Informatics (MPII)

The MPII dataset is used for single and multipose estimation and contains images and video frames [40]. It has approximately 25,000 images depicting 410 different activities. The dataset comprises around 22,000 training images, 3000 validation images, and 7000 testing images. Annotations represent 16 joints of the human body. Images were collected from YouTube videos, providing different scale variations and complex poses. The dataset also contains several annotated video clips. Figure 12 shows some examples from the MPII dataset.

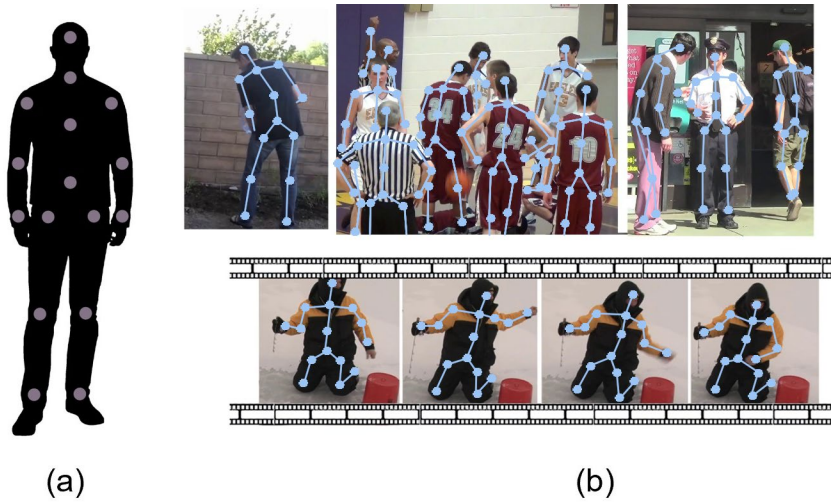


Figure 12. The MPII dataset, where (**a**) is the key points annotated of the dataset and (**b**) are some data examples.

4.6. Common Objects in Context (COCO)

The COCO dataset [41] contains object detection, segmentation, and keypoint detection annotations. Amazon's Mechanical Turk workers gather and annotate images from the internet. The dataset contains over 200,000 images and 250,000 human instances. It was released in 2014 and updated in 2017, splitting the training/validation images from 83,000/41,000 to 118,000/5000. The test set contains 20,000 images with annotations. Each 2D pose has 17 body joint annotations. The COCO dataset includes various human poses and objects of varying sizes and occlusion patterns, as shown in Figure 13.

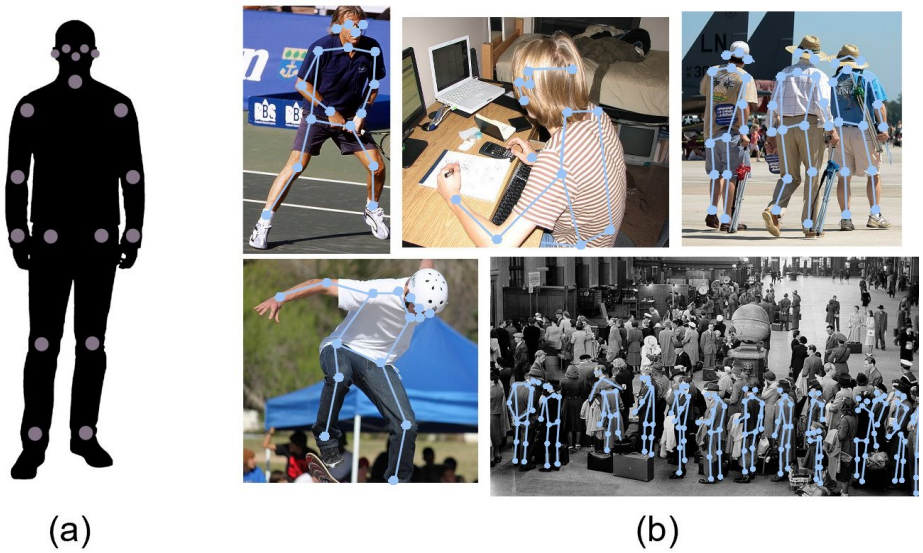


Figure 13. The COCO dataset, where (**a**) is the key points annotated of the dataset and (**b**) are some data examples.

4.7. PoseTrack

The PoseTrack dataset [47] is widely used to train models for estimating and tracking multiperson poses. This dataset contains challenging scenarios involving highly occluded individuals in crowded environments with complex movements. There are two versions of the PoseTrack dataset: PoseTrack 2017 and PoseTrack 2018. PoseTrack 2017 contains 550 videos, while PoseTrack 2018 is an extended version with approximately 1100 videos. Both versions include three sets (train, validation, and testing) and have 15 labeled joints, a person ID, and joint visibility annotations as labels. Not all video frames are annotated; only the middle 30 and a few others are annotated. However, Döring et al. [49] have recently extended pose annotations of the PoseTrack2018 dataset. This version of the extension is known as PoseTrack21. The new version includes the small persons' annotation in the crowded scene, where out of 177,164 pose annotations have been included. Figure 14 shows samples of data from the PoseTrack dataset.

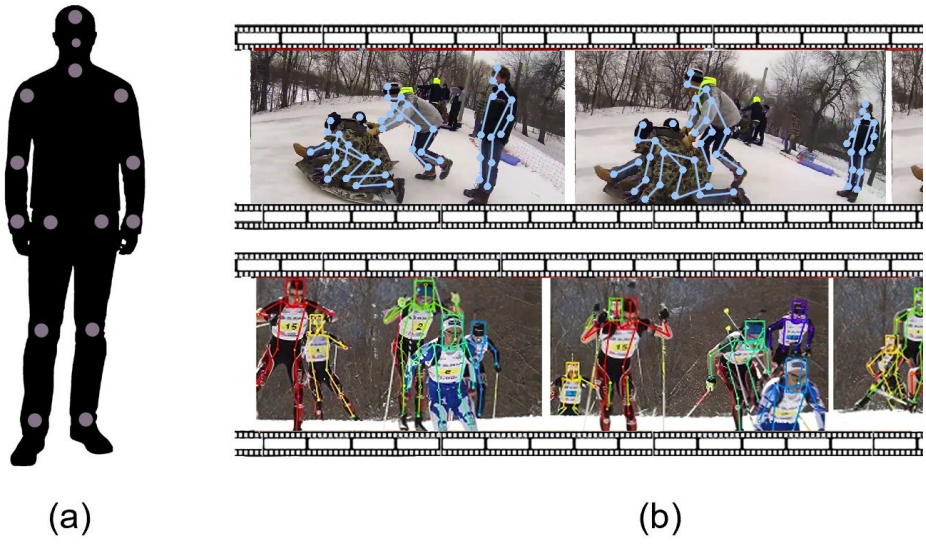


Figure 14. The PoseTrack dataset, where (**a**) is the key points annotated of the dataset and (**b**) are some data examples.

4.8. CrowdPose

The CrowdPose dataset is suitable for training models that aim to estimate single poses. However, this dataset focuses more on multiple poses and crowded scenarios [48]. It contains 10,000 training images, 2000 validation images, and 8000 testing images. Additionally, the CrowdPose dataset provides various joint annotations based on the accuracy of human capture. The source of CrowdPose images is based on three well-known datasets: COCO, MPII, and AI Challenger. Figure 15 shows some crowded scenarios in the dataset.

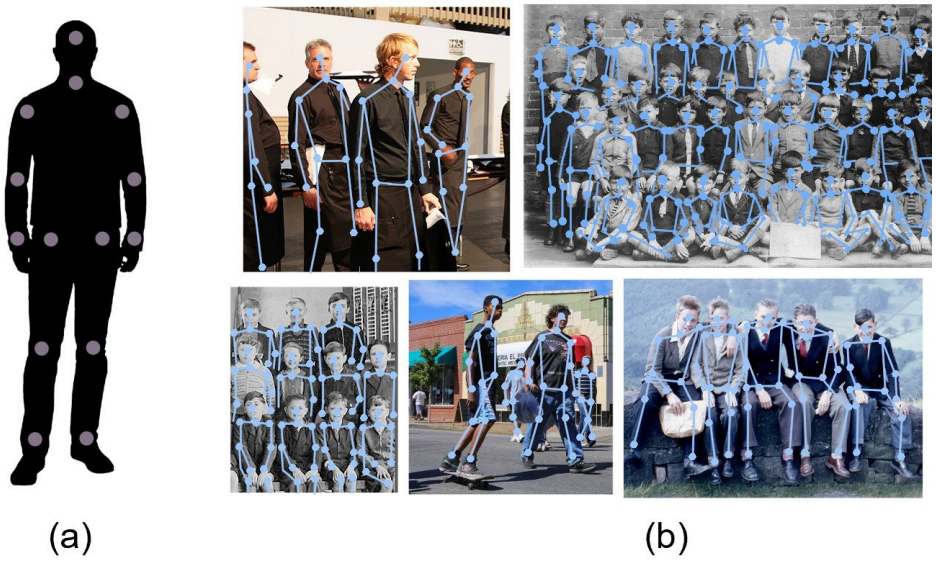


Figure 15. The CrowdPose dataset, where (**a**) is the key points annotated of the dataset and (**b**) are some data examples.

5. Loss Functions and Evaluation Metrics

Loss functions and evaluation metrics are essential to measure the model's performance. The loss function is used for training the model, while the metric is used for evaluating the model. This section will show the most loss functions and metrics available for person pose estimation.

5.1. Loss Function

A loss function is a mathematical function that trains a model by updating its parameters. The choice of an appropriate loss function significantly impacts the model's performance. In 2D HPE, selecting a suitable loss function depends on several factors, including the type of task (regression or classification), the model architecture, dataset size, and the presence of occlusions. The most commonly used loss functions for regression tasks in 2D HPE [17,50–53] are L_1 and L_2. Log loss is frequently used for classification tasks [26,54,55]. Other loss functions, such as structure-aware [56], composite [57], focal [58], auxiliary [59], and knowledge distillation [50], are also used. This section provides more detail about the most common and some other loss functions used for evaluating 2D HPE models.

5.1.1. Cross-Entropy

Cross-entropy loss, also known as negative log-likelihood or log loss, is commonly used in classification problems. For example, in 2D HPE, Zhou et al. [60] used this loss function to classify the predicted occlusion status of human joints. The value of the loss ranges from 0 to 1, with values closer to 0 being preferable. Cross-entropy loss is based on logarithmic functions and is calculated using the following equation:

$$CrossEntropy_{loss} = -\sum_{i=1}^{n} y_i \log(\hat{y}_i), \quad (1)$$

where n indicates the number of human joints, y_i is the true label of the ith joint visible, and $\hat{y}_i$ is the predicted label.

5.1.2. Focal

Focal loss is a variant of binary cross-entropy loss used in computer vision tasks such as object detection and pose estimation [61]. It was proposed to address the issue of class imbalance, where the majority of samples belong to the negative class. For example, Braso et al. [58] used focal loss to classify whether the predicted location of the center keypoint (used for the grouping method in a bottom-up approach) has a corresponding label in the ground truth or not. Focal loss is defined as follows:

$$Focal_{loss}(p_t) = -(1 - p_t)^{\gamma} \, log(p_t), \tag{2}$$

where p_t is the predicted probability of the positive class, i.e., the keypoint is present, and γ is a parameter that controls the degree of focus on hard examples and misclassified samples with high confidence.

5.1.3. Mean Absolute Error

Mean Absolute Error (MAE), or L_1 loss, measures the absolute differences between the ground truth and predicted values. It does not consider the direction of the joint [30], only measuring the magnitude of the error. The following equation gives the L_1 loss:

$$L_1 = \frac{1}{n} \sum_{i=1}^{n} |y_i - \hat{y}_i|, \tag{3}$$

where n is the number of person joints, y_i and $\hat{y}_i$ are the true and predicted values of the ith joint location, respectively. This loss is helpful when data outliers are present, as it is less affected by outliers compared to the L_2 loss.

5.1.4. Mean Squared Error

Mean Squared Error (MSE), also known as L_2 loss, is commonly used in 2D human pose estimation [51,62]. Like L_1 loss, it measures only the magnitude of the error. L_2 calculates the mean squared difference between predicted and ground truth joint positions, as shown in the following equation.:

$$L_2 = \frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2, \tag{4}$$

where n is the number of joints, y_i is the true value of the current ith joint location, and $\hat{y}_i$ is the predicted value of the current ith joint location. Unlike L_1, L_2 is more sensitive to data outliers, which may result in a less robust model.

5.1.5. Auxiliary

Auxiliary loss in HPE is a technique used to improve the training of deep neural networks [59] by introducing additional loss terms that help guide the optimization process. The main idea behind this loss function is to add additional outputs to the trained model to enhance its performance. It addresses the vanishing gradients in deep neural network models and prevents this by adding intermediate loss functions.

5.1.6. Knowledge Distillation

Knowledge distillation is a technique used to improve the performance and efficiency of neural networks in computer vision tasks, including human pose estimation. The idea is to train a student model (simpler model) to learn from a teacher model (complex model) by mimicking its output [50]. In the context of HPE, the teacher model could be a state-of-the-art model that produces accurate 2D joint locations. In contrast, the student

model could be faster and more lightweight, creating similar joint locations but with lower computational requirements.

5.2. Evaluation Metrics

Unlike loss functions, metrics are not used to update a model's parameters. Instead, they provide a quantitative measure of the performance of the trained model. Metrics allow researchers to compare different models and select the most effective one for their application.

Once human poses have been predicted, the predicted 2D joint locations are compared with the ground truth annotations in the dataset. Evaluation metrics are then calculated based on the difference between the predicted and ground truth joint locations. A model is considered to have high performance when it meets certain thresholds. These thresholds vary from metric to metric.

Several metrics are used to evaluate the performance of HPE models. AP, AR, and OKS are the most popular metrics for evaluating 2D multipose models [34,63], while PCP, PCK, and PDJ evaluate single-pose models [64,65]. This section provides detailed information about these metrics, while Table 5 summarizes them and shows their thresholds.

Table 5. Commonly existing metrics of 2D human pose estimation. "Target" could be either "Limbs" like arm, or "Joints" like hand. The "Threshold" column shows how to calculate the limit value the predicted value of the model must pass.

Metric	Variation	Target	Threshold
PCP	PCP@0.5	Limbs	Limb's truth value × 0.5
PCK	PCKh	Joints	Joint bounding box or head length × 0.5
AUC	-	Joints	Different PCK thresholds
PDJ	PDJ@0.2	Joints	Torso diameter × 0.2
IOU	-	Joints	Joint's bounding box
OKS	OKS@0.5 OKS@0.95	Joints	Close to the ground-truth joint
AP	mAP AP^{50} AP^{75} AP^{M} AP^{L}	Joints	Various OKS thresholds, including primary metric (OKS = 0.5:0.05:0.95), loose metric (OKS = 0.5), and strict metric (OKS = 0.75)
AR	AR^{50} AR^{75} AR^{M} AR^{L}	Joints	Same as AP, it uses various OKS thresholds

5.2.1. Percentage of Correct Parts

The higher the Percentage of Correct Parts (PCP) value, the better the human pose estimation model. This type of metric was commonly used in earlier works [66]. PCP determines the rate at which body parts (limbs) are detected. The rate value is high if the distance between the two predicted endpoints and the ground truth endpoints is less than 50 percent of the body part length [30]. Otherwise, the rate of detecting limbs is low.

5.2.2. Percentage of Correct Key points

Similar to PCP, a higher Percentage of Correct Key points (PCK) value indicates better model performance. This metric measures the accuracy of predicting human body joints within a certain threshold [31]. One such threshold is half the head segment length, denoted as PCKh@0.5. Another threshold is a fraction of the size of the person's bounding box.

5.2.3. Area under the Curve

The Area Under the Curve (AUC) metric measures object detection performance [67,68]. This metric uses a variant of PCK thresholds and analyzes different pose estimation algorithms in depth.

5.2.4. Percentage of Detected Joints

Like the PCK metric, the Percentage of Detected Joints (PDJ) metric follows a similar rule. However, it was introduced to address the problem of short limbs created using the same error threshold as PCP [17]. If a predicted joint is within a certain fraction of the torso length, its location is considered correctly located.

5.2.5. Intersection over Union

Intersection over Union (IoU) [67] measures the intersection of the predicted bounding box region with the ground truth bounding box. It determines the degree of similarity between two sets, as indicated by the equation below.

$$IoU = \frac{Area\ of\ Intersection}{\text{Area of Union}}, \tag{5}$$

As a result, *IoU* is also used to measure the accuracy of an object detector. *IoU* values range between 0 and 1. The accuracy of the object detector is higher if the *IoU* is closer to 1.

5.2.6. Object Keypoint Similarity

Object Keypoint Similarity (OKS) measures the accuracy of a predicted joint by finding the distance between the predicted joint and the ground truth joint. Both OKS and IoU metrics share the same task of object detection. The equation for OKS [63] is as follows:

$$OKS = \frac{\sum_i exp\left(-d_i^2/2s^2\,k_i^2\right)\,\delta(v_i > 0)}{\sum_i \delta(v_i > 0)}, \tag{6}$$

where v_i shows whether the joint is visible or not on the ground truth, d_i is the Euclidean distance between the estimated joint and the joint of the ground truth, s is the human scale, and k_i is a way to control falloff at each joint.

5.2.7. Average Recall and Average Precision

Average Recall (AR) and Average Precision (AP) metrics are the most commonly used for estimating multiple poses in many datasets, such as the MS-COCO dataset. Both metrics depend on OKS. Table 5 shows several forms of AP and AR. AP Across Scales is an alternative version of the AP metric [9]. It uses APMedium to measure how well a model performs with medium-sized objects (those with an area between 32^2 and 96^2) and AP^{Large} to measure how well it performs with large objects (those with an area above 96^2). Similarly, AR Across Scales has AR^{Medium} and AR^{Large}. Methods that use the PoseTrack and MPII datasets [31] are often evaluated using the mean Average Precision (mAP) across all classes.

6. Feature Extraction

The pose estimation task involves predicting each joint's position in a human pose from an image. A single image typically contains various objects besides humans, such as animals, cars, buildings, and furniture. Therefore, feature extraction is necessary to distinguish the human pose from these other objects before estimating the position of the joints. Unlike traditional HPE methods, which explicitly extract features from an image using techniques such as HOG (histogram of oriented gradients), deep-learning-based HPE generates features implicitly as part of a CNN operation [69]. In HPE tasks, CNNs are used to extract features that represent the shape of the human body.

A Convolutional Neural Network (CNN) is an artificial neural network used in computer vision for object detection, classification, categorization, and estimation [70]. As shown in Figure 16, CNNs consist of three primary layers: convolutional, pooling, and fully connected. The convolutional layer extracts features by applying a dot product operation between the input pixels and a filter (or kernel), where the kernel window size is smaller than the input size. The output of the convolutional layer is a feature map. Shallow layers (those close to the input image) produce abstract features such as edges and lines, while deeper layers produce more semantic features [2,67]. After the convolutional layer, the pooling layer reduces the input volume and computational complexity. Two types of pooling layers are commonly used: max pooling and average pooling. Max pooling takes the maximum value within the kernel window, while average pooling takes the mean value. The output of the convolutional and pooling layers is typically two-dimensional (width and height) or three-dimensional (with depth representing the number of feature maps). The fully connected layer converts these features into a one-dimensional representation. Each class's probability is then calculated using a classifier such as the SoftMax activation function.

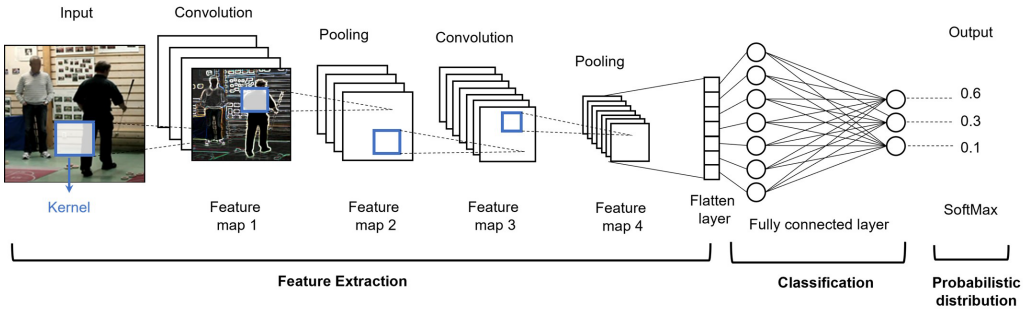


Figure 16. The general architecture of a convolutional neural network, whereas the kernel represents the window size used for extracting the features of the image.

Many researchers have developed custom CNN models to implicitly extract features using convolutional, pooling, and fully connected layers. These models are trained on large datasets such as ImageNet and Open Images. A convolutional network trained on a large dataset to extract image features is also known as a backbone [70]. The first backbone developed for implicit feature extraction was AlexNet [71], which consists of five convolutional layers and three fully connected layers, with max-pooling layers following some of the convolutional layers. The DeepPose model [17] adopted the AlexNet backbone to estimate human poses from images and was the first method to use deep learning for joint estimation. Since the introduction of AlexNet, many other models have been proposed, including GoogLeNet, VGGs, ResNet, ResNeXt, and HRNet [70]. Most HPE studies have recently used ResNet as their backbone [30,33].

Backbones can generally be classified as either deep or lightweight [72]. The main difference between these two types is the number of parameters in the model; deep models have more parameters and are larger than lightweight models. The choice between a deep or lightweight model depends on the goal of the pose estimation task. If precision is the primary objective, deep networks such as VGGs, ResNets, and HRNets are excellent backbones. If efficiency (speed) is more important, particularly for real-time estimation or mobile devices, lightweight networks such as GoogLeNet and MobileNetV2 are suitable backbones. Table 6 shows some popular CNN architectures used in HPE. Please refer to these works for more details about other CNN architectures [67,70,72].

Table 6. Popular backbones network used as an encoder in human pose estimation. The number of parameters and the FLOP metric help to show the size and speed of the model, respectively. "M" = million, and "G" = Giga, i.e., billion.

CNN Network	Year	Layers	Accuracy	#Params	FLOP	Keynote
Deeper network						
AlexNet [71]	2012	7	84.70%	61 M	7.27 G	First network used GPU.
VGGNet-16 [73]	2015	15	93.20%	138 M	154.7 G	Using AlexNet with modified filter sizes.
HRNet_W48 [74]	2019	48	94.00%	77.5 M	16.1 G	Providing high-resolution features.
ResNet-50 [75]	2016	49	96.40%	23.4 M	3.8 G	Skip connection.
ResNeXt [76]	2017	49	97.00%	23 M	-	A Derivative of ResNet.
Lightweight network						
MobileNetV2 [77]	2018	53	90.29%	3.5 M	300 M	Has a low number of parameters.
GoogLeNet [78]	2015	22	93.30%	6.8 M	1500 M	Design with multiple filter sizes.

Number of model parameters.

7. Existing Methods of Human Pose Estimation

Estimating a human pose from a video is different from an image. Videos present more challenges than images due to motion blur and dynamic backgrounds. In addition, estimating one person's pose (single pose) is less challenging than estimating two or more persons' poses (multipose). This section will review the existing methods for estimating single and multiple poses in images and videos. Figure 17 shows the general taxonomy of these methods.

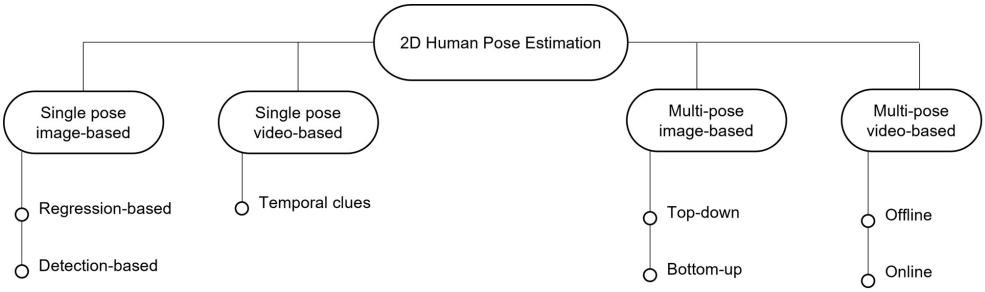


Figure 17. Survey taxonomy of 2D human pose estimation methods.

7.1. Single Pose Estimation Image-Based

If there is only one person in an image, two approaches can be used to estimate the pose: regression-based and detection-based. Regression-based methods use an end-to-end framework [34] to learn a mapping from an image to the joint coordinates of the body, directly producing joint coordinates. On the other hand, detection-based methods use heatmaps (2D Gaussian distribution) to independently detect key points [28,30] and then combine these maps to obtain a predicted pose. Figure 18 illustrates the pipelines of the regression and detection approaches. Both approaches have their pros and cons. While detection learning is supervised by dense pixel information, direct regression learning of a single point is challenging due to being a highly nonlinear problem. In the following sections, we will discuss existing methods for each approach in detail.

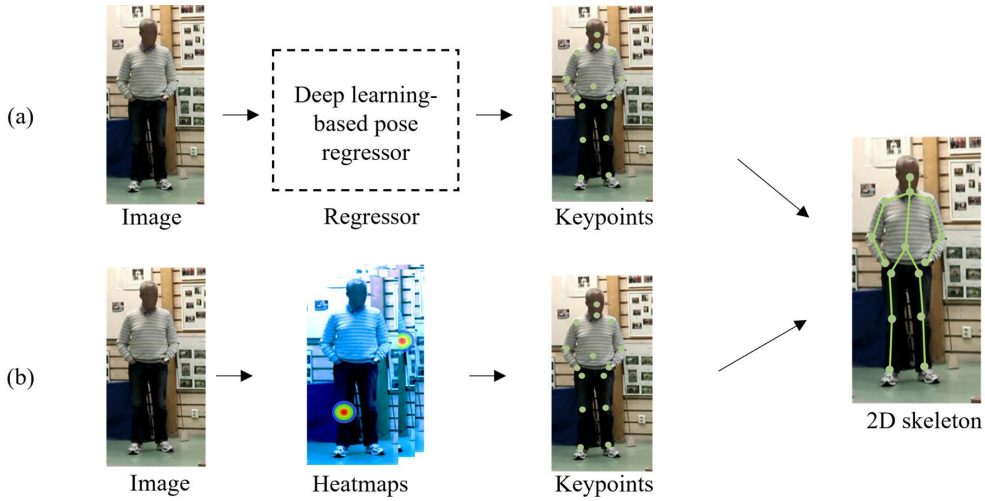


Figure 18. The typical approaches of estimating single person's pose in the image, where (**a**) is the regression approach and (**b**) is the detection approach.

7.1.1. Regression-Based Pose Estimation

Some works have followed the regression-based paradigm to predict joint location. For example, Toshev et al. [17] proposed the DeepPose model, the first work that uses deep neural networks to predict human body joints. The DeepPose framework, as shown in Figure 19, has an initial stage that estimates the initial joint positions, followed by multiple stages that improve performance by refining results from the previous stage. Another early work that used deep CNNs was by Li et al. [66]. The authors proposed a heterogeneous multitask framework with two tasks: a regression task for predicting body joints and a classification task using a sliding window to detect whether a joint or part exists.

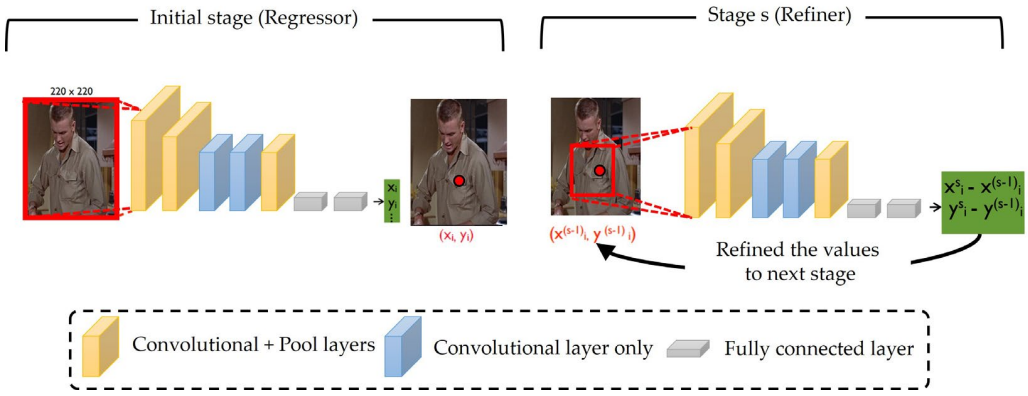


Figure 19. DeepPose framework, the first model that used deep learning for 2D human pose estimation.

Directly predicting joint coordinates from an input image is challenging. As a result, more powerful networks with refinement and body model structures have been proposed. Based on GoogLeNet, Carreira et al. [51] introduced Iterative Error Feedback (IEF), which recursively combines the input image and output results to provide a self-correcting model that iteratively refines a proposed solution by returning and correcting erroneous predictions. Sun et al. [79] suggested a regression method based on structure-aware information. The structure-aware representation includes information about the body's structure to obtain more reliable results than just using joint locations.

To provide a fully differentiable framework, Luvizon et al. [80] presented an end-to-end regression method that replaces the argmax function with soft-argmax to transform feature maps into joint body coordinates. This technique can indirectly learn heatmap representations. Zhang et al. [50] attempted to provide an efficient model with low com-

putational cost. They proposed a novel Fast Pose Distillation (FPD) method that led to the developing a lightweight human pose model through knowledge distillation from a teacher to a student model.

Li et al. [81] recently investigated Residual Log-likelihood Estimation (RLE) to capture changes in output distribution, facilitating the training process rather than relying on the original unreferenced underlying distribution. Instead of predicting key points individually, which is difficult when occlusion problems exist, Shamsafar et al. [82] proposed using both part-based and whole-body predictions. Tables 7 and 8 summarize the above methods and show their performance.

Table 7. Types and techniques of deep learning (DL) used by different studies to estimate human pose through a regression-based approach.

DL Type	Address the Issues	Techniques Used	Studies
CNN	Incorrect the predicted joint	Multistage	[17]
		Iterative optimization	[51]
		Graphical model	[79,83]
	Self/object occlusion	Multitask	[66]
	Limitation in device resources	Distillation	[50]

Table 8. Performance of the single-person pose estimation methods on the (MPII test set), where the input is an image. All these methods follow the regression-based approach. The number of parameters shows the model's size, while the GFLOP metric shows the model's speed.

Method	Year	Backbone	Input Size	#Params	GFLOPs	PCKh@0.5
IEF [51]	2016	GoogLeNet	224 × 224	-	-	81.3
Sun et al. [79]	2017	ResNet-50	224 × 224	-	-	86.4
FPD [50]	2019	Hourglass	256 × 256	3 M	9.0	90.8
Luvizon et al. [80]	2019	Hourglass	256 × 256	-	-	91.2

Number of model parameters.

7.1.2. Detection-Based Pose Estimation

In detection-based methods, ground truth is produced from joint locations by applying a 2D Gaussian distribution with its center at the joint. For each joint j_i with coordinates (x, y), the proposed methods must produce heatmap *Hi* [84], as shown in Figure 18b. The total number of heatmaps equals the total number of *N* joints, where the heatmap for each joint represents $\{H_1, H_2, \ldots, H_N\}$. The detection-based framework faces two difficulties [9]. The first is generating a keypoint heatmap by estimating the likelihood that each pixel represents a joint. The second involves refining the resulting joint confidence map.

One of the most significant networks that have served as a backbone for various studies is the Stacked Hourglass [85,86]. It consists of repeated down-sampling (bottom-up) and up-sampling (top-down) operations with intermediate supervision to estimate human poses. The bottom-up process converts high resolutions to low resolutions through pooling layers. The top-down approach then uses up-sampling layers to recover high-resolutions from low-resolution data. Figure 20 shows the framework of a single hourglass.

Many studies have improved the Hourglass model. Hua et al. [86] proposed an affinage module that refines low-resolution features from the up-sampling operation of Hourglass to obtain high-resolution feature maps. On the other hand, Yang et al. [87] improved the scale invariance of networks by using pyramid residual blocks instead of residual blocks in the stacked hourglass. The Pyramid Residual Modules (PRMs) framework learns convolutional filters from multiscale input features. Furthermore, Tian et al. [88] proposed the Densely Connected Attentional PRM (DCAPRM), which improved PRM through a densely connected network. Another work that uses intermediate supervision is

Convolutional Pose Machines (CPMs) [89]. Ke et al.'s method [56] also uses supervision and regression on a Stacked Hourglass at multiple scales.

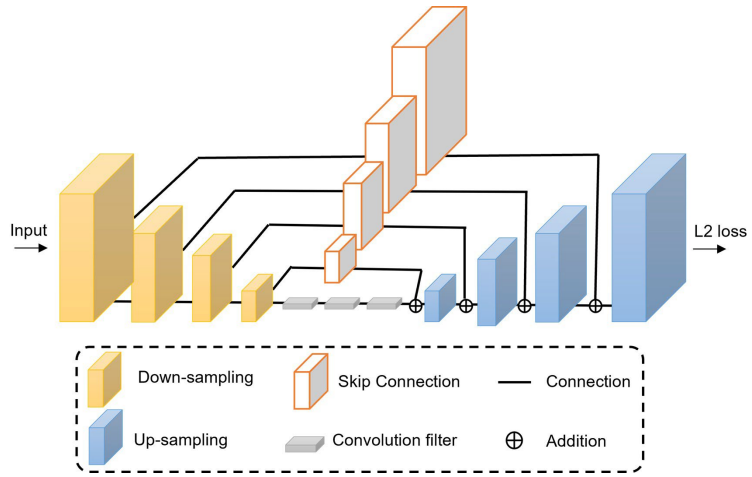


Figure 20. Single hourglass. The down-sampling operation uses the max pooling layer, whereas the up-sampling uses the transpose convolution layer. The input is the image, whereas the output is the predicted value of the joints calculated by L_2 loss function.

Instead of recovering high-resolution from low-resolution, Sun et al. [74] proposed a High-Resolution Network (HRNet) that maintains high-resolution representations of features throughout the entire network. Figure 21 shows the HRNet framework. Some studies focus on solving specific problems. For example, Belagiannis and Zisserman [59] improved the performance of the Human Pose Estimation (HPE) model by integrating feedforward and recurrent modules to fine-tune the results iteratively to predict occluded key points. Hwang et al. [90] combined information from local and global networks to estimate complex poses such as athlete poses. In contrast, Lifshitz et al. [91] suggested a voting technique that uses information from the entire image. Each pixel votes for the best possible position of each keypoint, and the results are tallied to obtain the optimal pose configuration.

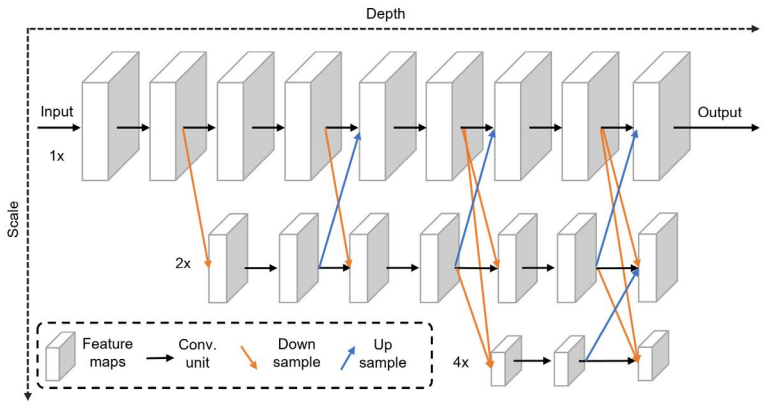


Figure 21. The stages and branches in a High-Resolution Network. This network has multiple stages and branches, representing depth and scale. As the branch level increased, the input resolution scale decreased. Each stage contains convolution layers (Conv. unit) to extract the feature, and the downsampling and upsampling help to extract deep features.

On the other hand, several studies have utilized graphical models to correct pose estimation by encoding body structure information into networks. Tompson et al. [83] used a convolutional Part-Detector network to produce a heatmap for each joint. A Markov Random Field (MRF) is then formulated from a Spatial-Model to remove incorrect pose

predictions implicitly. Chen et al. [92] presented a method for learning pairwise relationships from predicted confidence maps, in which conditional probabilities of joint and spatial relationships within image patches are learned with deep CNNs. Instead of using pose priors or a holistic perspective to estimate human poses, Fan et al. [14] introduced dual-source deep CNNs that combine these methods. The dual-source uses two types of image patches (whole body and local part) as inputs and produces joint detection results of sliding windows as heatmaps and joint localization coordinates. Based on graph convolutional networks, Bin et al.'s recent work [12] developed a novel Pose Graph Convolutional Network (PGCN) to construct a graph between body key points.

As graphical models do not account for the challenge of keypoint occlusion, Fu et al. [93] presented the Occlusion Relational Graphical Model (ORGM) to capture both occlusion by other objects and self-occlusion. Similarly, to reduce low-level image ambiguities caused by nearby persons, overlapping parts, and cluttered backgrounds, Tang et al. [94] proposed the Deeply Learned Compositional Model (DLCM). This model integrates bottom-up/top-down inference stages across multiple semantic levels to understand compositional relationships among body parts. Tang et al. [95] proposed a Part-based Branching Network (PBN) to learn representations unique to each group of body parts. Su et al. [96] introduced the Cascade Feature Aggregation (CFA) method, which attempts to be more resistant to changes such as low resolution and partial occlusions through cascades of several hourglasses.

Generative Adversarial Networks (GANs) have also been utilized to estimate human poses by providing adversarial supervision. Chen et al. [97] proposed an Adversarial PoseNet network to address the problem of occlusions and overlapping joints. Similarly, Chou et al. [13] employed a generator and a discriminator, each constructed from a stacked hourglass network. Shamsolmoali et al. [98] designed two residual Multiple-Instance Learning (MIL) networks, a generator and a discriminator, to learn constraints on human structure priors.

Recently, some studies have focused on enhancing model efficiency. For example, Papaioannidis et al. [57] proposed a fast CNN architecture using a global body head and a joint body regression head to quickly estimate human poses for lightweight embedded systems. Similarly, Dai et al. [99] presented the FasterPose model, which aims to reduce computational costs by analyzing and designing Low-Resolution (LR) features. Tables 9 and 10 provide summaries of detection-based methodologies and their performance, respectively.

Table 9. Types and techniques of deep learning (DL) used by different studies to estimate human pose through a detection-based approach.

DL Type	Address the Issues	Techniques Used	Studies
CNN	Incorrect the predicted joint	Multistage	[89,96]
		Refinement	[91]
		Graphical model	[12,14,83,92]
	Different scales of the human body	Multibranch	[87,88]
	Complex pose	Multitask	[90]
	Self/object occlusion	Graphical model	[91]
		Multistage	[56,94]
		Multibranch	[95]
	Feature resolution	Multistage	[85,86]
		Multistage/branch	[74]
	Limitation in device resources	Distillation	[57,99]
GAN	Self/object occlusion	Multistage	[13]
		Multitask	[97,98]
RNN	Incorrect the predicted joint	Multistage	[59]

Table 10. Performance of the single-person pose estimation methods on the (MPII test set), where the input is an image. All these methods follow the detection-based approach. The number of parameters shows the model's size, while the GFLOP metric shows the model's speed.

Method	Year	Backbone	Input Size	#Params	GFLOPs	PCKh@0.5
MRF [83]	2014	AlexNet	320 × 240	40 M	-	79.6
Lifshitz et al. [91]	2016	VGG	504 × 504	-	-	85.0
CPM [89]	2016	CPM	368 × 368	-	-	88.5
Stacked Hourglass [85]	2016	Hourglass	256 × 256	25.1 M	19.1	90.9
Hua et al. [86]	2020	Hourglass	256 × 256	41.9 M	56.3	91.0
Papaioannidis et al. [57]	2022	ResNet-50	256 × 256	-	-	91.3
Chou et al. [13]	2018	Hourglass	256 × 256	-	-	91.8
AdversarialPoseNet [97]	2017	En/Decoder	256 × 256	-	-	91.9
PRM [87]	2017	Hourglass	256 × 256	28.1 M	21.3	92.0
Ke et al. [56]	2018	Hourglass	256 × 256	-	-	92.1
DLCM [94]	2018	Hourglass	256 × 256	15.5 M	15.6	92.3
HRNet [74]	2019	HRNet	256 × 256	28.5 M	9.5	92.3
PGCN [12]	2020	Hourglass	256 × 256	-	-	92.4
PBN [95]	2019	Hourglass	256 × 256	26.69 M	-	92.7
DCAPRM [88]	2019	Hourglass	256 × 256	-	-	92.9
CFA [96]	2019	Hourglass	384 × 384	-	-	93.9

Number of model parameters.

7.2. Single Pose Estimation Video-Based

Many real-life applications, such as detecting suspicious activity during Hajj and Umrah, cannot be implemented by estimating pose from a single image. Therefore, tracking and predicting the location of key points across video frames is essential. However, estimating poses from videos presents challenges. Unlike image-based estimation, where the background is static, video-based estimation has a dynamic background due to camera movement [100]. This background change adds to the challenges of 2D Human Pose Estimation (HPE). Additionally, effects such as motion blur and changing light intensity also contribute to these challenges. However, in video-based estimation, estimating poses with occlusion problems is more straightforward than in image-based estimation because the location of invisible and occluded key points can be predicted from other frames.

Videos consist of multiple images (frames). Most video-based work on single-person pose estimation explores ways to refine results from a single frame by propagating temporal clues across frames [33,101]. Temporal clues can be sequential, local, or global, as shown in Figure 22. Sequence model-based methods, such as Recurrent Neural Networks (RNNs), rely on previous information to predict current information; thus, using only the previous frame to estimate pose in the current frame is known as sequential temporal clues. On the other hand, local cues estimate human pose using information from adjacent frames (previous and next frames). Unlike local cues, global cues obtain necessary information from different distant frames.

Pfister et al. [102] were the first to use a deep CNN to estimate human upper-body joints from videos. Their network exploits temporal information in videos by inserting multiple frames into color channels as input, where each video frame has three colors (RGB). In addition to color channels, Jain et al. [103] incorporated motion features, resulting in two inputs being inserted into the CNN framework, thus providing a spatial-temporal model. Nie et al. [104] also proposed a spatial-temporal model based on an And-Or graph, which combines action recognition with video pose estimation. However, this model suffers from occlusion problems due to hand-crafted features. Liu et al. [105] were also interested in the spatial-temporal model.

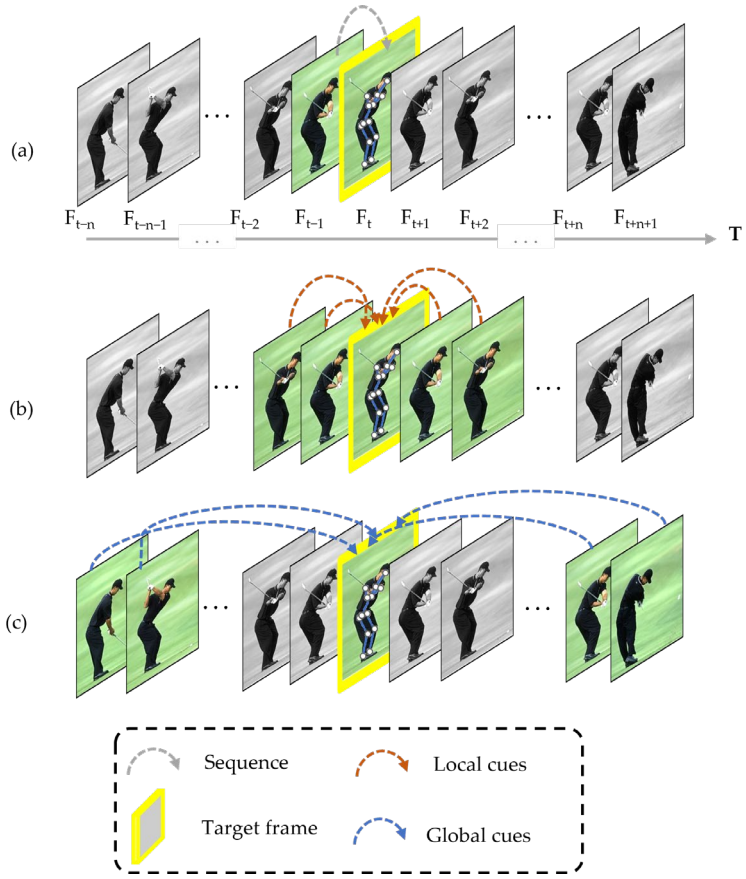


Figure 22. Different types of temporal clues across frames, (**a**) frame sequence, (**b**) local cues, and (**c**) global cues, are used to estimate the pose in the target frame.

Some studies have relied on optical flow models to identify the motion flow of humans across video frames. Pfister et al. [53] used optical flow to combine multiple frames by warping predicted heatmaps from neighboring frames onto a target frame. To improve pose estimation in long videos that may face occlusion problems, Charles et al. [106] introduced a personalized Convolutional Network (ConvNet) pose estimator that leverages annotated high-quality data to enhance the performance of a generic pose estimator. Song et al. [107] proposed a Thin-Slicing Network model that uses a flow-warping layer to align joints of the current frame with previous heatmaps.

Recurrent Neural Networks (RNNs) provide another means of incorporating temporal context information. Gkioxari et al. [108] proposed a chained model that adapts the sequence-to-sequence model to estimate spatial pose in videos by using the previous frame's result (body keypoint) as input for the current frame. Luo et al. [109] presented a recurrent CNN model with Long Short-Term Memory (LSTM) for Human Pose Estimation (HPE). Similarly, Artacho et al. [110] proposed the UniPose-LSTM framework, which leverages the memory capability of LSTM. Fan et al. [100] introduced the Motion Adaptive Pose Net (MAPN), which captures spatial-temporal features using motion-compensated convolutional LSTM and skips feature extractions based on residual information for a set of frames. Li et al. [111] presented a module called Temporal Consistency Exploration (TCE) that overcomes the shortcomings of both RNN and optical flow methods.

As most datasets annotate key points after the K_{th} frames, leaving some frames without annotation, several studies [8,112,113] have relied on unsupervised approaches to estimate key points along video frames. Zhang et al. [112] introduced the Key Frame Proposal Network (K-FPN), which recovers the entire pose sequence unsupervised by selecting spatial and temporal information from a set of keyframes. Schmidtke et al. [113] proposed an unsupervised method that uses simple 2D Gaussian templates for feature extraction.

However, this method is not robust against dynamic backgrounds. To address this issue, Jiao et al. [101] introduced a framework called Global-Local Enhanced Pose (GLPose) estimation, which integrates results from nearby temporal frames (local features) and similar global information to the target frame for pose prediction. Ludwig et al. [8] also proposed an unsupervised method using two techniques: selective pseudo labeling and mean teacher training.

Ma et al. [114] recently proposed a semi-supervised approach that utilizes labeled frames and temporal dynamics (predicted key points) to address the problem of limited availability of temporally sparse annotations in videos. They introduced the REinforced MOtion Transformation nEtwork (REMOTE) framework, where a Motion Transformer (MT) and an RL-based Frame Selection Agent (FSA) are combined. Nie et al. [115] proposed the Dynamic Kernel Distillation (DKD) model, which reduces the computational cost using pose kernel distillation for a lightweight pose estimation model.

Table 11 summarizes all of the above methods, while Table 12 shows their performance in estimating single-person poses in videos using the JHMDB dataset for testing.

Table 11. Types and techniques of deep learning (DL) used by different studies to estimate a single person's poses in the video.

DL Type	Address the Issues	Techniques Used	Studies
CNN	Few annotations	Multistage	[8,112,114]
		Multitask	[101]
	Limitation in device resources	Distillation	[115]
	Capturing spatial-temporal features	Multistage/branch	[103,111]
		Graphical model	[104]
		Optical flow	[53,106,107]
		Multistage	[102,105]
RNN	Capturing spatial-temporal features	Multistage/branch	[110]
		Multistage	[100,109]
		Multibranch	[108]

Table 12. Performance of the single-person pose estimation methods on the (Sub-JHMDB test set), where the input is a video.

Method	Year	Backbone	Input Size	GFLOPs	PCKh@0.2
And-Or graph [104]	2015	-	-	-	55.7
Song et al. [107]	2017	CPM	368 × 368	-	81.6
Luo et al. [109]	2018	-	368 × 368	70.98	93.6
DKD [115]	2019	ResNet50	255 × 256	8.65	94.0
K-FPN [112]	2020	ResNet17	224 × 224	4.68	94.5
MAPN [100]	2021	ResNet18	257 × 256	2.70	94.7
GLPose [101]	2022	HRNet	384 × 288	-	95.1
REMOTE [114]	2022	ResNet50	384 × 384	-	95.9
TCE [111]	2020	Res50-TCE-BC	256 × 256	-	96.5

7.3. MultiPose Estimation Image-Based

Multiperson pose estimation is more challenging than single-person estimation because the model needs to predict the location of all the key points for all individuals, regardless of the number of individuals. Figure 23 shows two well-known approaches for estimating multiperson poses: top-down and bottom-up. The top-down approach has two steps: first, detecting people in an input image and then estimating individual key points within each person's bounding box. On the other hand, the bottom-up approach simultaneously predicts all key points and then assigns them to different people. This section will cover most methods for these two approaches.

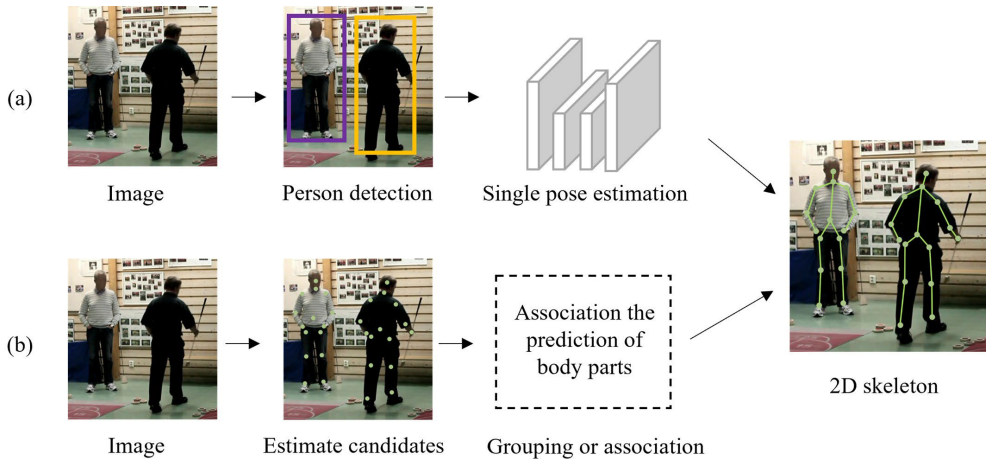


Figure 23. Typical approaches to estimate multiple people poses in the image, where (**a**) is the top-down approach and (**b**) is the bottom-up approach.

7.3.1. Top-Down Approach

The main idea of the top-down approach [67] is to use a separate stage of CNN and classification to recognize objects before estimating human poses. A region-proposal pipeline detector must first detect each human pose in the image before applying a single-person pose estimation method to predict human joints. The strength of this framework is its ability to extract features with high accuracy due to using a pretrained model for pose detection. However, its weakness is the slow processing time. This section will discuss some works that focus on building proposal pipeline detectors.

The first method in this area was R-CNN (Region-based Convolutional Network) [116], which integrated AlexNet with a selective search strategy considering edge, gradient, texture, and color. However, one of R-CNN's drawbacks was its slow detection speed. Later methods such as Fast R-CNN and Faster R-CNN addressed this issue. R-CNN was improved by Moon et al. [117], who utilized multiscale information for each person in the image to estimate poses in parallel.

As an alternative to extracting features from each proposed region, Fast R-CNN [118] used the entire image as input to extract features, speeding up the detection process. Fast R-CNN increased detection speed but relied on external region proposal methods (such as selective search), impacting the process's speed. Faster R-CNN [119] addressed this issue by replacing selective search with a Region Proposal Network (RPN).

Other proposed models focused on increasing object detection accuracy. One such model is Mask R-CNN [27], which extended Faster R-CNN and used two built-in feature extractors, ResNet and FPN, to achieve good accuracy and efficiency. It also replaced RoIPool with RoIAlign to obtain small feature maps and address the issue of feature extraction from the input. Fang et al. [120] proposed the Regional Multiperson Pose Estimation (RMPE) framework to address the problem of imprecise detection of human bounding boxes. According to Huang et al. [121], data transformation and encoding-decoding affected top-down approach performance. Therefore, they introduced the Unbiased Data Processing (UDP) method, which statistically analyzed frequently biased data processing in human pose estimation and processed data based on unit length rather than pixels.

Chen et al. [62] presented a Cascaded Pyramid Network (CPN) that generated features for simple joints such as eyes and heads using GlobalNet to address the failed estimation of difficult key points. RefineNet handled difficult key points caused by occlusions or invisibility. The CPN framework is shown in Figure 24. Li et al. [122] adopted the GlobalNet of CPN as a single-stage module for extracting features and proposed the MultiStage Pose Network (MSPN) for estimating multiperson poses.

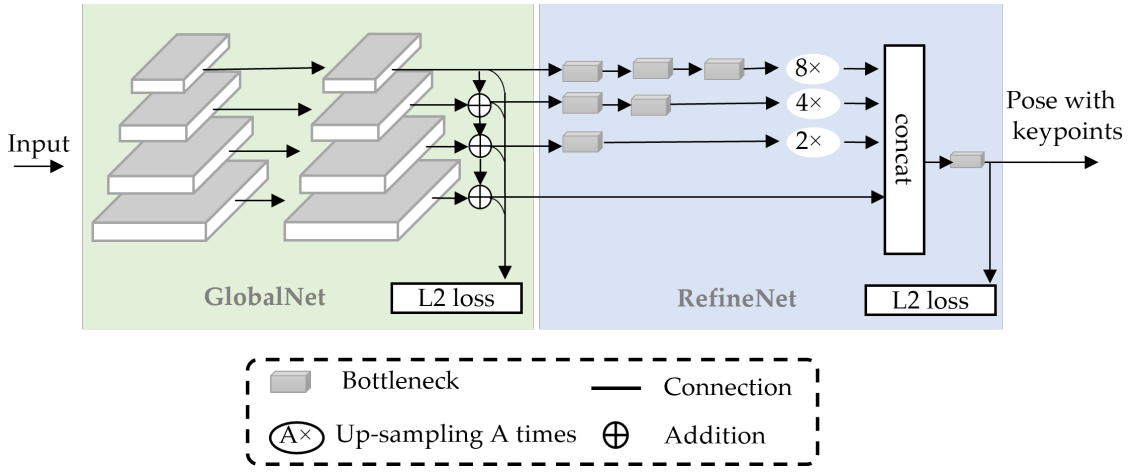


Figure 24. Cascaded Pyramid Network. One of the popular networks that follow a top-down approach, which is used to estimate multiperson poses.

Data augmentation was also used to estimate difficult joints. To overcome occlusion problems, Zhou et al. [60] adopted a Siamese network that took two inputs: a pose and the same pose with erased key points as an occlusion simulation. Similarly, Xie et al. [123] used data augmentation to avoid collapsing problems by feeding the network with a typical image and an image covering some key points. Moon et al. [55] presented an independent single-stage network called PoseFix to correct human poses by training the model using data augmentation that added synthesized errors such as jitter, inversion, and swap to pose ground truth.

The Graph-PCNN model, proposed by Wang et al. [124], has a two-stage, model-agnostic framework that uses a heatmap regressor for rough keypoint localization and a custom-designed graph pose refining module to increase accuracy. Cai et al. proposed the Residual Steps Network (RSN) [125], which won the COCO Keypoint 2019 challenge. While RSN can learn detailed local representations through intra-level features, its pose refinement machine was designed to find a better balance between using local and global representations in features. To reduce the negative impact of background, Dong et al. [65] used an hourglass network and added a global attention module. Wang et al. [54] used semantic-aware transfer to estimate multiperson poses in crowded scenes to alleviate missing key point detection. Zhang et al. [20] proposed one of the well-known top-down methods called DARK (Distribution-Aware coordinate Representation of Key points), which addressed quantization error by decoding predicted keypoint heatmaps into 2D coordinates (x, y) of joints.

Most studies modified [126,127] or used [20,54,121] the HRNet model as a backbone. However, following this method increases computational complexity. Therefore, recently, many models have been proposed, like LiteHRNet [128], HRNet-Lite [129], and SRPose [130], that aim to build small networks. Another work that builds an efficient network is McNally et al. [131]. They proposed the EvoPose2D network, designed using neuroevolution to provide an effective weight transfer scheme that accelerates 2D human pose networks. Xu et al. [132] proposed a Lightweight Dynamic convolution Network (LDNet) that aimed to reduce the number of network parameters using depthwise separable convolution. Recently, Xu et al. [133] designed ZoomNet, a single neural network for estimating whole-body human parts, including body, feet, face, and hands. Tables 13 and 14 summarize and display the performance of these methodologies, respectively.

Table 13. Types and techniques of deep learning (DL) used by different studies to estimate human pose through a top-down approach.

DL Type	Address the Issues	Techniques Used	Studies
CNN	Incorrect the predicted joint	Refinement	[55]
		Graphical model	[124]
	Incorrect the bounding box	Multistage/branch	[117]
		Multistage	[27,116,118,119]
		Nonmaximum Suppression	[120]
	Feature resolution	Multistage	[122,125]
		Multistage/branch	[127]
	Limitation in device resources	Multistage	[126]
		Multibranch	[130,131]
		Multistage/branch	[128,129]
	Self/object occlusion	Multibranch	[60]
		Multistage/branch	[62]
	Variant background	Multistage	[65]
		Multitask	[54]
	Quantization error	Modifying Gaussian kernel	[20,126]
	Estimating whole body	Multitask	[132]

Table 14. Performance of the multiperson poses estimation methods on the (COCO test-dev set), where the input is an image. All these methods follow the top-down approach. The number of parameters shows the model's size, while the GFLOP metric shows the model's speed.

Method	Year	Backbone	Input Size	#Params	GFLOPs	AP
RMPE [120]	2017	PyraNet	320 × 256	-	-	61.8
Mask R-CNN [27]	2017	ResNet	800 × 800	-	-	63.1
MSPN [122]	2019	ResNet-50-FPN	640 × 640	-	-	68.2
LiteHRNet [128]	2022	Lite-HRNet-30	384 × 288	1.8 M	0.70	69.7
Chen et al. [62]	2018	ResNet	384 × 288	102 M	6.2	72.1
LDNet [131]	2022	LDNet	384 × 288	5.1 M	3.7	72.3
HRNet-Lite [129]	2023	HRNet-W32	256 × 192	14.5 M	2.9	73.3
SRPose [130]	2023	HRFormer-S	256 × 192	8.86 M	3.34	75.6
EvoPose2D [126]	2021	EvoPose2D	512 × 384	14.7 M	17.7	75.7
HrFormer [127]	2021	HRFormer-B	384 × 288	43.2 M	26.8	76.2
DARK [20]	2020	HRNet	384 × 288	63.6 M	32.9	76.2
UDP [121]	2020	HRNet	384 × 288	63.8 M	33.0	76.5
PoseFix [55]	2019	HR + ResNet	384 × 288	-	-	76.7
Graph-PCNN [124]	2020	HRNet	384 × 288	-	-	76.8
Wang et al. [54]	2021	HRNet	384 × 288	63.9 M	35.4	76.8
Xie et al. [123]	2021	HRNet	384 × 288	63.6 M	32.9	77.2
DiffusionPose [126]	2023	HRNet-W48	384 × 288	74 M	49	77.6
RSN [125]	2020	4×RSN-50	384 × 288	111.8 M	65.9	78.6

Number of model parameters.

7.3.2. Bottom-Up Approach

Instead of using region proposals to detect people, DeepCut [26] adopted Fast R-CNN first to detect all key points and then group body parts into the same individual. Since the DeepCut model heavily relies on part detectors, DeeperCut [134] adopted DeepCut and replaced the old keypoint detector with robust body part detectors. Another work that achieved real-time performance is that of Varadarajan et al. [135]. Unlike DeepCut and DeeperCut models that use an Integer Linear Programming (ILP) solver to formulate keypoint assignment, which is time-consuming due to the NP-hard problem, they proposed an assignment algorithm that greedily selects body parts, reducing time complexity.

MultiPoseNet [136] offered a multitask model that independently identified key points and human proposals before using a pose residual network to allocate discovered key points to different pose bounding boxes. One of the most well-known methods in the

bottom-up approach is OpenPose [25], which proposed Part Affinity Fields (PAF) to learn keypoint locations and their associations through a set of 2D vector fields that indicate the location and orientation of body parts. The OpenPose model won the COCO 2016 key points challenge and achieved real-time performance with its two-branch, multistage architecture (Figure 25). Kreiss et al. [137] presented the PifPaf method, in which their network encoder two fields, Part Intensity Field (PIF) and PAF, for locating and associating body parts, respectively. Nasr et al. [138] employed various approaches for estimating multiperson poses, such as PAF for part association and person parsing, after applying a confidence map for keypoint identification.

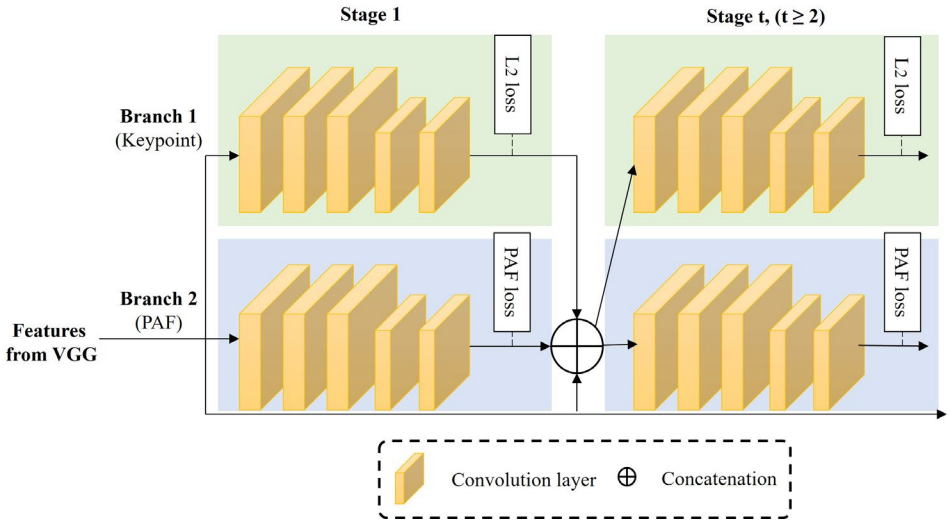


Figure 25. OpenPose architecture is one of the most popular methods of the bottom-up approach. Two losses are used in OpenPose, L_2 and PAF (part affinity fields), for predicting confidence heatmaps and the connection between the body parts.

While most bottom-up methods use post-processing to group key points, the Hierarchical Graph Grouping (HGG) method [139] provides an end-to-end trained network that detects key points and clusters them using a graph network. Another end-to-end trained method that includes a detector for grouping key points is CenterGroup [58]. Jin et al. [140] proposed a two-stage pipeline for estimating multiperson poses. The first stage detects all key points, while the second stage predicts offsets from key points to body centers using a greedy grouping strategy with a dynamic threshold. In addition to detecting all key points, Du et al. [141] combined groups of specific key points with prior knowledge to obtain better associative encoding.

Pose Partition Networks (PPN) [142] introduced a novel dense regression module to detect and partition key points for multiple people. HigherHRNet [143] attempted to estimate small-scale people of different sizes by adopting HRNet and adding a deconvolution module that generates multiresolution and high-resolution heat maps. Instead of predicting key points using fixed standard deviation, which sometimes fails to handle variations in human scales and labeling ambiguities, Luo et al. [144] proposed Scale-Adaptive Heatmap Regression (SAHR) to adjust the standard deviation for each key point automatically.

Like ZoomNet, Hidalgo et al. [145] proposed a single network for estimating whole-body key points, including body, face, hands, and feet. However, they followed a bottom-up approach instead of a top-down one. Zhao et al. [146] presented a network architecture consisting of a multistage and two branches that combined local and global features by combining dense and sparse keypoint clusters in each branch to detect key points. Additionally, intra- and interclusters were used for grouping predicted key points into individuals. Tables 15 and 16 summarize the methodologies of the bottom-up approach and their performance.

Table 15. Types and techniques of deep learning (DL) used by different studies to estimate human pose through a bottom-up approach.

DL Type	Address the Issues	Techniques Used	Studies
CNN	Self/object occlusion	Detecting and grouping	[26,58,128]
		Multitask	[129]
		Multibranch	[141]
	Limitation in device resources	Multitask	[136]
		Part Affinity Fields	[25,137]
		Multistage/branch	[142]
	Feature resolution	Multistage/branch	[143,146]
	Different scales of the human body	Iterative optimization	[140]
		Multibranch	[144]
	Incorrect the predicted joint	Part Affinity Fields	[147]
	Estimating whole body	Part Affinity Fields	[145]
GNN	Incorrect grouping of key points	Graph layers	[139]

Table 16. Performance of the multiperson pose estimation methods on the (COCO test-dev set), where the input is an image. All these methods follow the bottom-up approach. The number of parameters shows the model's size, while the GFLOP metric shows the model's speed.

Method	Year	Backbone	Input Size	#Params	GFLOPs	AP
OpenPose [25]	2017	CMU-Net	368 × 368	-	-	61.8
Zhao et al. [146]	2020	Hourglass	512 × 512	-	-	62.7
PifPaf [137]	2019	ResNet	401 × 401	-	-	66.7
HGG [139]	2020	Hourglass	512 × 512	-	-	67.6
Du et al. [141]	2022	HrHRNet	512 × 512	28.6 M	-	67.8
MultiPoseNet [136]	2018	ResNet	480 × 480	-	-	69.6
HigherHRNet [143]	2020	HRNet	640 × 640	63.8 M	154.3	70.5
Jin et al. [140]	2022	HrHRNet	640 × 640	67.0 M	177.6	70.6
CenterGroup [58]	2021	HRNet	512 × 512	-	-	71.4
SAHR [144]	2021	HRNet	640 × 640	63.8 M	154.6	72.0

Number of model parameters.

7.4. Multipose Estimation Video-Based

When multiple people are present in a single video frame [148], three processes are applied: pose detection, estimation, and tracking. There are two types [147] of multiperson pose tracking: offline and online. Offline tracking methods typically represent complex spatiotemporal interactions across multiple frames for robust tracking but have high computational costs. Offline pose tracking methods often employ graph partitioning-based methodologies. In contrast, online pose-tracking methods are more efficient because they avoid the need to model complex spatiotemporal interactions. This section discusses various methods for offline and online approaches.

7.4.1. Offline Approach

The PoseTrack approach [149] used a spatiotemporal graph to represent joint body detection for every three frames and track each person's head and neck edges in a video. This approach addressed problems such as truncation of persons and occlusions. Similarly, these problems were tackled by the ArtTrack model proposed by Insafutdinov et al. [150]. Recently, Liu et al. [18] proposed a deep Dual Consecutive Pose (DCPose) model for pose prediction that addressed motion blur and pose occlusion problems. Similarly, the TDMI model [151] overcomes the blur and occlusion problem by exploiting the temporal difference information of the video frame. Ruan et al. [152] proposed an end-to-end network called Pose-Guided Ovonic Insight Network (POINet) that provides a multistage process consisting of feature extraction, similarity estimation, and identity assignment as a unified network.

Mask R-CNN [27] was extended by Girdhar et al. [52] to include temporal information as a third dimension (3D). The 3D Mask R-CNN has two stages: the first extracts pose features using Mask R-CNN, while the second tracks multiperson poses using temporal information. Similarly, HRNet [74] was extended by Wang et al. [19] to include temporal information between key points as 3D for tracking poses in videos. The approach used a clip-tracking network to estimate and track pose joints, followed by a video-tracking pipeline to merge the predicted poses of the same person.

Xiao et al. introduced two simple baselines for pose estimation and pose tracking called FlowTrack as model complexity increased [153]. A ResNet with a few deconvolutional layers was used to extract features from image frames (as shown in Figure 26), and the 3D Mask R-CNN pipeline was adapted with changes to the human detector and metric for pose tracking. In contrast, Bertasius et al. [154] proposed PoseWarper, which achieved strong pose detection performance. Table 17 summarizes the methodologies of the offline approach, while Table 18 shows their performance.

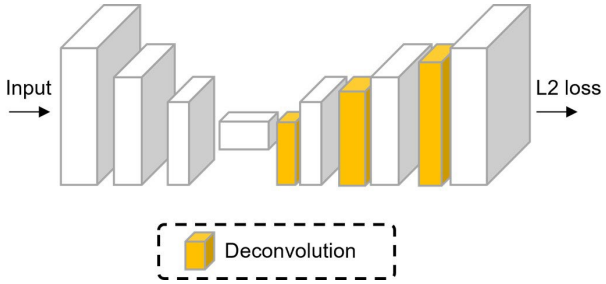


Figure 26. Simple Baseline network. Another popular network used to estimate multiperson pose.

Table 17. Types and techniques of deep learning (DL) used by different studies to estimate human pose through an offline approach.

DL Type	Address the Issues	Techniques Used	Studies
CNN	Few annotations	Multitask	[154]
	Self/object occlusion	Multistage/branch	[152]
		Multistage	[151]
	Model high complexity	Multistage	[153]
	Capturing spatial-temporal features	Graph layers	[149]
		Multistage	[52,150]
		Multistage/branch	[18,19]

Table 18. Performance of the multiperson pose estimation methods on the (PoseTrack 2017 test set), where the input is a video. The methods in this table follow an offline approach. MOTA metric is used to evaluate human pose tracking. The symbol * indicates the model was test based on the (PoseTrack 2017 val set).

Method	Year	Backbone	Input Size	Total mAP	Total MOTA
PoseTrack [149]	2017	-	-	59.4	48.4
ArtTrack [150]	2017	ResNet-101	-	59.4	48.1
3D Mask R-CNN [52]	2018	ResNet-18	256 × 256	59.6	51.8
Ruan et al. [152]	2019	ResNet-101	900 × 900	72.5	58.4
3D HRNet [19]	2020	3D HRNet	-	74.1	64.1
FlowTrack [153]	2018	ResNet-152	384 × 288	76.7	65.4
Bertasius et al. [154]	2019	HRNet	-	77.9	-
DCPose [18]	2021	HRNet	384 × 288	79.2	-
TDMI-ST [151]	2023	-	-	85.9 *	-

7.4.2. Online Approach

The PoseFlow model [36] was the first online pose tracker. Xiu et al. [155] proposed the PoseFlow model to leverage pose tracking using only a few frames. This model consists of three pipelines: a pose estimator based on Faster R-CNN, a pose flow builder, and a nonmaximal suppression pipeline to enhance tracking. Guo et al. [156] won the PoseTrack ECCV 2018 challenge by training their network using their proposed multidomain pose prediction method, which takes all poses embedded in a single frame and trains them using three datasets: COCO, MPII, and PoseTrack. Ning et al. [148] followed a top-down approach for estimating and tracking multiperson poses in videos.

Inspired by Part Affinity Fields (PAF) [25] representation, Doering et al. [157] proposed the JointFlow model that applied a Siamese network to extract pose features such as belief maps or PAFs. Temporal flow fields were then used to track multiperson poses between two frames (previous and current). Similarly, Raaj et al. [158] used a recurrent network that predicted poses from previous and current frames to present Spatio-Temporal Affinity Fields (STAF) based on PAFs. Yang et al. [159] employed a Graph Neural Network (GNN) to address the issue of missed detections across frames by learning pose dynamics from previous pose sequences and using that information to inform pose detection in the current frame. To solve ambiguity problems in crowded scenes, Stadler et al. [160] introduced different strategies based on a tracking-by-detection approach using a distance matrix.

Jin et al.'s [161] framework used two components, SpatialNet and TemporalNet, for detecting body parts and tracking human poses, respectively. The authors also proposed human and temporal instance embeddings to avoid drifting problems due to camera motion and achieve temporal consistency in video frames. Zhou et al. [147] improved pose association and estimation by proposing two modules: temporal keypoint matching and refinement for learning similarity metrics and correcting individual poses, respectively. The above online methodologies are summarized in Table 19, while Table 20 shows their performance.

Table 19. Types and techniques of deep learning (DL) used by different studies to estimate human pose through an online approach.

DL Type	Address the Issues	Techniques Used	Studies
CNN	Incorrect the predicted joint features Capturing spatial-temporal features	Multibranch Nonmaximum Suppression Multistage/branch Part Affinity Fields	[156] [155] [161] [157]
GNN	Capturing spatial-temporal features	Graph layers	[148,159]
RNN	Capturing spatial-temporal features	Part Affinity Fields	[158]

Table 20. Performance of the multiperson pose estimation methods on the (PoseTrack 2017 test set), where the input is a video. The methods in this table follow an online approach. MOTA metric is used to evaluate human pose tracking. The symbol * indicates the pose inference time is excluded.

Method	Year	Backbone	Input Size	Total mAP	Total MOTA	FPS
PoseFlow [155]	2018	Hourglass	-	63.0	51.0	10.0 *
JointFlow [157]	2018	-	-	63.4	53.1	0.2
Ning et al. [148]	2020	FPN	-	66.5	55.1	0.7
STAF [158]	2019	VGG	368 × 368	70.3	53.8	2.0
Guo et al. [156]	2018	ResNet-152	384 × 288	75.0	50.6	-
Jin et al. [161]	2019	Hourglass	-	77.0	71.8	-
Zhou et al. [147]	2020	HRNet	-	79.5	72.2	-
Yang et al. [159]	2021	HRNet	384 × 288	81.1	73.4	-

8. Discussion

The main objective of this survey is to analyze findings related to 2D human pose estimation (HPE) studies. We systematically collected papers on person pose estimation and detailed the process and results of collecting articles in Sections 2 and 3. Our procedures in searching and collecting articles were limited in estimating the person's pose as a 2D skeleton using deep learning methods from images and videos. As a result, we selected 100 papers and summarized them in Section 7 based on our selection criteria. We also discussed the four main components used to build HPE models: available datasets, loss functions, evaluation metrics, and pretrained feature extraction models (Sections 4–6). This section aims to further analyze these findings by answering the survey questions in Table 2. The answers to the survey questions are provided below.

8.1. Which Datasets Are Used to Analyze the Performance of the Deep Learning Methods (RQ_1)?

The dataset is a crucial component of the HPE process for training and testing models. Many datasets are available online, as listed in Section 4. When choosing a dataset, several requirements should be considered:

- Size: HPE is a nonlinear problem, and the size of the dataset affects model performance. Therefore, larger datasets are generally better for improving model accuracy;
- Data quality: A large amount of low-quality data can negatively impact performance. Data should be high-resolution and free of watermarks;
- Diversity: Datasets must include diverse data so that models can handle real-world scenarios. Hence, the datasets should provide different camera angles, poses, body shapes, races, ages, clothing styles, illumination conditions, and backgrounds;
- Complexity: Datasets must contain a variety of poses and actions to apply HPE to diverse applications. The datasets should include a range of poses, from simple ones such as standing and walking to more complex ones like flipping and kicking;
- Challenges: Datasets should include occlusions, cluttered backgrounds, and poses that change over time to assess the robustness of the models;
- Annotation quality: Annotations for key points and person detection in any dataset should be consistent, complete, and accurate.

Unfortunately, existing datasets only meet some of these requirements (as shown in Table 21), which can negatively impact model performance.

Table 21. Available datasets of 2D human pose estimation. The requirements, with their meaning, are listed in Section 8.1. The ✔ indicates that the dataset fully meets a requirement, whereas the ✘ mean does not. The * symbol indicates that some characteristics of requirements are met.

Dataset	Size	Data Quality	Diversity	Complexity	Challenges	Annotation Quality
LSP [42]	2 K	✔	✘	✔	*	*
LSP Extended [43]	10 K	✔	*	✔	*	*
FLIC [44]	5 K	*	*	*	✘	*
PennAction [45]	2.3 K	✔	*	✔	✔	*
JHMDB [46]	900	*	*	*	*	*
MPII [40]	30.5 K	✔	*	*	*	*
COCO [41]	200 K	✔	*	*	*	*
PoseTrack [47]	550	✔	*	*	*	*
CrowdPose [48]	80 K	*	*	✔	*	*

Most data in standard datasets are high-resolution and contain various pose activities such as running, walking, talking, and kicking. However, only a few datasets, including MPII, COCO, and JHMDB, provide occlusion annotations; others offer only key point coordinates as annotations. Some annotations in existing datasets are inaccurate, with missed or wrongly labeled key points. Images and videos in these datasets typically contain large and medium-sized human bodies. Although some datasets contain crowded

scenes with small-sized human poses, these poses often do not have labels. In other words, not all poses in images and videos are annotated. Additionally, some datasets, such as PennAction and LSP, focus on specific activities like sports, while others provide daily life activities. Most datasets also have limitations in data diversity, such as race, age, view angle, and activity.

Due to these issues with existing datasets, most studies [14,57,109] prefer to combine two or more datasets when training their models. Table 22 shows the combinations of datasets used by different studies and illustrates the pros and cons of these combination processes.

Table 22. The advantages and disadvantages of training human pose estimation models using multiple datasets.

Training Dataset	Advantage	Disadvantage	Studies
LSPE + FLIC	• A combination of single and multipose. • Learning simple and complex poses.	• Some data in LSPE dataset have incorrect annotations. • The number of key points annotations is different.	[14,83,92]
MPII + LSPE	• Exploits the scale, center, and invisible joint annotations from MPII. • Learning simple and complex poses.	• Some data in LSPE dataset have incorrect annotations.	[12,13,26,60,65,82, 87,89,91,97]
MPII + COCO	• Images are high resolution. • Exploits person boxes annotation from the COCO dataset.	• Unifying the size of the image is required. • Data augmentation is used to learn estimating hard key points.	[20,25,56,99,120, 123,132,138,145]
PennAction + JHMDB	• Increasing the size of training. • Exploits the invisible joint annotation from the PennAction dataset.	• The other data set is used for pretraining. • Varying length of video frames. • Data augmentation is needed.	[100,104,107,109, 111,112,114,115].

Combining datasets aims to address the shortcomings of individual datasets. For example, the MPII dataset is used with other datasets, such as LSPE and COCO, to exploit occluded keypoint annotations and enable models to estimate difficult joints. The LSPE dataset contains complex poses but cannot be used alone to train models. Therefore, it is combined with other datasets to teach models to predict unusual poses. Combining PennAction and JHMDB datasets increases the number of training samples for estimating single-person poses in videos. For predicting multiperson poses, the PoseTrack dataset provides sufficient data for training. Despite the characteristics of these datasets, data augmentation is still necessary to increase data diversity. Therefore, a comprehensive dataset that offers diverse data addresses the challenge of occlusions and provides precise labeling is still needed for 2D HPE tasks.

8.2. Which Loss Functions and Evaluation Criteria Are Used to Measure the Performance of Deep Learning Methods in Human Pose Estimation (RQ_2)?

As discussed in Section 5.1, several loss functions are used for HPE tasks. The choice of the loss function to optimize the network parameters depends on the model task. For example, if the model attempts to predict the location of key points (regression task), loss functions such as L_1 and L_2 are used. The difference between these losses and similar

loss functions (e.g., Smooth L_1 loss) is their sensitivity to data outliers (data with different values than other data in the same dataset). It is known that the loss function L_1 is less sensitive than L_2. Therefore, choosing the L_1 loss function in the regression task is best if the datasets contain noisy data.

On the other hand, if the model focuses on classification tasks like joint visibility or recognizing the pose behavior, loss functions such as cross-entropy and focal losses are used. Cross-entropy loss is suitable for classifying more than one class, while loss like binary cross-entropy is used to classify binary values (e.g., is the joint occluded or not). Whenever imbalanced classes exist in the dataset, the focal loss is utilized to solve such problems.

After updating the model parameters to minimize the error between predicted and ground truth values during the training mode, measuring the model's performance on unseen data is necessary. The choice of evaluation metric depends on the specific task in HPE. For example, PCP (Percentage of Correct Parts) and PCK (Percentage of Correct Key points) metrics are used to measure the performance of models estimating single poses from images [12,51,79] and videos [101,111]. In contrast, the AP (Average Precision) metric is used to measure the performance of models estimating multiperson poses from images [58,121] and video frames [18,161]. For methods that include object detection tasks in pose estimation, metrics such as IoU (Intersection over Union) are used [120,136]. Detailed information about these evaluation metrics is discussed in Section 5.2.

We used PCKh, AP, and mAP metrics in the tables in Section 7 to compare the performance of different HPE methods. Most HPE methods use these metrics to measure their model's performance, and one primary reason is that the benchmark datasets use these metrics. The other reason is [40] using metrics like PCP to measure the performance of estimating foreshortened body parts is challenging and affects model evaluation. One drawback of the PCK metric is relying on a fraction of the pose bounding box as a threshold, making this metric articulation-dependent. Therefore, Andriluka et al. [40] have modified the PCK metric to make it articulation-independent by changing the threshold from the bounding box to the length of the pose head size. They named that metric as PCKh. However, these metrics are not suitable to measure the model performance of a single pose [26] as they do not penalize false positives (the key points of the target pose and other poses are recognized as they belong to a single pose). Therefore, metrics such as AP measure the model performance that estimates multiposes.

8.3. What Are the PreTrained Models Used for Extracting the Features of the Human Pose (RQ_3)?

There are many pretrained models available to use in extracting human body features. Huge datasets like ImageNet train these models. Pretrained models such as Hourglass, ResNet, HRNet, and HrHRNet are the most used by different studies [137,143,146]. Each of these models attempts to solve specific problems. HRNet, for example, maintains the input resolution along the network structure. Hence, it solves the problem of predicting the key points' location from low-feature resolution.

On the other hand, Hourglass resolves the vanishing gradient problem by adding Intermediate supervision in every stage. Another model that provides a way to solve the vanishing gradient problem is ResNet model, which uses the skip connections between network stages. As the resolution of the input is essential to estimate the different sizes of the joints, HRNet and HrHRNet are the best choices. However, they affect the size of the HPE model and are unsuitable for real-time applications. Therefore, other pretrain models, such as GoogLeNet and MobileNet, were used but did not achieve high accuracy. More detailed information about the pretrained models was discussed in Section 6.

8.4. What Are the Existing Deep Learning Methods Applied for 2D Human Pose Estimation (RQ_4)?

Deep learning has significantly improved the performance of HPE from images and videos. After summarizing the candidate papers, we identified four types of deep learning

used for HPE: CNNs (Convolutional Neural Networks), GANs (Generative Adversarial Networks), GNNs (Graph Neural Networks), and RNNs (Recurrent Neural Networks). Most studies [14,83,116,125,137] used CNNs to estimate the person's pose from a single image due to its ability to extract the body joint's features implicitly using different kernel sizes that capture the spatial information. Popular methods that used CNNs, such as HR-Net [74] and Stacked Hourglass [85], achieved good accuracy but had high computational complexity. Therefore, techniques such as distillation were used to reduce the complexity of HPE models [57,115].

A few studies [13,100,139] used other types of deep learning to enhance HPE performance. For example, GANs were used to train models on real and fake data to address the problem of occlusions. According to Chou et al. [13], computation is unaffected because the discriminator is not included after training the model. GNNs and RNNs were also used to address the problem of occlusions, which is a significant cause of low HPE performance. GNNs were used to represent human joints as graphical representations [148] to refine human poses after predicting key points.

Similarly, RNNs were used for the same purpose, especially when the input was a sequence of images. RNNs exploit their memory component to use information from previous frames to refine poses in the current frame. Compared to CNNs for estimating human poses in videos, RNNs have lower computational complexity because they do not need to extract features from all frames [100].

Therefore, each deep learning type has its strengths and weaknesses. For example, while CNNs are excellent at extracting features relevant to the body joints, it is limited to dealing with sequential data (i.e., video frames), which is where RNN comes in. RNN is designed for capturing temporal dependencies and dealing with variable-length sequences of video frames. However, when sequences are long, they fail to estimate the human pose efficiently. On the other hand, GNN is designed to capture and represent the relationships between body joints as a graph. At the same time, GAN trains the model with synthetic data, augmenting limited annotated datasets. However, building an accurate graph using GNN is challenging, and using GAN may fail to produce diverse samples. Therefore, some studies [79,83,110] combine these deep learning types to leverage their strengths and mitigate their weaknesses.

Most HPE approaches use CNNs. For image-based single-person pose estimation, approaches may follow either regression or detection. In general, detection-based methods outperform regression-based methods (as shown in Tables 8 and 10). However, model sizes in detection-based approaches are larger than those in regression-based approaches due to the varying scales of heatmaps used in detection-based approaches. Since no additional data are required for regression-based approaches other than full/patch images, regression-based methods run faster than detection-based methods. Despite high accuracy in estimating single-person poses in images for both approaches, self-occlusions between symmetric body parts (e.g., left and right legs) still affect performance.

Estimating human poses becomes more challenging when more than one person is present in an image, and their joints must be predicted. Two approaches exist for this situation: top-down and bottom-up. As shown in Tables 14 and 16, most top-down methods outperform bottom-up methods. However, due to the person detection step, the top-down approach has a high computational time. Compared to single-person pose estimation, the accuracy of both approaches decreases when multiple people are present in an image due to challenges such as self/object occlusion, different backgrounds, varying sizes of people, and diverse clothing. Many methods for estimating multiperson poses aim to balance network size and computational speed. Therefore, accurately estimating multiperson poses in an image remains a challenging task.

The same challenges apply to video-based single/multiperson pose estimation. However, video-based methods require tracking poses across video frames. Compared to single-person pose estimation in videos (see Table 12), most methods for estimating multi-person poses in videos (see Tables 18 and 20) have lower accuracy. Unlike single-person

pose estimation, multiperson pose estimation requires additional methods to track each pose throughout the video. Possible reasons for the low accuracy of these models include frequent occlusions due to the continuous movement of people, varying scales of human bodies, and truncated body parts in some frames. These challenges remain unsolved and provide opportunities for researchers to develop suitable solutions.

The task of estimating human poses is classified as a nonlinear problem. Therefore, designing simple networks is inefficient. Many studies [89,96,110] use techniques such as multiple stages, branches, and tasks to increase the accuracy of the network. These techniques help produce multiple scales of features from multiple input resolutions, which increases the accuracy of the HPE model in predicting the location of body joints. For example, the multistage technique transfers what the previous stage learned (i.e., low-level features) to the next stage sequentially to obtain more high-level features. Figure 27 shows these techniques.

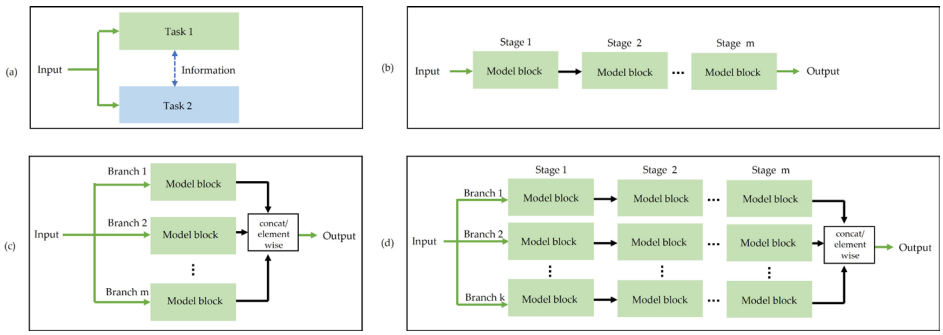


Figure 27. Techniques used in building the network of pose estimation, where (**a**) is the multitask, (**b**) is the multistage, (**c**) multibranch, and (**d**) is the multistage and branch.

Another example of shared features is the multiple-task technique in which two or more networks (e.g., one for capturing local features and the other for capturing global features) of different tasks work together to estimate the joints of human pose. The main drawback of using these techniques is increasing the model size. Therefore, a technique like distillation reduces the model's size to be usable in real-time applications. However, a limitation of the distillation technique is its effect on accuracy.

Another solution that handles the HPE challenge, especially the body occluded problem, is incorporating the graphical model into the HPE model. The graphical model is considered a traditional technique that uses the knowledge of human structure to know the relationship between the joints. However, some poses may fail to be represented by this technique. Table 23 shows the advantages and disadvantages of standard techniques used in HPE.

Table 23. Standard techniques used in human pose estimation tasks.

Technique	Advantage	Disadvantage	Example	Studies
Multistage	• Provide low and high features. • Improve accuracy. • End-to-end learning.	• Without intermediate supervision, it faces a vanishing gradient. • Adding more stages means more computational resources. • Require efficient way to recover the high-resolution.	Hourglass network	[8,13,27,52,56,65,89,94, 96,100,102,105,118,153]

Table 23. *Cont.*

Technique	Advantage	Disadvantage	Example	Studies
Multibranch	• Provide various scales of features. • Provide multiscale input resolution. • Improve accuracy. • End-to-end learning.	• A way to initialize the weights of multibranch networks is required. • Adding more branches means more computational resources.	Pyramid network	[12,14,60,83,92,95,108, 132,141,144,156]
Multistage/ branch	• Improve extraction of the feature. • Maintain the high resolution of the input. • Improve accuracy. • End-to-end learning.	• Increase the size of the model. • Increase in the computational.	High-Resolution network	[18,19,62,74,103,110, 111,117,142,143,146]
Multitask	• Shared features extraction. • Improve accuracy. • End-to-end learning.	• Increase the size of the model. • Increase in the computational.	GLPose model	[54,66,90,97,98,101,133, 136]
Graphical model	• Incorporation of prior body structure. • Handel more pose challenges. • Describe relationships between body parts.	• Limited representation. • Increase in the computational.	Graph-PoseCNN model	[12,14,83,92,93,104,124]
Part Affinity Fields	• Real-time performance. • Multiposes estimation. • Full pose estimation. • End-to-end learning.	• Ambiguity in complex poses.	OpenPose model	[25,137,138,145,157, 158]

9. Future Directions

Despite significant progress in estimating human poses from images or videos, existing deep-learning models still face challenges with accuracy and efficiency, particularly in the presence of occlusions and crowded scenes. In this section, we list several potential ideas for future directions in 2D HPE:

- While CNNs are commonly used in HPE studies [50,79,83] for their effectiveness in implicit feature extraction from images, a few studies [108,139] have explored other deep learning methods such as GANs [13], GNNs [139], and RNNs [59]. The relative performance of these methods is unclear and warrants further research.
- Many studies [12,95,96] have found that detection-based approaches outperform regression-based approaches for estimating single poses. Recently, Gu et al. [162] analyzed these two approaches to determine why detection-based methods are superior to regression-based methods. They ultimately proposed a technique that showed regression-based approaches could outperform detection-based approaches, especially when facing complex problems. Further study of this work may open new directions for estimating single-person poses;
- While many studies [88,96,111] have achieved good performance (above 90% accuracy) in estimating single-person poses from images or videos, performance significantly decreases when estimating multiperson poses due to challenges such as occlusions and varying human sizes. Various studies [18,144,159] have proposed methodologies to address these challenges, but finding the best solutions remains an open problem;

- Optical flow has been used by some studies [53,107] to track motion in videos. However, it is easily affected by noise and can have difficulty tracking human motion in noisy environments. To improve performance, a few works [100,111] have replaced optical flow with other techniques, such as RNNs or temporal consistency. Focusing on such techniques may further boost performance;
- Many studies [119,124] use post-processing steps such as search algorithms or graphical models to group predicted key points into individual humans in bottom-up approaches. However, some recent works [58,139] have incorporated graphical information into neural networks to make the training process differentiable. This area warrants further investigation;
- Improving the efficiency of HPE tasks is not limited to enhancing models; dataset labels also play a significant role. In addition to keypoint position labels, only a few datasets [40,45] provide additional labels, such as visibility of body joints, that can help address the challenge of occlusions. As occlusion is one of the main challenges in 2D HPE, researchers need to increase the number of occluded labels in datasets. Unsupervised/semi-supervised and data augmentation methods are currently used to address this limitation;
- Another challenge in 2D HPE is crowded scenes. Only a few datasets provide data with crowded scenarios (e.g., CrowdPose and COCO), and their data consist only of images. Recently, a dataset called HAJJv2 [163] was introduced that provides more than 290,000 videos for detecting abnormal behaviors during Hajj religious events. The data in this dataset are diverse in terms of race, as many people from all over the world [164,165] perform Hajj rituals. They also have a large crowd scale, providing nine classes with normal and abnormal behaviors for each category. This dataset may help train 2D HPE models;
- An excellent example of research attempting to solve the problem of HPE in crowded scenes is research on Hajj and Umrah events [3,163,166], where more than 20 people must be detected, estimated, and classified in real-time. These types of research heavily rely on HPE techniques. For example, models such as YOLO and OpenPose are used for detecting and estimating poses to identify suspicious behavior during Hajj events. However, these models still face challenges in handling large numbers of poses in real-time. Developing methods to address this problem remains an open challenge.

10. Conclusions

This review systematically reviewed state-of-the-art deep learning types and techniques in 2D human pose estimation (HPE). We collected different articles published between 2014 and 2023 that interested 2D HPE. We selected and summarized 107 articles according to several criteria that we listed. Then, we classified their methodology based on the number of people estimated (single-person or multiple persons pose estimation) and the input type (image- or video-based). Under each of these classifications, many methods were present. Therefore, we grouped these methods according to their general approach.

These approaches are regression-based and detection-based, targeted to estimate a single pose in the image, and top-down and bottom-up approaches, targeted to estimate multiple poses from the image. Each approach proposes a general framework for solving HPE challenges like occluded and invisible joints. While the methods that follow the regression-based and detection-based approaches achieved an excellent performance in estimating single pose, the methods in top-down and bottom-up approaches still face challenges in increasing the accuracy. One reason is not knowing the number of people, as they must be detected first by using the pose detector (top-down approach) or discovering all the joints in the image and then grouping them for the appropriate individual (bottom-up approach). The other reason is the crowd scene's high occlusion between body joints. We found the Stacked Hourglass, HRNet, DARK, CPM, and OpenPose models are the most well-known models that significantly boost the accuracy of estimating single and multipose images. While designing the network architecture using techniques like multistage (e.g.,

Stacked Hourglass) and multibranch (e.g., Pyramid network) increased the accuracy of the HPE model, it is not suitable for real-time applications. Many articles focused more on improving the model's accuracy than reducing the model size. Therefore, designing a network that aims to increase the HPE model's efficiency is one area that needs more research.

One main contribution of our systematic review is the summary of different methods that focused on estimating single and multiple poses from videos. Like image-based methods, we grouped the various video-based methods according to their general approach. These approaches are temporal cues, offline, and online approaches. The first approach estimates a single pose through propagated temporal clues to track the human pose. In contrast, the last two approaches estimate multiple poses in video frames. We found the RNN was the most popular deep learning type used for estimating the pose from the target frame by combining the features from other frames, resulting in a robust performance. Different video-based models also used the graphical model.

In contrast to the offline approach, which is accurate in estimating the poses of many subjects but has a high time complexity, the online approach aims to estimate many people and maintain a balance between accuracy and time complexity. Many studies used lightweight methods such as PAF (part affinity fields) to achieve real-time performance. However, the analysis shows that estimating multiple poses in real-time from video (online approach) is still challenging.

In addition to analyzing different 2D HPE methodologies, we also examined the main components of any HPE model, including the existing datasets, loss functions, evaluation metrics, and pretrained features extraction model. The results show LSP, COCO, MPII, and PennAction are the most used datasets for training different models. These datasets have standard evaluation metrics such as PCKh, AP, and their variation is used to evaluate the model's performance. Our analysis shows that most of the studies train their models using two or more datasets to generalize the model in dealing with data. Because HPE used two tasks, classification and regression, different loss functions, such as Log and focal losses, are used. Furthermore, our review listed the standard feature extraction models like HRNet used as the backbone of the HPE model.

Our analysis found that CNN and RNN are the most common types of deep learning used in 2D HPE. CNN works well in detecting human body joints from a single image. However, it fails to capture the temporal clues. Therefore, RNN is used for estimating the human pose if the input is video. Other types, like GAN and GNN, need to be investigated more.

Additionally, the approaches that hardly maintain the performance between accuracy and efficiency must be improved. Since the occlusion and the crowded scenario are still the main challenges in HPE, using datasets with crowd scenes to train the model may solve these problems. Our review has considered these points as potential future directions for HPE.

Finally, as the analysis of the collecting process shows a significant increase in publishing articles annually, we found that the number of articles interested in estimating the poses from images is higher than estimating the poses from videos. Therefore, more research on human pose estimation in a video is required. In addition, building an accurate and small size of the network is still an open area challenge.

Author Contributions: E.S. and M.A. (Muhammad Arif) contributed to the whole process of writing the paper, including conceptualization, methodology, critical analysis of selected papers, writing, and revisions. M.A. (Manal Alghamdi) contributed to the critical analysis of papers, writing, and revisions. M.A.A.G. contributed to writing, critical analysis and revisions. All authors have read and agreed to the published version of the manuscript.

Funding: Deputyship for Research & Innovation, Ministry of Education in Saudi Arabia through the project number: IFP22UQU4250002DSR226.

Data Availability Statement: No new data were created or analyzed in this study. Data sharing is not applicable to this article.

Acknowledgments: The authors extend their appreciation to the Deputyship for Research & Innovation, Ministry of Education in Saudi Arabia for funding this research work through the project number: IFP22UQU4250002DSR226.

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

References

1. Sun, J.; Chen, X.; Lu, Y.; Cao, J. 2D Human Pose Estimation from Monocular Images: A Survey. In Proceedings of the IEEE 3rd International Conference on Computer and Communication Engineering Technology, Beijing, China, 14–16 August 2020; pp. 111–121.
2. Gong, W.; Zhang, X.; Gonzàlez, J.; Sobral, A.; Bouwmans, T.; Tu, C.; Zahzah, E.H. Human pose estimation from monocular images: A comprehensive survey. *Sensors* **2016**, *16*, 1966. [CrossRef]
3. Miao, Y.; Yang, J.; Alzahrani, B.; Lv, G.; Alafif, T.; Barnawi, A.; Chen, M. Abnormal Behavior Learning Based on Edge Computing toward a Crowd Monitoring System. *IEEE Netw.* **2022**, *36*, 90–96. [CrossRef]
4. Pardos, A.; Menychtas, A.; Maglogiannis, I. On unifying deep learning and edge computing for human motion analysis in exergames development. *Neural Comput. Appl.* **2022**, *34*, 951–967. [CrossRef]
5. Kumarapu, L.; Mukherjee, P. Animepose: Multi-person 3d pose estimation and animation. *Pattern Recognit. Lett.* **2021**, *147*, 16–24. [CrossRef]
6. Khan, M.A. Multiresolution coding of motion capture data for real-time multimedia applications. *Multimed. Tools Appl.* **2017**, *76*, 16683–16698. [CrossRef]
7. Lonini, L.; Moon, Y.; Embry, K.; Cotton, R.J.; McKenzie, K.; Jenz, S.; Jayaraman, A. Video-based pose estimation for gait analysis in stroke survivors during clinical assessments: A proof-of-concept study. *Digit. Biomark.* **2022**, *6*, 9–18. [CrossRef]
8. Ludwig, K.; Scherer, S.; Einfalt, M.; Lienhart, R. Self-supervised learning for human pose estimation in sports. In Proceedings of the IEEE International Conference on Multimedia & Expo Workshops, Shenzhen, China, 5–9 July 2021; pp. 1–6.
9. Gamra, M.B.; Akhloufi, M.A. A review of deep learning techniques for 2D and 3D human pose estimation. *Image Vis. Comput.* **2021**, *114*, 104282. [CrossRef]
10. Li, T.; Yu, H. Visual-Inertial Fusion-Based Human Pose Estimation: A Review. *IEEE Trans. Instrum. Meas.* **2023**, *72*, 1–16. [CrossRef]
11. Nguyen, H.C.; Nguyen, T.H.; Scherer, R.; Le, V.H. Unified end-to-end YOLOv5-HR-TCM framework for automatic 2D/3D human pose estimation for real-time applications. *Sensors* **2022**, *22*, 5419. [CrossRef]
12. Bin, Y.; Chen, Z.M.; Wei, X.S.; Chen, X.; Gao, C.; Sang, N. Structure-aware human pose estimation with graph convolutional networks. *Pattern Recognit.* **2020**, *106*, 107410. [CrossRef]
13. Chou, C.J.; Chien, J.T.; Chen, H.T. Self adversarial training for human pose estimation. In Proceedings of the Asia-Pacific Signal and Information Processing Association Annual Summit and Conference, Honolulu, HI, USA, 12–15 November 2018; pp. 17–30.
14. Fan, X.; Zheng, K.; Lin, Y.; Wang, S. Combining local appearance and holistic view: Dual-source deep neural networks for human pose estimation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 1347–1355.
15. Liu, Z.; Zhu, J.; Bu, J.; Chen, C. A survey of human pose estimation: The body parts parsing based methods. *J. Vis. Commun. Image Represent.* **2015**, *32*, 10–19. [CrossRef]
16. Alsubait, T.; Sindi, T.; Alhakami, H. Classification of the Human Protein Atlas Single Cell Using Deep Learning. *Appl. Sci.* **2022**, *12*, 11587. [CrossRef]
17. Toshev, A.; Szegedy, C. Deeppose: Human pose estimation via deep neural networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014; pp. 1653–1660.
18. Liu, Z.; Chen, H.; Feng, R.; Wu, S.; Ji, S.; Yang, B.; Wang, X. Deep dual consecutive network for human pose estimation. In Proceedings of the IEEE Conference on European Conference on Computer Vision, Nashville, TN, USA, 20–25 June 2021; pp. 525–534.
19. Wang, M.; Tighe, J.; Modolo, D. Combining detection and tracking for human pose estimation in videos. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 11088–11096.
20. Zhang, F.; Zhu, X.; Dai, H.; Ye, M.; Zhu, C. Distribution-aware coordinate representation for human pose estimation. In Proceedings of the IEEE Conference on European Conference on Computer Vision, Seattle, WA, USA, 13–19 June 2020; pp. 7093–7102.
21. Moeslund, T.B.; Granum, E. A Survey of Computer Vision-Based Human Motion Capture. *Comput. Vis. Image Underst.* **2001**, *81*, 231–268. [CrossRef]
22. Moeslund, T.B.; Hilton, A.; Krüger, V. A survey of advances in vision-based human motion capture and analysis. *Comput. Vis. Image Underst.* **2006**, *104*, 90–126. [CrossRef]
23. Perez-Sala, X.; Escalera, S.; Angulo, C.; Gonzàlez, J. A Survey on Model Based Approaches for 2D and 3D Visual Human Pose Recovery. *Sensors* **2014**, *14*, 4189–4210. [CrossRef]
24. Dubey, S.; Dixit, M. A comprehensive survey on human pose estimation approaches. *Multimed. Syst.* **2023**, *29*, 167–195. [CrossRef]

25. Cao, Z.; Simon, T.; Wei, S.E.; Sheikh, Y. Realtime multi-person 2d pose estimation using part affinity fields. In Proceedings of the IEEE Conference on European Conference on Computer Vision, Honolulu, HI, USA, 21–26 July 2017; pp. 7291–7299.
26. Pishchulin, L.; Insafutdinov, E.; Tang, S.; Andres, B.; Andriluka, M.; Gehler, P.V.; Schiele, B. Deepcut: Joint subset partition and labeling for multi person pose estimation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 4929–4937.
27. He, K.; Gkioxari, G.; Dollár, P.; Girshick, R. Mask r-cnn. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Yibin, China, 22–29 October 2017; pp. 2961–2969.
28. Dang, Q.; Yin, J.; Wang, B.; Zheng, W. Deep learning based 2D human pose estimation: A survey. *Tsinghua Sci. Technol.* **2019**, *24*, 663–676. [CrossRef]
29. Song, L.; Yu, G.; Yuan, J.; Liu, Z. Human pose estimation and its application to action recognition: A survey. *J. Vis. Commun. Image Represent.* **2021**, *76*, 103055. [CrossRef]
30. Munea, T.L.; Jembre, Y.Z.; Weldegebriel, H.T.; Chen, L.; Huang, C.; Yang, C. The progress of human pose estimation: A survey and taxonomy of models applied in 2D human pose estimation. *IEEE Access* **2020**, *8*, 133330–133348. [CrossRef]
31. Chen, Y.; Tian, Y.; He, M. Monocular human pose estimation: A survey of deep learning-based methods. *Comput. Vis. Image Underst.* **2020**, *192*, 102897. [CrossRef]
32. Toshpulatov, M.; Lee, W.; Lee, S.; Haghighian Roudsari, A. Human pose, hand and mesh estimation using deep learning: A survey. *J. Supercomput.* **2022**, *78*, 7616–7654. [CrossRef]
33. Liu, W.; Bao, Q.; Sun, Y.; Mei, T. Recent advances of monocular 2D and 3D human pose estimation: A deep learning perspective. *ACM Comput. Surv.* **2022**, *55*, 1–41. [CrossRef]
34. Zheng, C.; Wu, W.; Chen, C.; Yang, T.; Zhu, S.; Shen, J.; Kehtarnavaz, N.; Shah, M. Deep Learning-Based Human Pose Estimation: A Survey. *J. ACM* **2023**, *37*, 35. [CrossRef]
35. Lan, G.; Wu, Y.; Hu, F.; Hao, Q. Vision-Based Human Pose Estimation via Deep Learning: A Survey. *IEEE Trans. Hum.-Mach. Syst.* **2023**, *53*, 253–268. [CrossRef]
36. dos Reis, E.S.; Seewald, L.A.; Antunes, R.S.; Rodrigues, V.F.; da Rosa Righi, R.; da Costa, C.A.; da Silveira, L.G., Jr.; Eskofier, B.; Maier, A.; Horz, T.; et al. Monocular multi-person pose estimation: A survey. *Pattern Recognit.* **2021**, *118*, 108046. [CrossRef]
37. Badiola-Bengoa, A.; Mendez-Zorrilla, A. A Systematic Review of the Application of Camera-Based Human Pose Estimation in the Field of Sport and Physical Exercise. *Sensors* **2021**, *21*, 5996. [CrossRef]
38. Difini, G.M.; Martins, M.G.; Barbosa, J.L.V. Human pose estimation for training assistance: A systematic literature review. In Proceedings of the Multimedia and the Web, Belo, Brazil, 5–12 November 2021; pp. 189–196.
39. Topham, L.; Khan, W.; Al-Jumeily, D.; Hussain, A. Human Body Pose Estimation for Gait Identification: A Comprehensive Survey of Datasets and Models. *ACM Comput. Surv.* **2022**, *55*, 1–42. [CrossRef]
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 IEEE Conference on European Conference on Computer Vision, Columbus, OH, USA, 23–28 June 2014; pp. 3686–3693.
41. 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 European Conference on Computer Vision, Zurich, Switzerland, 6–12 September 2014; pp. 740–755.
42. Johnson, S.; Everingham, M. Clustered pose and nonlinear appearance models for human pose estimation. In Proceedings of the British Machine Vision Conference, Aberystwyth, UK, 31 August–3 September 2010; Volume 2, p. 5.
43. Johnson, S.; Everingham, M. Learning effective human pose estimation from inaccurate annotation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Colorado Springs, CO, USA, 20–25 June 2011; pp. 1465–1472.
44. Sapp, B.; Taskar, B. Modec: Multimodal decomposable models for human pose estimation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Portland, OR, USA, 23–28 June 2013; pp. 3674–3681.
45. Zhang, W.; Zhu, M.; Derpanis, K.G. From actemes to action: A strongly-supervised representation for detailed action understanding. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Sydney, Australia, 1–8 December 2013; pp. 2248–2255.
46. Jhuang, H.; Gall, J.; Zuffi, S.; Schmid, C.; Black, M.J. Towards understanding action recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Sydney, Australia, 1–8 December 2013; pp. 3192–3199.
47. Andriluka, M.; Iqbal, U.; Insafutdinov, E.; Pishchulin, L.; Milan, A.; Gall, J.; Schiele, B. Posetrack: A benchmark for human pose estimation and tracking. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 5167–5176.
48. Li, J.; Wang, C.; Zhu, H.; Mao, Y.; Fang, H.S.; Lu, C. Crowdpose: Efficient crowded scenes pose estimation and a new benchmark. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 10863–10872.
49. Doering, A.; Chen, D.; Zhang, S.; Schiele, B.; Gall, J. Posetrack21: A dataset for person search, multi-object tracking and multi-person pose tracking. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 20963–20972.
50. Zhang, F.; Zhu, X.; Ye, M. Fast human pose estimation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 3517–3526.

51. Carreira, J.; Agrawal, P.; Fragkiadaki, K.; Malik, J. Human pose estimation with iterative error feedback. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 4733–4742.
52. Girdhar, R.; Gkioxari, G.; Torresani, L.; Paluri, M.; Tran, D. Detect-and-track: Efficient pose estimation in videos. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 350–359.
53. Pfister, T.; Charles, J.; Zisserman, A. Flowing convnets for human pose estimation in videos. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Santiago, Chile, 7–13 December 2015; pp. 1913–1921.
54. Wang, X.; Gao, L.; Dai, Y.; Zhou, Y.; Song, J. Semantic-aware transfer with instance-adaptive parsing for crowded scenes pose estimation. In Proceedings of the ACM International Conference on Multimedia, Chengdu, China, 20–24 October 2021; pp. 686–694.
55. Moon, G.; Chang, J.Y.; Lee, K.M. Posefix: Model-agnostic general human pose refinement network. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 7773–7781.
56. Ke, L.; Chang, M.C.; Qi, H.; Lyu, S. Multi-scale structure-aware network for human pose estimation. In Proceedings of the European Conference on Computer Vision, Munich, Germany, 8–14 September 2018; pp. 713–728.
57. Papaioannidis, C.; Mademlis, I.; Pitas, I. Fast CNN-based Single-Person 2D Human Pose Estimation for Autonomous Systems. *IEEE Trans. Circuits Syst. Video Technol.* **2022**, *33*, 1262–1275. [CrossRef]
58. Brasó, G.; Kister, N.; Leal-Taixé, L. The center of attention: Center-keypoint grouping via attention for multi-person pose estimation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Montreal, QC, Canada, 10–17 October 2021; pp. 11853–11863.
59. Belagiannis, V.; Zisserman, A. Recurrent human pose estimation. In Proceedings of the IEEE International Conference on Automatic Face & Gesture Recognition, Washington, DC, USA, 30 May–3 June 2017; pp. 468–475.
60. Zhou, L.; Chen, Y.; Gao, Y.; Wang, J.; Lu, H. Occlusion-aware siamese network for human pose estimation. In Proceedings of the European Conference on Computer Vision, Glasgow, UK, 23–28 August 2020; pp. 396–412.
61. Lin, T.Y.; Goyal, P.; Girshick, R.; He, K.; Dollár, P. Focal loss for dense object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Venice, Italy, 22–29 October 2017; pp. 2980–2988.
62. Chen, Y.; Wang, Z.; Peng, Y.; Zhang, Z.; Yu, G.; Sun, J. Cascaded pyramid network for multi-person pose estimation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 7103–7112.
63. Munea, T.L.; Yang, C.; Huang, C.; Elhassan, M.A.; Zhen, Q. SimpleCut: A simple and strong 2D model for multi-person pose estimation. *Comput. Vis. Image Underst.* **2022**, *222*, 103509. [CrossRef]
64. Nguyen, H.C.; Nguyen, T.H.; Nowak, R.; Byrski, J.; Siwocha, A.; Le, V.H. Combined YOLOv5 and HRNet for high accuracy 2D keypoint and human pose estimation. *J. Artif. Intell. Soft Comput. Res.* **2022**, *12*, 281–298. [CrossRef]
65. Dong, X.; Yu, J.; Zhang, J. Joint usage of global and local attentions in hourglass network for human pose estimation. *Neurocom-Puting* **2022**, *472*, 95–102. [CrossRef]
66. Li, S.; Liu, Z.Q.; Chan, A.B. Heterogeneous multi-task learning for human pose estimation with deep convolutional neural network. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014; pp. 482–489.
67. Arulprakash, E.; Aruldoss, M. A study on generic object detection with emphasis on future research directions. *J. King Saud Univ.-Comput. Inf. Sci.* **2022**, *34*, 7347–7365. [CrossRef]
68. Aly, S.; Gutub, A. Intelligent recognition system for identifying items and pilgrims. *NED Univ. J. Res.* **2018**, *15*, 17–23.
69. Desai, M.M.; Mewada, H.K. Review on Human Pose Estimation and Human Body Joints Localization. *Int. J. Comput. Digit. Syst.* **2021**, *10*, 883–898. [CrossRef]
70. Elharrouss, O.; Akbari, Y.; Almaadeed, N.; Al-Maadeed, S. Backbones-review: Feature extraction networks for deep learning and deep reinforcement learning approaches. *arXiv* **2022**, arXiv:2206.08016.
71. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. *Commun. ACM* **2017**, *60*, 84–90. [CrossRef]
72. Nguyen, T.D.; Kresovic, M. A survey of top-down approaches for human pose estimation. *arXiv* **2022**, arXiv:2202.02656.
73. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv* **2014**, arXiv:1409.1556.
74. Sun, K.; Xiao, B.; Liu, D.; Wang, J. Deep high-resolution representation learning for human pose estimation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 5693–5703.
75. 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, 27–30 June 2016; pp. 770–778.
76. Xie, S.; Girshick, R.; Dollár, P.; Tu, Z.; He, K. Aggregated residual transformations for deep neural networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 1492–1500.
77. 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.
78. Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; Rabinovich, A. Going deeper with convolutions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 1–9.
79. Sun, X.; Shang, J.; Liang, S.; Wei, Y. Compositional human pose regression. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Venice, Italy, 22–29 October 2017; pp. 2602–2611.

80. Luvizon, D.C.; Tabia, H.; Picard, D. Human pose regression by combining indirect part detection and contextual information. *Comput. Graph.* **2019**, *85*, 15–22. [CrossRef]
81. Li, J.; Bian, S.; Zeng, A.; Wang, C.; Pang, B.; Liu, W.; Lu, C. Human pose regression with residual log-likelihood estimation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Montreal, QC, Canada, 10–17 October 2021; pp. 11025–11034.
82. Shamsafar, F.; Ebrahimnezhad, H. Uniting holistic and part-based attitudes for accurate and robust deep human pose estimation. *J. Ambient. Intell. Humaniz. Comput.* **2021**, *12*, 2339–2353. [CrossRef]
83. Tompson, J.J.; Jain, A.; LeCun, Y.; Bregler, C. Joint training of a convolutional network and a graphical model for human pose estimation. *Adv. Neural Inf. Process. Syst.* **2014**, *27*, 1799–1807.
84. Chen, H.; Feng, R.; Wu, S.; Xu, H.; Zhou, F.; Liu, Z. 2D Human pose estimation: A survey. *Multimed. Syst.* **2022**, *29*, 3115–3138. [CrossRef]
85. Newell, A.; Yang, K.; Deng, J. Stacked hourglass networks for human pose estimation. In Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 11–14 October 2016; Springer: Berlin/Heidelberg, Germany, 2016; pp. 483–499.
86. Hua, G.; Li, L.; Liu, S. Multipath affinage stacked—Hourglass networks for human pose estimation. *Front. Comput. Sci.* **2020**, *14*, 1–12. [CrossRef]
87. Yang, W.; Li, S.; Ouyang, W.; Li, H.; Wang, X. Learning feature pyramids for human pose estimation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Venice, Italy, 22–29 October 2017; pp. 1281–1290.
88. Tian, Y.; Hu, W.; Jiang, H.; Wu, J. Densely connected attentional pyramid residual network for human pose estimation. *Neurocomputing* **2019**, *347*, 13–23. [CrossRef]
89. Wei, S.E.; Ramakrishna, V.; Kanade, T.; Sheikh, Y. Convolutional pose machines. In Proceedings of the IEEE conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 4724–4732.
90. Hwang, J.; Park, S.; Kwak, N. Athlete pose estimation by a global-local network. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 58–65.
91. Lifshitz, I.; Fetaya, E.; Ullman, S. Human pose estimation using deep consensus voting. In Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 11–14 October 2016; pp. 246–260.
92. Chen, X.; Yuille, A.L. Articulated pose estimation by a graphical model with image dependent pairwise relations. *Adv. Neural Inf. Process. Syst.* **2014**, *27*, 1736–1744.
93. Fu, L.; Zhang, J.; Huang, K. ORGM: Occlusion relational graphical model for human pose estimation. *IEEE Trans. Image Process.* **2016**, *26*, 927–941. [CrossRef]
94. Tang, W.; Yu, P.; Wu, Y. Deeply learned compositional models for human pose estimation. In Proceedings of the European Conference on Computer Vision, Munich, Germany, 8–14 September 2018; pp. 190–206.
95. Tang, W.; Wu, Y. Does learning specific features for related parts help human pose estimation? In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 1107–1116.
96. Su, Z.; Ye, M.; Zhang, G.; Dai, L.; Sheng, J. Cascade feature aggregation for human pose estimation. *arXiv* **2019**, arXiv:1902.07837.
97. Chen, Y.; Shen, C.; Wei, X.S.; Liu, L.; Yang, J. Adversarial posenet: A structure-aware convolutional network for human pose estimation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Venice, Italy, 22–29 October 2017; pp. 1212–1221.
98. Shamsolmoali, P.; Zareapoor, M.; Zhou, H.; Yang, J. Amil: Adversarial multi-instance learning for human pose estimation. *ACM Trans. Multimed. Comput. Commun. Appl.* **2020**, *16*, 1–23. [CrossRef]
99. Dai, H.; Shi, H.; Liu, W.; Wang, L.; Liu, Y.; Mei, T. FasterPose: A faster simple baseline for human pose estimation. *ACM Trans. Multimed. Comput. Commun. Appl.* **2022**, *18*, 1–16. [CrossRef]
100. Fan, Z.; Liu, J.; Wang, Y. Motion adaptive pose estimation from compressed videos. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Montreal, QC, Canada, 10–17 October 2021; pp. 11719–11728.
101. Jiao, Y.; Chen, H.; Feng, R.; Chen, H.; Wu, S.; Yin, Y.; Liu, Z. GLPose: Global-Local Representation Learning for Human Pose Estimation. *ACM Trans. Multimed. Comput. Commun. Appl.* **2022**, *18*, 1–16. [CrossRef]
102. Pfister, T.; Simonyan, K.; Charles, J.; Zisserman, A. Deep convolutional neural networks for efficient pose estimation in gesture videos. In Proceedings of the Asian Conference on Computer Vision, Singapore, 1–5 November 2014; pp. 538–552.
103. Jain, A.; Tompson, J.; LeCun, Y.; Bregler, C. Modeep: A deep learning framework using motion features for human pose estimation. In Proceedings of the Asian Conference on Computer Vision, Singapore, 1–5 November 2014; pp. 302–315.
104. Xiaohan Nie, B.; Xiong, C.; Zhu, S.C. Joint action recognition and pose estimation from video. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 1293–1301.
105. Liu, S.; Li, Y.; Hua, G. Human pose estimation in video via structured space learning and halfway temporal evaluation. *IEEE Trans. Circuits Syst. Video Technol.* **2018**, *29*, 2029–2038. [CrossRef]
106. Charles, J.; Pfister, T.; Magee, D.; Hogg, D.; Zisserman, A. Personalizing human video pose estimation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 3063–3072.
107. Song, J.; Wang, L.; Van Gool, L.; Hilliges, O. Thin-slicing network: A deep structured model for pose estimation in videos. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 4220–4229.

108. Gkioxari, G.; Toshev, A.; Jaitly, N. Chained predictions using convolutional neural networks. In Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 11–14 October 2016; Springer: Berlin/Heidelberg, Germany, 2016; pp. 728–743.
109. Luo, Y.; Ren, J.; Wang, Z.; Sun, W.; Pan, J.; Liu, J.; Pang, J.; Lin, L. LSTM Pose Machines. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 5207–5215.
110. Artacho, B.; Savakis, A. Unipose: Unified human pose estimation in single images and videos. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 7035–7044.
111. Li, Y.; Li, K.; Wang, X.; Da Xu, R.Y. Exploring temporal consistency for human pose estimation in videos. *Pattern Recognit.* **2020**, *103*, 107258. [CrossRef]
112. Zhang, Y.; Wang, Y.; Camps, O.; Sznaier, M. Key frame proposal network for efficient pose estimation in videos. In *Proceedings of the European Conference on Computer Vision*; Springer: Glasgow, UK, 2020; pp. 609–625.
113. Schmidtke, L.; Vlontzos, A.; Ellershaw, S.; Lukens, A.; Arichi, T.; Kainz, B. Unsupervised human pose estimation through transforming shape templates. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 2484–2494.
114. Ma, X.; Rahmani, H.; Fan, Z.; Yang, B.; Chen, J.; Liu, J. Remote: Reinforced motion transformation network for semi-supervised 2d pose estimation in videos. In Proceedings of the Conference on Artificial Intelligence, Palo Alto, CA, USA, 22 February–1 March 2022; Volume 36, pp. 1944–1952.
115. Nie, X.; Li, Y.; Luo, L.; Zhang, N.; Feng, J. Dynamic kernel distillation for efficient pose estimation in videos. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 6942–6950.
116. Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Rich feature hierarchies for accurate object detection and semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014; pp. 580–587.
117. Moon, G.; Chang, J.Y.; Lee, K.M. Multi-scale Aggregation R-CNN for 2D Multi-person Pose Estimation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 1–9.
118. Girshick, R. Fast r-cnn. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Santiago, Chile, 7–13 December 2015; pp. 1440–1448.
119. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster r-cnn: Towards real-time object detection with region proposal networks. *Adv. Neural Inf. Process. Syst.* **2015**, *28*, 91–99. [CrossRef] [PubMed]
120. Fang, H.S.; Xie, S.; Tai, Y.W.; Lu, C. Rmpe: Regional multi-person pose estimation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Venice, Italy, 22–29 October 2017; pp. 2334–2343.
121. Huang, J.; Zhu, Z.; Guo, F.; Huang, G. The devil is in the details: Delving into unbiased data processing for human pose estimation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 5700–5709.
122. Li, W.; Wang, Z.; Yin, B.; Peng, Q.; Du, Y.; Xiao, T.; Yu, G.; Lu, H.; Wei, Y.; Sun, J. Rethinking on multi-stage networks for human pose estimation. *arXiv* **2019**, arXiv:1901.00148.
123. Xie, R.; Wang, C.; Zeng, W.; Wang, Y. An empirical study of the collapsing problem in semi-supervised 2d human pose estimation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Montreal, QC, Canada, 10–17 October 2021; pp. 11240–11249.
124. 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 European Conference on Computer Vision, Glasgow, UK, 23–28 August 2020; pp. 492–508.
125. Cai, Y.; Wang, Z.; Luo, Z.; Yin, B.; Du, A.; Wang, H.; Zhang, X.; Zhou, X.; Zhou, E.; Sun, J. Learning delicate local representations for multi-person pose estimation. In Proceedings of the European Conference on Computer Vision, Glasgow, UK, 23–28 August 20200; pp. 455–472.
126. Qiu, Z.; Yang, Q.; Wang, J.; Wang, X.; Xu, C.; Fu, D.; Yao, K.; Han, J.; Ding, E.; Wang, J. Learning Structure-Guided Diffusion Model for 2D Human Pose Estimation. *arXiv* **2023**, arXiv:2306.17074.
127. Yuan, Y.; Rao, F.; Lang, H.; Lin, W.; Zhang, C.; Chen, X.; Wang, J. Hrformer: High-resolution transformer for dense prediction. *arXiv* **2021**, arXiv:2110.09408.
128. 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 IEEE Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 10440–10450.
129. Li, Y.; Liu, R.; Wang, X.; Wang, R. Human pose estimation based on lightweight basicblock. *Mach. Vis. Appl.* **2023**, *34*, 3. [CrossRef]
130. Wang, H.; Liu, J.; Tang, J.; Wu, G. Lightweight Super-Resolution Head for Human Pose Estimation. In Proceedings of the 31st ACM International Conference on Multimedia, Ottawa, ON, Canada, 29 October–3 November 2023; pp. 2353–2361.
131. McNally, W.; Vats, K.; Wong, A.; McPhee, J. EvoPose2D: Pushing the boundaries of 2d human pose estimation using accelerated neuroevolution with weight transfer. *IEEE Access* **2021**, *9*, 139403–139414. [CrossRef]
132. Xu, D.; Zhang, R.; Guo, L.; Feng, C.; Gao, S. LDNet: Lightweight dynamic convolution network for human pose estimation. *Adv. Eng. Inform.* **2022**, *54*, 101785. [CrossRef]
133. Xu, L.; Jin, S.; Liu, W.; Qian, C.; Ouyang, W.; Luo, P.; Wang, X. Zoomnas: Searching for whole-body human pose estimation in the wild. *IEEE Trans. Pattern Anal. Mach. Intell.* **2022**, *45*, 5296–5313. [CrossRef] [PubMed]

134. Insafutdinov, E.; Pishchulin, L.; Andres, B.; Andriluka, M.; Schiele, B. Deepercut: A deeper, stronger, and faster multi-person pose estimation model. In Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 11–14 October 2016; pp. 34–50.
135. Varadarajan, S.; Datta, P.; Tickoo, O. A greedy part assignment algorithm for real-time multi-person 2D pose estimation. In Proceedings of the IEEE Winter Conference on Applications of Computer Vision, Lake Tahoe, NV, USA, 12–15 March 2018; pp. 418–426.
136. Kocabas, M.; Karagoz, S.; Akbas, E. Multiposenet: Fast multi-person pose estimation using pose residual network. In Proceedings of the European Conference on Computer Vision, Munich, Germany, 8–14 September 2018; pp. 417–433.
137. Kreiss, S.; Bertoni, L.; Alahi, A. Pifpaf: Composite fields for human pose estimation. In Proceedings of the IEEE Conference on European Conference on Computer Vision, Long Beach, CA, USA, 16–20 June 2019; pp. 11977–11986.
138. Nasr, M.; Ayman, H.; Ebrahim, N.; Osama, R.; Mosaad, N.; Mounir, A. Realtime multi-person 2D pose estimation. *Int. J. Adv. Netw. Appl.* **2020**, *11*, 4501–4508. [CrossRef]
139. Jin, S.; Liu, W.; Xie, E.; Wang, W.; Qian, C.; Ouyang, W.; Luo, P. Differentiable hierarchical graph grouping for multi-person pose estimation. In Proceedings of the European Conference on Computer Vision, Glasgow, UK, 23–28 August 2020; pp. 718–734.
140. Jin, L.; Wang, X.; Nie, X.; Liu, L.; Guo, Y.; Zhao, J. Grouping by Center: Predicting Centripetal Offsets for the bottom-up human pose estimation. *IEEE Trans. Multimed.* **2022**, *25*, 3364–3374. [CrossRef]
141. Du, C.; Yan, Z.; Yu, H.; Yu, L.; Xiong, Z. Hierarchical Associative Encoding and Decoding for Bottom-Up Human Pose Estimation. *IEEE Trans. Circuits Syst. Video Technol.* **2022**, *33*, 1762–1775. [CrossRef]
142. Nie, X.; Feng, J.; Xing, J.; Yan, S. Pose partition networks for multi-person pose estimation. In Proceedings of the European Conference on Computer Vision, Munich, Germany, 8–14 September 2018; pp. 684–699.
143. Cheng, B.; Xiao, B.; Wang, J.; Shi, H.; Huang, T.S.; Zhang, L. Higherhrnet: Scale-aware representation learning for bottom-up human pose estimation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 5386–5395.
144. Luo, Z.; Wang, Z.; Huang, Y.; Wang, L.; Tan, T.; Zhou, E. Rethinking the heatmap regression for bottom-up human pose estimation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 13264–13273.
145. Hidalgo, G.; Raaj, Y.; Idrees, H.; Xiang, D.; Joo, H.; Simon, T.; Sheikh, Y. Single-network whole-body pose estimation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Seoul, Republic of Korea, 15–20 June 2019; pp. 6982–6991.
146. Zhao, Y.; Luo, Z.; Quan, C.; Liu, D.; Wang, G. Cluster-wise learning network for multi-person pose estimation. *Pattern Recognit.* **2020**, *98*, 107074. [CrossRef]
147. Zhou, C.; Ren, Z.; Hua, G. Temporal keypoint matching and refinement network for pose estimation and tracking. In Proceedings of the European Conference on Computer Vision, Glasgow, UK, 23–28 August 2020; pp. 680–695.
148. Ning, G.; Pei, J.; Huang, H. Lighttrack: A generic framework for online top-down human pose tracking. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 1034–1035.
149. Iqbal, U.; Milan, A.; Gall, J. Posetrack: Joint multi-person pose estimation and tracking. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 2011–2020.
150. Insafutdinov, E.; Andriluka, M.; Pishchulin, L.; Tang, S.; Levinkov, E.; Andres, B.; Schiele, B. Arttrack: Articulated multi-person tracking in the wild. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 6457–6465.
151. Feng, R.; Gao, Y.; Ma, X.; Tse, T.H.E.; Chang, H.J. Mutual Information-Based Temporal Difference Learning for Human Pose Estimation in Video. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 18–22 June 2023; pp. 17131–17141.
152. Ruan, W.; Liu, W.; Bao, Q.; Chen, J.; Cheng, Y.; Mei, T. Poinet: Pose-guided ovonic insight network for multi-person pose tracking. In Proceedings of the ACM International Conference on Multimedia, Nice, France, 21–25 October 2019; pp. 284–292.
153. Xiao, B.; Wu, H.; Wei, Y. Simple baselines for human pose estimation and tracking. In Proceedings of the European Conference on Computer Vision, Munich, Germany, 8–14 September 2018; pp. 466–481.
154. Bertasius, G.; Feichtenhofer, C.; Tran, D.; Shi, J.; Torresani, L. Learning temporal pose estimation from sparsely-labeled videos. *Adv. Neural Inf. Process. Syst.* **2019**, *32*, 3027–3038.
155. Xiu, Y.; Li, J.; Wang, H.; Fang, Y.; Lu, C. Pose Flow: Efficient online pose tracking. *arXiv* **2018**, arXiv:1802.00977.
156. Guo, H.; Tang, T.; Luo, G.; Chen, R.; Lu, Y.; Wen, L. Multi-domain pose network for multi-person pose estimation and tracking. In Proceedings of the European Conference on Computer Vision, Munich, Germany, 8–14 September 2018; pp. 209–216.
157. Doering, A.; Iqbal, U.; Gall, J. Joint flow: Temporal flow fields for multi person tracking. *arXiv* **2018**, arXiv:1805.04596.
158. Raaj, Y.; Idrees, H.; Hidalgo, G.; Sheikh, Y. Efficient online multi-person 2d pose tracking with recurrent spatio-temporal affinity fields. In Proceedings of the IEEE Conference on European Conference on Computer Vision, Long Beach, CA, USA, 16–20 June 2019; pp. 4620–4628.
159. Yang, Y.; Ren, Z.; Li, H.; Zhou, C.; Wang, X.; Hua, G. Learning dynamics via graph neural networks for human pose estimation and tracking. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 8074–8084.

160. Stadler, D.; Beyerer, J. Modelling ambiguous assignments for multi-person tracking in crowds. In Proceedings of the IEEE Winter Conference on Applications of Computer Vision, Waikoloa, HI, USA, 3–8 January 2022; pp. 133–142.
161. Jin, S.; Liu, W.; Ouyang, W.; Qian, C. Multi-person articulated tracking with spatial and temporal embeddings. In Proceedings of the IEEE Conference on European Conference on Computer Vision, Long Beach, CA, USA, 16–20 June 2019; pp. 5664–5673.
162. Gu, K.; Yang, L.; Yao, A. Dive deeper into integral pose regression. In Proceedings of the International Conference on Learning Representations, Online, 25–29 April 2022.
163. Alafif, T.; Hadi, A.; Allahyani, M.; Alzahrani, B.; Alhothali, A.; Alotaibi, R.; Barnawi, A. Hybrid Classifiers for Spatio-Temporal Abnormal Behavior Detection, Tracking, and Recognition in Massive Hajj Crowds. *Electronics* **2023**, *12*, 1165. [CrossRef]
164. Khan, E.A.; Shambour, M.K.Y. An analytical study of mobile applications for Hajj and Umrah services. *Appl. Comput. Inform.* **2018**, *14*, 37–47. [CrossRef]
165. Alharthi, N.; Gutub, A. Data visualization to explore improving decision-making within Hajj services. *Sci. Model. Res.* **2017**, *2*, 9–18. [CrossRef]
166. Shambour, M.K.; Gutub, A. Progress of IoT research technologies and applications serving Hajj and Umrah. *Arab. J. Sci. Eng.* **2022**, *47*, 1253–1273. [CrossRef] [PubMed]