Journal of Electrical and Computer Engineering Innovations
(JECEI)
Journal homepage: http://www.jecei.sru.ac.ir

Research paper

Multi-Task Learning Using Uncertainty for Realtime Multi-Person Pose Estimation

Z. Ghasemi-Naraghi, A. Nickabadi* *, R. Safabakhsh

Computer Engineering Department, Amirkabir University of Technology, Tehran, Iran.

Article Info

Article History:
Received 24 June 2023
Reviewed 24 August 2023
Revised 12 September 2023
Accepted 08 October 2023

Keywords:
Realtime multi-person pose estimation
Multi-Task learning
Loss function
Task-dependent uncertainty

*Corresponding Author's Email Address: *nickabadi@aut.ac.ir*

Abstract

Background and Obejctives: Multi-task learning is a widespread mechanism to improve the learning of multiple objectives with a shared representation in one deep neural network. In multi-task learning, it is critical to determine how to combine the tasks loss functions. The straightforward way is to optimize the weighted linear sum of multiple objectives with equal weights. Despite some studies that have attempted to solve the realtime multi-person pose estimation problem from a 2D image, major challenges still remain unresolved.

Methods: The prevailing solutions are two-stream, learning two tasks simultaneously. They intrinsically use a multi-task learning approach for predicting the confidence maps of body parts and the part affinity fields to associate the parts to each other. They optimize the average of the two tasks loss functions, while the two tasks have different levels of difficulty and uncertainty. In this work, we overcome this problem by applying a multi-task objective that captures task-based uncertainties without any additional parameters. Since the estimated poses can be more certain, the proposed method is called "CertainPose".

Results: Experiments are carried out on the COCO keypoints data sets. The results show that capturing the task-dependent uncertainty makes the training procedure faster and causes some improvements in human pose estimation.

Conclusion: The highlight advantage of our method is improving the realtime multi-person pose estimation without increasing computational complexity.

Introduction

Multi-person pose estimation is an important open problem in computer vision. Human pose estimation (HPE) is widely used in many applications such as human-computer interaction, action recognition, motion capture, virtual reality, video surveillance, healthcare, gaming, and sports. HPE aims to automatically locate the human parts or keypoints (e.g. ankles, knees, hips, elbows) on images and videos. In many real-world applications, the desired HPE model is expected to: 1) run in realtime, 2) estimate the poses of several people simultaneously, and 3) extract poses from 2D images. Each one of these requirements introduces many challenges. The focus of this research is on the realtime localization of body parts of individuals in 2D images.

The challenging issues of the single-person pose estimation include the variety of clothes, scenes, body shapes, positions, and the scales of the persons in the scenes. The multi-person pose estimation imposes more challenges as an unknown number of people can appear in images at any position and scale. Interactions between people may cause occluded joints or interrupted limbs. In addition, if the runtime complexity of the solution grows with the number of people in the image, it may not be useful in some real-world applications.

The initial studies of the single-person pose estimation [1] were based on the pictorial structure models [2].

Traditionally, the focus was on hand-crafted features such as the histogram of oriented gradients (HOG). But, these methods have not shown promising generalization performance in detecting the accurate location of the body parts. Deep learning, especially the convolutional neural networks (CNNs), made a significant improvement in this field [3]. Some HPE approaches used famous deep neural networks such as ResNet [4] and Faster R-CNN [5] to detect keypoints more accurately [6], [7]. Another example of a DNN-based HPE model is a convolutional pose machine (CPM) which consists of a sequence of convolutional neural networks that repeatedly produce more precise 2D confidence maps for the locations of human body parts at each stage [8]. However, there is still a long way towards the complete resolution of dominating some challenges of single-person pose estimation such as occlusion of body parts and abnormal body poses.

Multi-person 2D pose estimation is a widely investigated form of HPE. The solutions provided for this task can be divided into two main categories: top-down and bottom-up. Top-down methods [7], [9]-[11] first detect the people in the image and then utilize single-person pose estimation for each individual. The speed and accuracy of top-down methods depend on the human detection speed and accuracy. Moreover, these models fail to estimate human poses in crowded scenes and nearby individuals. On the other hand, the bottom-up methods [12]-[16] for first, detect human body parts without knowing the number and location of people in the image; and after that, they associate the parts of each individual to each other. The inference time of the bottom-up methods is usually satisfactory and independent of the number of people while preserving high-quality results. However, these approaches suffer from difficulty in grouping body parts when there is a large overlap between people. Another weakness of most of the bottom-up methods is the low resolution of the position of the individuals, which can be solved by increasing the width of the network or defining an additional unit to compute the more precise locations for each candidate point. Considering the goals of this investigation, we follow the bottom-up methods.

Recent years have witnessed a huge growth in realtime multi-person 2D pose estimation research. The winner of the COCO Keypoints 2016 challenge, CMU-Pose, is the first realtime multi-person pose estimator on 2D images [14]. The newer version of this model, OpenPose [17], is an open-source library [18] to localize full-body points on single images. Several researchers have tried to improve the OpenPose method [16], [19], [20]. The prevailing methods are bottom-up methods that learn confidence maps and part affinity fields (PAFs) simultaneously. Confidence maps and PAFs locate body parts and limbs, respectively. Limb refers to the virtual line between two keypoints. PAFs are utilized in associating the detected body parts to each individual at the inference time. A confidence map is a gray-level image in which the pixel value refers to the likelihood of the intended part on it.

The above models intrinsically use a multi-task learning (MTL) approach in which the learning of confidence maps of the body parts and the PAFs can be treated as two different tasks. Most of them consider the average mean square error of the two outputs as the multi-task loss function. Although the two tasks have different levels of difficulty, they are given the same weight in the loss function. It has been shown that in MTL, finding appropriate weights of different tasks plays an important role [21]. In this work, we explain an MTL strategy for realtime multi-person pose estimation from a 2D image. The proposed model, called CertainPose, captures task-dependent uncertainty in a two-stream network that jointly produces confidence maps and PAFs. For the purpose of capturing uncertainty without increasing the parameters and computational complexity, the model is trained with a new loss function which is derived in this manuscript.

In summary, the main contribution of this research is twofold. First, a novel multi-task loss function is introduced that captures task-dependent uncertainty in multi regression tasks models. Second, a two-task architecture is trained by the new loss function for multi-person pose estimation. Our experiments show that the proposed model reduces the training time and improves the accuracy of the pose estimation without increasing the process time and trainable parameters.

This paper is organized as follows: First, the related literature is briefly reviewed in Section “Related Work”. Next, the proposed method is described. We report the results of the experiments in Section “Experiments”. Finally, the paper is concluded in Section “Conclusion”.

Related Work

Human pose estimation has been a popular subject of research in recent years. There are some invaluable surveys on HPE methods [22]-[25]. HPE problems are divided into single and multi-person human pose estimation problems. In this section, we summarize some of the most important 2D HPE methods and their cons and pros. We also review some studies which solve 3D pose estimation by incorporating depth information. Given that our innovation is focused on reducing uncertainty in HPE models, our next step is to review the existing literature on the sources and effects of uncertainty in pose estimation.

A. Single-Person Pose Estimation

One of the oldest methods for estimating and tracking the human pose is the motion capture technique in which

the performer has to wear markers (e.g. LED, magnetic, and reflective markers) near each joint so the joints can be easily identified. This has been a useful method in filmmaking and animation and is still useful for laboratory activities [26]. However, it requires special hardware and software to obtain and process data. In most real-world applications, it is necessary to estimate human pose without using markers.

The earlier approaches for human pose estimation from image or video consider a graphical structure to model the interactions between body parts obtained from local observations. The extracted features can be classified into low-level, mid-level, and high-level features with regard to human visual perception. Silhouette, contour, and edges are some famous low-level features. These features are not useful in situations with complex backgrounds and scenes. SIFT, Freak, and shapelet are known as mid-level features. HOG has been the most popular mid-level feature in HPE [25]. Context features [6], mixtures of parts [1] and PAFs [14] are examples of high-level features used for HPE. In addition to the aforementioned features extracted from the input image, body structure models are also employed in HPE to provide prior knowledge about the relation of different parts of the body. Kinematic models [2], cardboard models [27], and volumetric models [28] are the usual body structures used in the literature. Kinematic models consider a line for the connections between pairs of body parts and it is possible to define some priors about joint angles. The cardboard models are composed of information about body part rectangular shapes. Volumetric Models realistically represent 3D body shapes and poses.

The pictorial structure model (PSM) was the first model to recognize the objects based on the positions of their components. In PSM, objects are modeled with a graph in which nodes refer to the body components and edges refer to the relations of these components. Most PSM-based human pose estimation methods consider ten body parts as rectangles and find the best parameters of these rectangles (e.g. center, scale, and rotation) by using the extracted features and the angles between the pairs of body parts [1].

The emergence of deep neural networks significantly affected HPE as many other artificial intelligence applications. In 2014, the replacement of handcrafted features with the features extracted by convolutional neural networks made notable improvements in HPE [3]. As the first example, [3] optimizes an energy function which contains two parts: 1) a unary potential which identifies the body parts likelihood in all image pixels and 2) a pairwise potential that models the relations of neighbor parts by considering the relative location and the size of the angle between the links to the parent and child nodes.

Neural networks and probabilistic graphical models are two basic and useful tools in HPE that have exclusive weak points. In [29], both paradigms are combined to improve the HPE accuracy. This repetitive algorithm computes the likelihood of each part in all pixels of the image as a confidence map by using the prior of the intended part and the conditional likelihood of it given other parts.

To enable tractable inference, PSM-based methods have been restricted to tree-structured body models. Pose machine [30] is an iterative pose prediction algorithm that incorporates richer spatial interconnection among multiple parts and shares information across parts of different scales. The input of the pose machine model is an image that goes through multiple stages. Each stage includes multi predictors which predict confidence maps of different parts in different scales. Practically, feeding the output of the predictors of one stage to the next stage gradually improves confidence maps predictions.

The Convolutional Pose Machine brought about a significant improvement in single-person pose estimation accuracy and speed [8]. Actually, this model implements the pose machine idea by convolutional neural networks in multi-stages. Increasing the number of stages with a constant kernel size enlarges the receptive fields. Moreover, the multi stages of the algorithm improve the accuracy and confidence of estimating difficult parts' localizations by utilizing easy parts locations. In addition, the vanishing gradient problem of the deep neural networks is solved here by using intermediate supervision enforcing at the end of each stage.

The second winner of the COCO 2016 keypoints challenge [31] represents a method [6] based on ResNet [4]. First, they predict the confidence maps for body parts by a ResNet. The low resolution of the ResNet's outputs enforces estimating offsets for each part. This method is very accurate in predicting the pose, but due to the use of a very deep ResNet, it has a high computational complexity.

B. Multi-Person Pose Estimation

Multi-person pose estimation is more difficult than the single-person case due to the interactions between people, which increases the inference complexity. Increasing the number of people makes realtime performance a challenge for multi-person pose estimation models. Multi-person human pose estimation models can be divided into two main categories: top-down and bottom-up approaches. The top-down methods first detect each person in the image and then perform a single-pose estimation for each person. But, in the bottom-up methods, the human body parts in the image are first detected and then associated with each other to form humans and human poses. Although top-

down methods provide good accuracies, their speed and accuracy greatly depend on the human detection model. The computational cost of these models increases with the number of detected people. Also, crowded scenes and high interactions between people are challenging situations for top-down methods. In contrast, the bottom-up approach represents realtime methods with satisfying accuracy. The two challenges of the bottom-up methods are how to associate the parts to bodies and how to cope with the low resolution of each person in images that can be processed by the related neural networks. The latter problem can be resolved by increasing the width of the network or computing the precise locations of the body parts by searching the surrounding area of the approximate part locations. In this subsection, some top-down and bottom-up methods are described, respectively.

As the first example of the top-down methods, [10] proposes a probabilistic approach for parts grouping and labeling which uses HOG features for part detection. It is developed as a part-based approach by optimizing an articulated pictorial structure and a pixel-based method for image labeling. The multi-person human pose estimation is treated as an optimization problem with a single energy function. The goal of the inference step of this model is three-fold: 1) to determine the number of people and their locations, 2) to localize their joints, and 3) to assign every pixel of the input image to the background or a body part of a person.

A local joint-to-person method is presented for estimating the truncated or occluded poses in [11]. First, the people bounding boxes are detected by Faster R-CNN [5][5]. Then, the joint candidates are localized for each person and his neighbors by the convolutional pose machine [8]. In the end, a fully connected graph from the set of the detected joint candidates is constructed and the joint-to-person association is carried out locally with integer linear programming.

The second winner of COCO 2016 keypoint challenge [31] first detects the people in an image by a ResNet [4], and then, as described in the first part of this section, carries out the HPE by another ResNet [6]. Although this model provides accurate pose estimations, its computation complexity is high.

As the last top-down method, we refer to one of the state-of-the-art methods, Mask R-CNN [7]. Inspired by Faster R-CNN [5], Mask R-CNN is proposed which belongs to the top-down category of object detection models. The features are extracted using a standard convolutional neural network such as ResNet [4]. Some regions are suggested by the region proposal network (RPN) and then the proposed regions and extracted features aid to localize people and predict the confidence maps of each body part. The RPN and the body parts localization units have common feature extractor layers.

Deepcut [12] is one of the bottom-up multi-person pose estimation methods that performs the body part detection and pose estimation simultaneously. It employs an integer linear programming formulation to partition and label the set of body parts detected by a CNN-based part detector. It detects some candidates for body parts and determines their type, e.g. head, foot, and hand. Deepcut considers a complete graph on detected parts. Then, it solves the optimization problem by integer linear programming, for purpose of removing the edges and segmenting the graph into some disjointed subgraphs. As a result, each subgraph refers to a person's pose. Deepcut theory is satisfactory, but in practice, its speed is very slow. It needs about 72 hours for processing an image.

A deeper, stronger, and faster Deepcut method is proposed in Deepercut model [13] which uses a deeper part detector based on ResNet [4] and novel stronger image-conditioned pairwise terms in the objective function. Due to its pairwise and incremental optimization, Deepercut is faster than Deepcut. It first finds heads and shoulders locations. Then, elbows and wrists are added to the first stage solution and re-optimization is performed. Finally, the rest of the body parts are added to the previous stage solution and re-optimized. Yet, Deepercut is still too slow for realtime problems. It takes about 8 minutes for processing one image.

The winner of the COCO 2016 keypoints challenge [31] was the CMU-Pose method [14] which was motivated by the Convolutional Pose Machine [8]. The CMU-Pose method includes a feature extractor unit and multiple stages of convolutional neural networks. In each stage, confidence maps of each part and PAFs for encoding part-to-part associations are predicted and refined. PAFs are unit vectors defined for each pixel that show the direction of the limbs connecting body parts. The width of each limb is determined from the length of the connected line between the two parts. During test time, they compute the line integral over the corresponding PAFs along the lines connecting the candidate part locations.

The winner of PoseTrack 2017 challenge [15] improves the CMU-Pose method. They consider a deeper network for feature extraction and empirically increase the number of network stages from 6 to 7. The main contribution of this paper is the definition of enhanced PAFs. In the CMU-Pose method, $n-1$ PAFs are defined for the n body parts. But in this work, additional PAFs are considered. For example, in addition to PAFs between hip and knee and also between knee and ankle, they define additional PAFs between hip and ankle.

Another variant of the CMU-Pose model appeared in OpenPose which increases both speed and accuracy [17]. They released an open-source library which was the first

available system for realtime multi-person 2D pose estimation, including body, foot, hand, and facial keypoints [18]. They found that PAFs refinement is more important than confidence map refinement. So, they remove the part refinement stages and increase the depth of the network. An important aspect of OpenPose is that it includes the location of the feet in its pose estimation. Some applications such as filmmaking require foot information. In addition, the foot keypoints (e.g. big toe and heel) localization helps to estimate the whole-pose more accurately. To address these issues, a small subset of foot instances is labeled. OpenPose first obtains body and foot keypoints locations [14] and then runs hand and face keypoints detectors [32] for each detected person.

The deep Whole-body method [20] applies an MTL approach to the OpenPose network to train the model with different scale properties. To improve face and hand keypoints localization, the network increases the input resolution. Unfortunately, this implicitly reduces the effective receptive fields and therefore reduces body/foot localization accuracy. To solve this issue, the number of convolutional layers in each PAF stage is increased to recover the effective receptive field that was previously reduced. As the result, while the large receptive field is preserved, a high resolution for precise face and hand keypoint detection is provided. The new approach yields higher accuracy than that of the original OpenPose, especially for face and hand keypoint detection in occluded, blurry, and low-resolution images. Additionally, its total training time and inference runtime are less than the previous OpenPose.

While most of the HPE models improve the accuracy of the previous models by increasing the number of the model's parameters, Liu et al. propose a method for increasing the accuracy without very additional complexity [19]. Their contributions are resolution irrelevant encoding (RIE) and difficulty balanced loss (DBL). RIE is an inner block offset supervision that aids to learn the more precise locations for keypoints. Furthermore, DBL is a loss function containing two parts: 1) a Gaussian loss weight for different pixels which guides the network focus on useful information, and 2) the progressive punishment that discerns between left and right joints.

In [16], a lightweight architecture is designed to perform pose estimation on edge devices. They follow the OpenPose model, because of its quality and robustness to the number of people inside the frames. The parameters and complexity of the designed network are just 15 percent of the baseline 2-stage OpenPose with almost the same quality.

Pifpaf [33] is a multi-person human pose estimation method that is suited for low resolution and crowded scenes. They use two units, a part intensity field (PIF) to localize body parts and a part association field (PAF) to associate body parts with each other to form full human poses. Part Association Field predicts two vectors to the two parts at every image pixel. They use Laplace loss for regressions which incorporates a notion of uncertainty.

C. Pose Estimation by Using Depth

3D pose estimation is useful in widespread applications, such as human motion analysis, human-computer interaction, and robotics. A large number of approaches have been developed for pose estimation of one or several people, cars, or even dishes. When the depth information is available, 3D pose estimation is simple. However, it is possible to estimate depth from a monocular image or images from multiple camera views. As an example, [34] uses OpenPose with multiple synchronized video cameras for developing a 3D markerless motion capture technique. Here, we review some works which utilize or estimate depth to address their problem.

TesseTrack [35] is a top-down approach to estimate and track 3D body joints from a video in an end-to-end network. Central to this work is a novel spatio-temporal formulation that estimates a spatio-temporal volume around each person by a 4D CNN. The evaluation demonstrates the excellent performance of TesseTrack.

Occlusion-Net [36] is a self-supervised network that predicts 2D and 3D locations of occluded keypoints for objects, especially for cars. At the core of this network are two losses: 1) a trifocal tensor loss that provides indirect self-supervision for occluded keypoint locations that are visible in other views of the object, and 2) the self-supervised reprojection loss which estimates the 3D shape and camera pose.

In [37], the integration of bottom-up and top-down approaches is proposed to exploit their strengths. Their bottom-up network incorporates normalized heatmaps based on human detection, and their top-down network estimates human joints from all persons rather than from one. Finally, 3D poses are estimated from the top-down and bottom-up estimated 3D poses by an integration network. Also, to enforces natural two-person interactions, a two-person pose discriminator is proposed.

VoxelPose [38] estimates 3D poses of several persons from multiple camera views. It directly operates in the 3D space by aggregating the features in all camera views in the 3D voxel space. Then the features are fed into a network to localize all people. Finally, another network estimates a detailed 3D pose for each proposal.

A multi-stream multi-task network [39] for RGB-D-based human detection and head pose estimation is introduced to overcome challenges due to variations of illumination, clothing, resolution, pose, occlusion, and background. They integrate RGB, depth, and optical flow

data, as inputs to represent the appearance, shape, and motion information of humans, which makes full use of all the information provided by RGB-D video sequences to achieve state-of-the-art performance on three challenging datasets.

Depth sensors are prevalent in today's robotics, but large amount of data for training CNN is not available. Regarding the importance of object recognition and pose estimation from RGB-D images and the expensive cost of creating and annotating datasets for learning, [40] tries to address the problem with transfer learning. They propose a transfer learning from deep convolutional neural networks (CNN) that are pre-trained and provide a rich, semantically meaningful feature set. They transform depth data into a representation that is easily interpretable by a CNN trained on color images. Actually, instead of handcrafting or learning features, they relied on a convolutional neural network (CNN) which was trained on a large image dataset. They show that supervised learning on the CNN features outperforms state-of-the-art methods.

6D pose estimation is a type of pose estimation that is an important task in robotics. It is the task of detecting the 3D location and 3D orientation of an object. Given the depth information makes it feasible to extract the full 6D pose of object instances present in the scene. [41] uses analysis-by-synthesis which is a method to compare the observation with the generated output. They learn a CNN that compares observed and synthesized images. In particular, for pose estimation, a forward synthesis model generates images from possible poses and then selects the best match with the observed image.

D. Uncertainty in Pose Estimation

Despite the great achievements of deep neural networks in many applications, they still suffer from some weaknesses. While DNNs show excellent ability in perception, they fail in proper thinking and relational reasoning. DNNs are data-driven and need a lot of diverse data to learn a task perfectly. Practically, the insufficiency of the training data in terms of the number or diversity of the data increases the uncertainty of the DNNs' predictions. There are also uncertainties related to the nature of data and tasks. Capturing different kinds of uncertainties in training DNNs, especially in multi-task problems, may improve the efficiency of the training process and increase the accuracy of the developed model. In summary, three types of uncertainties are captured by Bayesian deep learning [42], [43]:

(1) Epistemic uncertainty is caused by the lack of data in training the deep model. If the test data is different from the training set, epistemic uncertainty increases more. Epistemic uncertainty can be resolved by increasing diverse training data or defining a prior distribution over the weights of the neural network. Some effective and simple algorithms are employed for estimating epistemic uncertainty [44]. For example, abnormal human poses which are not found in training data increase epistemic uncertainty in an HPE task.

(2) Heteroscedastic aleatoric uncertainty depends on the input data and differs from one to another input. Unlike epistemic uncertainty, heteroscedastic aleatoric uncertainty does not increase for out-of-date samples and does not decrease with more data. It is predicted by considering a distribution over the model outputs. As an example, the pose estimation of a person whose clothes' color or skin tone is very similar to the background is more uncertain than that of a person with distinct cloth color or skin tone. Modeling heteroscedastic uncertainties can be simple with less complexity.

(3) Homoscedastic aleatoric uncertainty does not depend on the inputs and is constant for all input data. Actually, it is related to tasks and hence is called task-dependent uncertainty. The common noises inherent in the observations or sensors cause this type of uncertainty in deep networks. For example, for estimation of human pose on a single 2D image, the lack of depth data causes uncertainty which is present in all outputs. It can be captured as the output of a model and can be decreased by utilizing other information e.g. estimated depth of an image. In this research, we only consider task-dependent uncertainty.

Uncertainty-Based Multi-Person Pose Estimation: CertainPose

Multi-task learning (MTL) is an efficient way to improve the learning of multiple tasks with a shared representation in a network. MTL increases the prediction accuracy by involving joint learning of various tasks. Besides, combining multiple objectives in a model reduces the computational complexity, so it is useful in realtime systems. A naive approach for learning multiple tasks is to minimize a weighted linear sum of multiple objectives with equal or fixed predefined weights. while it is important to determine the optimal weights [21], manual tuning of weights is difficult and inefficient.

Some studies are carried out to find an appropriate approach to combine the tasks' loss functions. As an example, the Cross-stitch network proposes a new unit that learns optimal coefficients for multiple objectives [45][45] In another attempt, Gradient Normalization (GradNorm), an adaptive multi-task loss balancing technique, normalizes across tasks instead of batch data in batch normalization [46]. The Human can learn from knowledge, but deep networks are data-driven and, unlike the probabilistic graphical models, cannot model the probability and the uncertainty well. Uncertainty is increased by the lack of training data and their diversity. Even if there is enough and diverse data, some uncertainties still remain due to the nature of the data

and the task.

Considering the different degrees of difficulty and uncertainty in various tasks, [47] learns the task-based uncertainties as additional parameters in the network and uses them to combine multi-task loss functions in an MTL loss function.

In recent years, the prevailing approach for multi-person pose estimation [15]-[17], [19], [20] has been the extraction of a set of shared features for learning two disjoint representations, simultaneously: confidence maps of the joints and part affinity fields (PAFs). Confidence maps show the likelihood of the presence of each keypoint at each pixel of the input image. On the other hand, PAFs represent limbs, the connections between keypoints, by a set of unit vectors. Actually, these models leverage MTL to learn two tasks with two distinct loss functions using a shared representation in a deep neural network.

In most networks which predict confidence maps and PAFs simultaneously, the simplest way of MTL is applied. The loss function of each task is the mean square error of the predicted and the actual outputs and the total loss function of the network is the average of the two objectives. However, the uncertainties of the two tasks are not necessarily equal, and using different weights for the two tasks boosts the learning efficiency. In this research, we intend to learn the two tasks more efficiently with the same computational complexity as the base networks. The main novelty of this work is to capture task-dependent uncertainties in an MTL method without any additional parameters. So, we call the proposed method "CertainPose", as it captures the uncertainty and estimates more certain poses. This section continues by describing the overall architecture of our model. Then, we explain three types of uncertainties and an MTL approach that model task-dependent uncertainties. Finally, a new loss function is introduced for training confidence maps and PAFs more fairly, which captures task-dependent uncertainty without any additional parameters.

A. Network Architecture

The role of the CertainPose model is to predict confidence maps and PAFs for the input image, as shown Fig. 1. There are three main components: 1) Feature extractor, 2) Confidence maps predictor, and 3) PAFs predictor. The input of the model is an image that is first fed into a feature extractor, and then the extracted features are used by the Confidence maps predictor and PAFs predictor. The second component predicts the confidence map of each body joint which represents the confidence score of that joint at the location at each pixel. The third component predicts the PAFs for each body limb, consisting of a directional vector at each pixel of the input image. While the input and architecture of these two networks are similar, their goals are different. They are trained in parallel, but with different loss functions and ground truth maps and fields, which result are yielded from the ground truth coordinates of multi-person poses. Therefore, the parameters of the two networks are trained with different loss values. To calculate an overall loss value for training the CertainPose network, the loss functions of the two parallel networks can be aggregated in different ways. We propose a new loss function that is described later.

The detailed architecture of the CertainPose model is outlined in Fig. 2. First, the input image (I) is fed into the feature extraction unit. The extracted features are then passed to two branches of the model, each of which consisting of a series of convolutional and pooling layers. In the first branch, a feedforward network predicts a set of 2D confidence maps (S) for the body parts' locations. There are J confidence maps for different parts (keypoints) of the body in $S = \{S_1, S_2, \ldots, S_J\}$. In the second branch, a set of 2D vector fields (L) of PAFs are predicted which encode the unit vectors in the direction of limbs, resembling the connections between adjacent body parts. The PAFs set consists of C PAF related to the C limbs $L = \{L_1, L_2, \ldots, L_C\}$. The two-branch network is repeated over t successive stages to refine the predictions. At each stage, the confidence maps and PAFs of the previous step along with the extracted features are taken as the input and the refined confidence maps and PAFs are generated as the outputs.

The internal structure of the feature extractor and the units of the two branches of the network are demonstrated in Fig. 2 Similar to CMU-Pose [17], CertainPose uses the first ten layers of the VGG-19 network as the feature extractor, and adds two more 3x3 convolutional layers to these layers.

The two branches of the network consist of 6 stages. In the first stage, shared features are fed into two disjoint CNN networks with the same layers: two 3x3 convolutional layers followed by two 1x1 convolutional layers regressing the tasks' outputs, i.e. the confidence maps and PAFs.

The inputs of the next successive stages consist of the concatenation of the shared features and the outputs of the previous stage (confidence maps and PAFs).

In stages 2 to 6, there are two CNN networks in each stage which are similar to each other with four 7x7 convolutional layers and two 1x1 convolutional layers. The number of parameters in each layer of CertainPose is shown in Fig. 3.

Moreover, ReLu is used as the activation function in all neurons.

The deep networks suffer from the vanishing gradient problem.

The intermediate supervision at each stage addresses this problem by replenishing the gradient periodically.

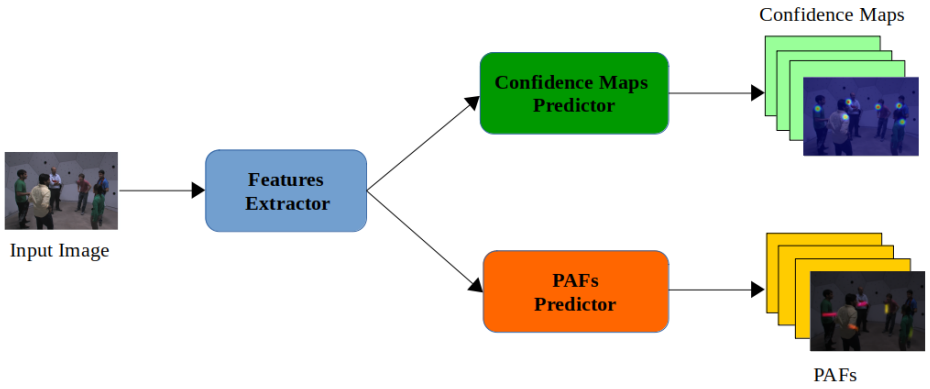


Fig. 1: The block diagram of CertainPose model.

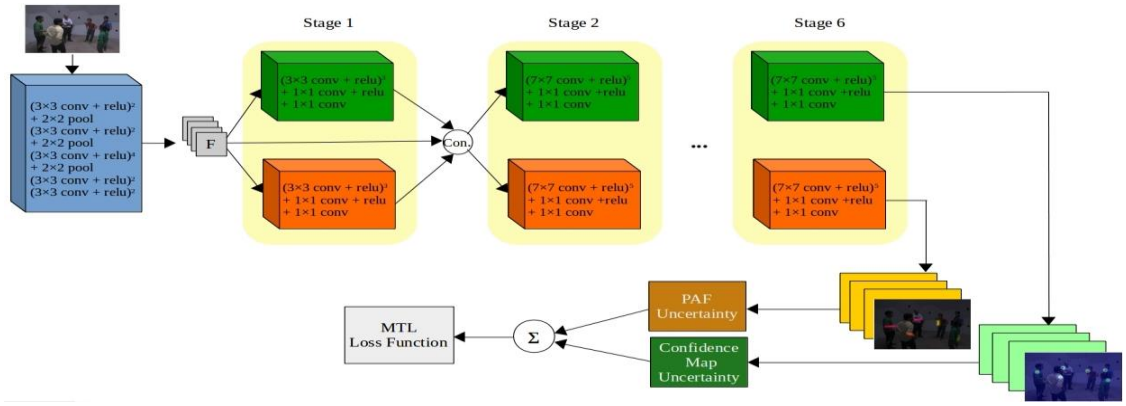


Fig. 2: The detailed architecture of CertainPose.

Number Of Parameters				
Feature Extractor layers	Confidence maps predictor		PAFs predictor	
1792	Stage 1	Stage 2 to 6	Stage 1	Stage 2 to 6
36928	147584	1160448	147584	1160448
73856	147584	802944	147584	802944
147584	147584	802944	147584	802944
295168	66048	802944	66048	802944
590080	19494	802944	9747	802944
590080		16512		16512
590080		4902		2451
1180160				
2359808				
1179904				
295040				
Total Parameters			**52,311,446**	
Trainable Parameters			**44,970,966**	
Non-Trainable Parameters			**7,340,480**	

Fig. 3: The number of parameters of CertainPose layers.

To guide the network to estimate the confidence maps and PAFs more accurately, the network is trained with a multi-task loss function. The loss function of each task is the mean of squared differences between the estimated and the actual outputs.

As the total loss function, CMU-Pose and the other approaches following it consider the average of the two tasks' loss functions. But, we attempt to have a fairer loss function for learning the two objectives by considering task-dependent uncertainties for the two tasks. Because of the more certain estimated poses, the new model is called 'CertainPose'. The new loss function is derived in subsection "Loss Function".

At the inference step, CertainPose predicts PAFs and Confidence maps for the input image. Similar to [14], non-maximum suppression is carried out to discretize the confidence maps and obtain some candidates for each part. A graph is then formed using candidate parts as vertices and candidate limbs as edges. To perform multi-person pose estimation, we should parse the graph and select the optimal set of limbs by measuring the association scores of the edges and removing the non-optimal edges. The score of a candidate limb is calculated by the line integral over the corresponding PAF along the candidate limb, virtual line segment connecting the candidate parts.

To speed up the parsing procedure, we use a greedy method. A greedy algorithm is a problem-solving approach that involves selecting the most advantageous

option at each step. While this strategy may not yield the best solution in all cases, it can produce locally optimal solutions that approximate the global optimal solution. Although greedy algorithms are not guaranteed to find the best solution, they are known for their speed and simplicity, making them a popular choice in real-time applications. We first consider a spanning tree skeleton for the human body instead of a complete graph, e.g. we ignore the virtual connected line between the head and the elbow. Then, we solve a bipartite matching problem to detect each limb. Bipartite matching is finding a set of edges between two vertices of two disjoint sets of vertices in the way that no two edges share an endpoint. For example, if we have three head and three shoulder keypoints, we should find the best three edges between the heads and shoulders without any shared point. Bipartite matching of disjoint parts' pairs obtains the limb connection candidates for each limb independently. Therefore, we can estimate the full-body poses of multiple people by assembling the candidate limbs.

B. Uncertainty

As described before, we only consider task-dependent uncertainty. Due to the importance of capturing task-based uncertainties and appropriately weighting the losses in multi-task learning, the uncertainty-based weighting method seems to be better than equal weighting of the losses [47]. The weights can be learned as a part of the convolutional neural network and loss functions. If the probabilistic likelihood of a regression task is considered as a Gaussian distribution, the variance parameter represents the noise and uncertainty of the task. In the following subsection, we describe this approach.

The problem (1) is finding the best weights w for the multi-task network f using the training data set $\{(I^{(i)}, S^{(i)}, L^{i}): i = 1,2, \dots, N\}$ where $I^{(i)}$, $S^{(i)}$, and $L^{(i)}$ refer to i-th sample input image, output set of confidence maps and PAFs, respectively.

$$\arg \max_{w} \mathcal{J}(w) \tag{1}$$

where

$$\mathcal{J}(w) = \prod_{i=1}^{N} p\left(S^{(i)}, L^{(i)} | I^{(i)}, w\right)$$
$$S = \mathcal{N}(f^{w_S}(I), \sigma_S^2); \; w_S \subset w$$
$$L = \mathcal{N}(f^{w_L}(I), \sigma_L^2); \; w_L \subset w \tag{2}$$

where w_S and w_L are weights related to the confidence maps and PAFs regression tasks, respectively. We assume that the two tasks are independent, so $\mathcal{J}(w)$ can be written as (3).

$$\mathcal{J}(w) = \prod_{i=1}^{N} p\left(S^{(i)} | I^{(i)}, w_S\right) p\left(L^{(i)} | I^{(i)}, w_L\right)$$
$$= \prod_{i=1}^{N} \frac{1}{\sqrt{2\pi}\sigma_S} exp\left(\frac{-\| S^{(i)} - f^{w_S}(I^{(i)}) \|^2}{2\sigma_S^2}\right)$$
$$\frac{1}{\sqrt{2\pi}\sigma_L} exp\left(\frac{-\| L^{(i)} - f^{w_L}(I^{(i)}) \|^2}{2\sigma_L^2}\right) \tag{1}$$

We solve the problem by minimizing the loss function, $\mathcal{L}$, instead of maximizing $\mathcal{J}$ (4).

$$\max_{w} \mathcal{J}(w) \equiv \min_{w} \mathcal{L}(w)$$
$$\mathcal{L}(w) = -\log\left(\mathcal{J}(w)\right)$$
$$= \sum_{i=1}^{N} \log\left(\sqrt{2\pi}\sigma_S\right) + \frac{\| S^{(i)} - f^{w_S}(I^{(i)}) \|^2}{2\sigma_S^2}$$
$$+\log\left(\sqrt{2\pi}\sigma_L\right) + \frac{\| L^{(i)} - f^{w_L}(I^{(i)}) \|^2}{2\sigma_L^2}$$
$$\mathcal{L}(w) = N\log(\sigma_S) + N\log(\sigma_L)$$
$$+\frac{1}{2\sigma_S^2}\sum_{i=1}^{N} \| S^{(i)} - f^{w_S}(I^{(i)}) \|^2$$
$$+\frac{1}{2\sigma_L^2}\sum_{i=1}^{N} \| L^{(i)} - f^{w_L}(I^{(i)}) \|^2 \tag{3}$$

$$\mathcal{L}(w) = \log(\sigma_S) + \log(\sigma_L)$$
$$+\frac{1}{2\sigma_S^2}\mathcal{L}_S(w_S) + \frac{1}{2\sigma_L^2}\mathcal{L}_L(w_L)$$
$$\mathcal{L}_S(w_S) = \frac{1}{N}\sum_{i=1}^{N} \| S^{(i)} - f^{w_S}(I^{(i)}) \|^2$$
$$\mathcal{L}_L(w_L) = \frac{1}{N}\sum_{i=1}^{N} \| L^{(i)} - f^{w_L}(I^{(i)}) \|^2 \tag{4}$$

Equation (5) shows the final solution of the proposed loss function, where σ_S and σ_L should be learned. As shown in this section, the task-dependent uncertainty can be captured by multi-task learning, while learning the additional parameters increases computational complexity.

C. Loss Function

We propose a new method for capturing the task-based uncertainties. In this method, the network architecture does not change and no further computational complexity is required. The new loss function (6) is derived as (7).

$$\arg \min_{w} \mathcal{L}(w) \tag{5}$$

where

$$\mathcal{L}(w) = \sum_{i=1}^{N} \log(\sigma_S) + \log(\sigma_L) + \frac{1}{2\sigma_S^2}\mathcal{L}_S(w_S) + \frac{1}{2\sigma_L^2}\mathcal{L}_L(w_L)$$
$$\mathcal{L}_S(w_S) = \frac{1}{N}\sum_{i=1}^{N} \| S^{(i)} - f^{w_S}(I^{(i)}) \|^2$$
$$\mathcal{L}_L(w_L) = \frac{1}{N}\sum_{i=1}^{N} \| L^{(i)} - f^{w_L}(I^{(i)}) \|^2 \quad (6)$$

In regression tasks, the likelihood is considered as a Gaussian whose mean is the output of the model (8). This means when the network's parameters are learned through the training process, the network's output for each task approaches the mean of the corresponding response variable given the input, i.e. $\overline{S}$ and $\overline{L}$ for our two tasks. So, σ_S^2 and σ_L^2 can be estimated with sample variances $\overline{Var_S^2}$ and $\overline{Var_L^2}$, respectively.

$$p(S_i \mid \overline{S}) = \mathcal{N}(\overline{S}, \sigma_S^2)$$
$$p(L_i \mid \overline{L}) = \mathcal{N}(\overline{L}, \sigma_L^2) \quad (7)$$

$$\mathcal{L}_S(w_S) = \frac{1}{N}\sum_{i=1}^{N} \| S^{(i)} - \overline{S} \|^2 = \overline{Var_S^2}$$
$$\mathcal{L}_L(w_L) = \frac{1}{N}\sum_{i=1}^{N} \| L^{(i)} - \overline{L} \|^2 = \overline{Var_L^2} \quad (8)$$

As a result, a simple equation is obtained for the loss function (10). Therefore, we consider the average log of the tasks' loss functions instead of the mean of the loss functions themselves.

$$\mathcal{L}(w) = \frac{1}{2}\Big(\log\big(\mathcal{L}_S(w_S)\big) + \log\big(\mathcal{L}_L(w_L)\big)\Big) \quad (9)$$

The new loss function aims to improve the validity of the body part predictions by capturing task-based uncertainty without changing the complexity of the model.

Experiments

In this section, we first introduce the datasets and evaluation metrics and then report the experimental results and analyze them.

A. Datasets and Metrics

We conduct the experiments on the COCO keypoints 2014 and COCO keypoints 2017 datasets [48]. These datasets are the largest collection of multi-instance person keypoint annotations which has been widely used in many studies. COCO datasets consist of many challenging situations for multi-person pose estimation problems. 17 keypoints including 12 human body parts and 5 facial keypoints are localized in the COCO keypoints dataset. COCO keypoints 2014 consists of 83k training data and 41k test data and COCO keypoints 2017 consists of 118k training and 41k test data. The COCO training set consists of over 100K person instances labeled with over 1 million keypoints. We report the results on both versions of COCO keypoints.

The performance of the proposed method is evaluated based on the object keypoint similarity (OKS) which is defined in COCO evaluation [49]. The role of OKS is the same as the IoU in object detection. OKS measures the degree of match between real and predicted poses. It ranges from 0 to 1 which refers to poor to perfect match. The mean average recall (AR) and the mean average precision (AP) over 10 OKS thresholds are used as the main competition metrics. Moreover, we assess the methods by AP and AR over thresholds 0.5 and 0.75, which are indicated by AP^{50} and AR^{50}, and AP^{75} and AR^{75}, respectively. Besides, results per each body part are presented to have a better analysis.

The results are reported on two models: 1) CMU-Pose: a model whose architecture is similar to our proposed model, but uses the CMU-Pose loss function [14] which is the average of the two tasks' loss functions and does not capture uncertainty, 2) CertainPose: the proposed method which captures task-based uncertainty as a new loss function.

B. Results and Discussion

Both models, CertaionPose and CMU-Pose, are trained on COCO keypoints 2014 training data. The MultiSGD optimizer with a learning rate of 2e-5 is used and the size of each batch is 10 images. CMU-Pose and CertainPose are trained for 100 and 18 epochs respectively. Practically, CertainPose can be trained faster than CMU-Pose.

Table 1: Comparison between CertainPose and CMU-Pose by mean of AP and AR metrics over all body parts on COCO validation sets 2014 and 2017

DB	Methods	AP	$\boldsymbol{AP^{50}}$	$\boldsymbol{AP^{75}}$	AR	$\boldsymbol{AR^{50}}$	$\boldsymbol{AR^{75}}$
Val2014	CMU-Pose	**0.59**	0.792	0.637	0.623	0.806	0.664
	CertainPose	0.589	**0.802**	**0.643**	**0.626**	**0.816**	**0.671**
Val2017	CMU-Pose	**0.578**	0.78	**0.625**	0.613	0.795	0.654
	CertainPose	0.575	**0.79**	0.624	**0.614**	**0.804**	**0.66**

The results of the test procedure for CMU-Pose and CertainPose on both datasets are shown in Table 1. The higher AP values refer to the more precise localization, and the higher ARs show more valid predictions. The results are measured by mean AP and mean AR over three values of OKS thresholds (0.5, 0.75, and 0.05:0.95) for all body parts. The higher threshold value considers the more perfect match between the estimated and real parts locations. The results show that CertainPose: 1) improves

AR measure definitely, 2) improves AP measure with lower OKS, and 3) has AP factor comparable to the base model when OKS is increased.

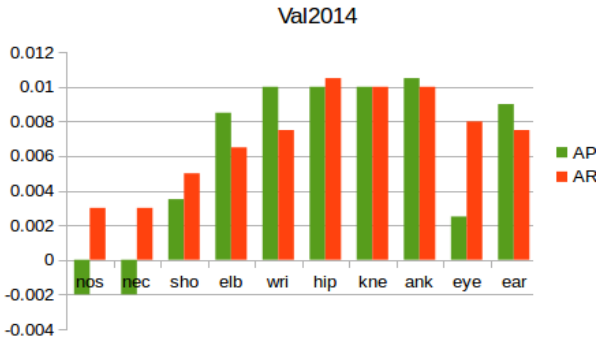


Fig. 3: The improvements of CertainPose for different keypoints on Val2014 dataset by $AP^{50}$and AR^{50}metrics for OKS=0.5.

Here, we explain the conclusions more clearly. First, CertainPose improves the AR measure by capturing task-based uncertainty through the loss function. This results in more valid and more certain outputs. In other words, the false positive rate is reduced in this method. Second, the AP measure improves because of training the two tasks fairer and predicting more accurate PAFs, which are impressive in keypoint association and body pose estimation. Third, while the CertainPose AP measure improves for lower OKS, it is not better than CMU-Pose for the higher OKS threshold, e.g. 0.95. It means that our model estimates more valid and more accurate body poses, but it fails to localize body parts more precisely since we apply a log operator over the distance between the predicted and the actual outputs in the loss function.

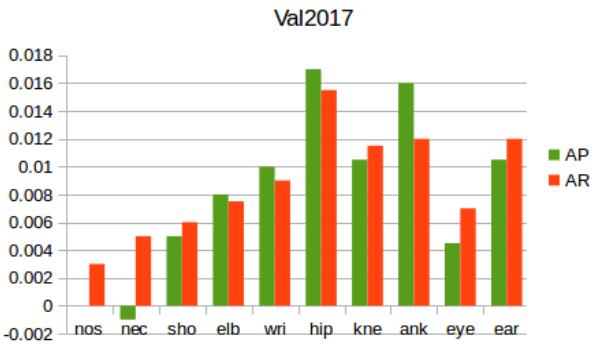


Fig. 4: The improvements of CertainPose for different keypoints on Val2017 dataset by $AP^{50}$and AR^{50}metrics for OKS=0.5.

CertainPose can associate the localized keypoints more accurately because PAFs are predicted more precisely. In the CMU-Pose loss function, equal weights are considered for PAFs and Confidence maps, but CertainPose considers task-dependent uncertainties to weigh the sub-loss functions. PAFs are more difficult than confidence maps to predict. Therefore, PAFs need higher weights and CertainPose predicts PAFs more precisely. Fig. 4 and Fig. 5 show the improvements of CertainPose for each keypoint in comparison with CMU-Pose on COCO val2014 and val2017 datasets. The keypoints like the elbows, hips, ankles, and knees which are connected with more clear limbs are localized more precisely.

We further show qualitative results for some images in Fig. 6. The (a) and (b) parts show the results of CertainPose and CMU-Pose, respectively. Some keypoints such as knees and elbows are predicted more accurately by the CertainPose method. Predicting the right elbow and left knee causes the more correct poses in the first two of the above images. Other shown samples demonstrate the power of CertainPose in PAF estimation. Higher accuracy in PAFs estimation is the reason of correct poses in the left hand and the left leg of the men, and the left hand of the baby in other three images, respectively.

We have analyzed the cases where our approach fails. Fig. 5. shows an overview of some failure cases and compares with the base method. The low resolution is the main cause of errors in the joint localization and PAFs estimation.

Realtime estimation is an important characteristic of HPE models in many real-world applications. CMU-Pose is the popular realtime multi-person pose estimation method. Its speed is independent of the number of people in the image. CertainPose improves the base model without adding any parameters. The main contribution of CertainPose is improving the CMU-Pose accuracy without decreasing the speed and increasing the complexity.

Table 2 shows the almost equal time of the two methods when perform single-scale process with the CUDA toolkit.

Table 2: Comparison between CertainPose and CMU-Pose by process time (ms)

Methods	CertainPose	CMU-Pose
CUDA (ms)	88.74	86.82

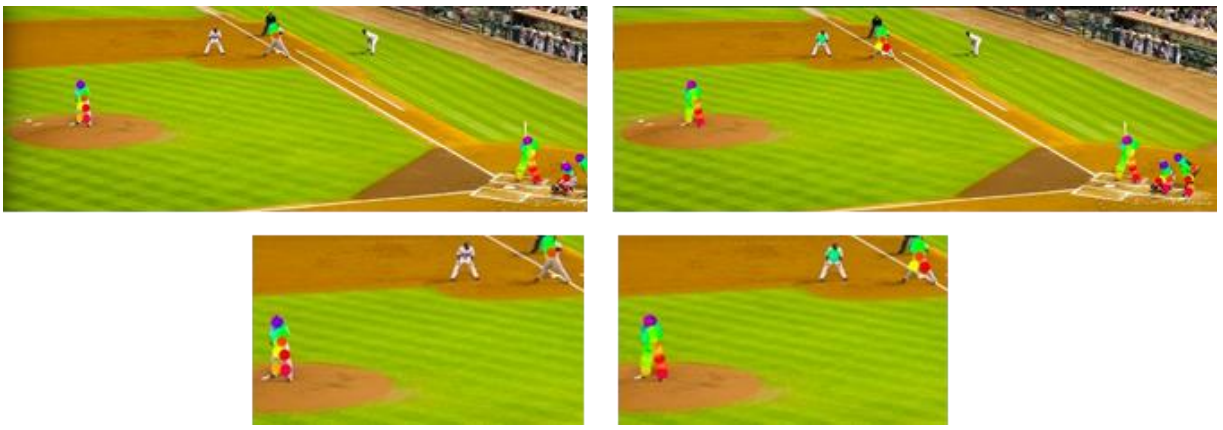
The goal of this research was to introduce the new loss function and investigate its performance in an applicable network, 2D HPE.

We show that we can increase the accuracy with the same number of parameters and inference speed. It is true that the improvement is not very significant, but one should note that the cost is not increased, either. In addition, the task-dependent uncertainties are captured and a few epochs are needed in the training step. The comparison of CertainPose and some other studies on COCO val split is shown in Table 3.

(a)

(b)

Fig. 6: The qualitative results of (a) CertainPose and (b) CMU-Pose.



(a)

(b)

(c)

Fig. 5: Visualization of some failure results of CertainPose on the COCO dataset and comparison between CertainPose (left) and CMU-Pose (right).

Osokin [16] introduces a lightweight OpenPose with fewer parameters and lower complexity compared to OpenPose. Cao et al. [14], the winner of the COCO 2016 keypoints challenge reports 58.4 AP for a model that is similar to CertainPose but, with two less layers. Newell et al. [50] propose a new approach for detections and group assignments. As reported in [51], their AP on COCO dataset is 56.9. Kocabas et al. [51] improve the base method by using a new grouping idea to associate body joints.

Table 3: Comparison of CertainPose and other works.

Methods	Osokin 161 [16]	Cao et al. [14]	Newell et al. [50]	Kocabas et al. [51]	CertainPose
AP	48.6	58.4	56.9	59.2	58.9

In summary, we compare the proposed idea and the baseline in Table 4. However, CMU-Pose is trained for 100 epochs, CertainPose needs 18 epochs. The runtime and number of parameters are almost the same.

The AP and AR comparisons show that CertainPose estimates more valid and accurate poses, and finds the less precise location for keypoints.

Conclusion

To obtain a more certain realtime multi-person pose estimation network, we propose a method to capture task-dependent uncertainties across the loss functions without increasing the number of parameters. As comparison Table 4 shows, the experiments prove that CertainPose: 1) needs fewer epochs for training, 2) preserves the realtime pose estimation property, 3) provides more valid and accurate estimations, and 4) locates keypoints less precisely.

In future work, we intend to examine different tasks and information to improve multi-person pose estimation.

The main weakness of our work is focusing on PAFs which causes less precise predicted heatmaps, particularly for keypoints with lower resolution (Fig. 5). We can use high resolution architecture instead of predictor units.

Also, the CertainPose idea can be the base method for incorporating other information to improve the pose estimation accuracy.

Table 4: Comparison between CertainPose and CMU-Pose.

Method	**Epochs**	**Runtime**	**Parameters**	AP^{50} **(V14)**	AP^{50} **(V17)**	AR^{50} **(V14)**	AR^{50} **(V17)**
CertainPose	18	88.74	44,970,966	0.802	0.79	0.816	0.804
CMU-Pose	100	86.82	44,970,966	0.792	0.78	0.806	0.795

Author Contributions

Z. Ghasemi-Naraghi implemented the proposed model and performed the experiments. Z. Ghasemi-Naraghi and A. Nickabadi interpreted the results. Z. Ghasemi-Naraghi, A. Nickabadi and R. Safabakhsh wrote the manuscript.

Conflict of Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Abbreviations

HPE Human Pose Estimation
HOG Histogram of Oriented Gradients
CNN Convolutional Neural Network
CPM Convolutional Pose Machine
PAFs Part Affinity Fields
MTL Multi-Task Learning
PSM Pictorial Structure Model
RPN Region Proposal Network
RIE Resolution Irrelevant Encoding
DBL Difficulty Balanced Loss
PIF Part Intensity Field
PAF Part Association Field
CNN Convolutional Neural Networks
MTL Multi-Task Learning
OKS Object Keypoint Similarity
AR Average Recall
AP Average Precision

References

[1] Y. Yang, D. Ramanan, "Articulated pose estimation with flexible mixtures-of-parts," in Proc. CVPR 2011: 1385–1392, 2011.

[2] P. F. Felzenszwalb, D. P. Huttenlocher, "Pictorial structures for object recognition," Int. J. Comput. Vision, 61(1): 55–79, 2005.

[3] X. Chen, A. L. Yuille, "Articulated pose estimation by a graphical model with image dependent pairwise relations," in Proc. NIPS, 2014.

[4] K. He, X. Zhang, S. Ren, J. Sun, “Deep residual learning for image recognition,” in Proc. the IEEE Conference on Computer Vision and Pattern Recognition: 770–778, 2016.

[5] S. Ren, K. He, R. Girshick, J. Sun, “Faster r-cnn: Towards real-time object detection with region proposal networks,” in Proc. NIPS, 2015.

[6] G. Papandreou, T. Zhu, N. Kanazawa, A. Toshev, J. Tompson, C. Bregler, K. Murphy, “Towards accurate multi-person pose estimation in the wild,” in Proc. the IEEE Conference on Computer Vision and Pattern Recognition: 4903–4911, 2017.

[7] K. He, G. Gkioxari, P. Dollár, R. Girshick, “Mask r-cnn,” in Proc. the IEEE International Conf. on Computer Vision: 2961–2969, 2017.

[8] S. E. Wei, V. Ramakrishna, T. Kanade, Y. Sheikh, ” Convolutional pose machines,” in Proc. the IEEE Conference on Computer Vision and Pattern Recognition: 4724–4732, 2016.

[9] H. S. Fang, S. Xie, Y. W. Tai, C. Lu, “Rmpe: Regional multi-person pose estimation,” in Proc. the IEEE International Conference on Computer Vision: 2334–2343, 2017.

[10] L. Ladicky, P. H. Torr, A. Zisserman, “Human pose estimation using a joint pixel-wise and part-wise formulation,” in Proc. IEEE Conf. on Computer Vision and Pattern Recognition: 3578– 585, 2013.

[11] U. Iqbal, J. Gall, “Multi-person pose estimation with local joint-to-person associations,” in Proc. European Conference on Computer Vision: 627–642, 2016.

[12] L. Pishchulin, E. Insafutdinov, S. Tang, B. Andres, M. Andriluka, P. V. Gehler, B. Schiele, “Deepcut: Joint subset partition and labeling formulti person pose estimation,” in Proc. the IEEE Conference on Computer Vision and Pattern Recognition: 4929–4937, 2016.

[13] E. Insafutdinov, L. Pishchulin, B. Andres, M. Andriluka, B. Schiele, “Deepercut: A deeper, stronger, and faster multi-person pose estimation model,” in Proc. European Conference on Computer Vision: 34–50, 2016.

[14] Z. Cao, T. Simon, S. E. Wei, Y. Sheikh, “Realtime multi-person 2dpose estimation using part affinity fields,” in Proc. the IEEE Conf. Computer Vision and Pattern Recognition: 7291–7299, 2017.

[15] X. Zhu, Y. Jiang, Z. Luo, “Multi-person pose estimation for posetrack with enhanced part affinity fields,” in Proc. ICCV PoseTrack Workshop, 7, 2017.

[16] D. Osokin, “Real-time 2d multi-person pose estimation on cpu: Lightweight openpose,” in Proc. the 8th International Conference on Pattern Recognition Applications and Methods: 744–748, 2019.

[17] Z. Cao, G. Hidalgo, T. Simon, S. E. Wei, Y. Sheikh, “Openpose: real-time multi-person 2d pose estimation using part affinity fields,” IEEE Trans. Pattern Anal. Mach. Intell., 43(1): 172–186, 2019.

[18] OpenPose library. https://github.com/CMU-Perceptual-Computing-Lab/openpose.

[19] H. Liu, D. Luo, S. Du, T. Ikenaga, “Resolution irrelevant encoding and difficulty balanced loss based network independent supervision for multi-person pose estimation,” in Proc. 13th International Conf.HumanSystem Interaction (HSI):112–117, 2020.

[20] G. H. Martınez, “OpenPose: Whole-Body Pose Estimation,” April, 2019.

[21] T. Gong, T. Lee, C. Stephenson, V. Renduchintala, S. Padhy, A. Ndirango, G. Keskin, O. H. Elibol, “A comparison of loss weighting strategies for multi task learning in deep neural networks,” IEEE Access, 7: 141627–141632, 2019.

[22] Q. Dang, J. Yin, B. Wang, W. Zheng, “Deep learning based 2d human pose estimation: A survey,” Tsinghua Sci. Technol., 24(6): 663–676, 2019.

[23] C. Zheng, W. Wu, T. Yang, S. Zhu, C. Chen, R. Liu, J. Shen, N. Kehtarnavaz, M. Shah, “Deep learning-based human pose estimation: A survey,” arXiv preprint arXiv:2012.13392, 2020.

[24] T. L. Munea, Y. Z. Jembre, H. T. Weldegebriel, L. Chen, C. Huang, C. Yang, “The progress of human pose estimation: A survey and taxonomy of models applied in 2d human pose estimation,” IEEE Access, 8: 133330–133348, 2020.

[25] W. Gong, X. Zhang, J. Gonzàlez, A. Sobral, T. Bouwmans, C. Tu, E. H. Zahzah, “Human pose estimation from monocular images: A comprehensive survey,” Sensors, 16(12): 1966, 2016.

[26] G. Rogez, C. Schmid, “Mocap-guided data augmentation for 3d pose estimation in the wild,” in Proc. Advances in Neural Information Processing Systems (NIPS): 3108–3116, 2016.

[27] H. Jiang, “Finding human poses in videos using concurrent matching and segmentation,” in Proc. Asian Conference on Computer Vision: 228–243, 2010.

[28] H. Sidenbladh, F. De la Torre, M. J. Black, “A framework for modeling the appearance of 3d articulated figures,” in Proc. Fourth IEEE International Conference on Automatic Face and Gesture Recognition (Cat. No. PR00580): 368–375, 2000.

[29] J. J. Tompson, A. Jain, Y. LeCun, C. Bregler, “Joint training of a convolutional network and a graphical model for human pose estimation,” in Proc. Advances in Neural Information Processing Systems, 27, 2014.

[30] V. Ramakrishna, D. Munoz, M. Hebert, J. A. Bagnell, Y. Sheikh, “Pose machines: Articulated pose estimation via inference machines,” in Proc. European Conf. Computer Vision: 33–47, 2014.

[31] MSCOCO Dataset, https://cocodataset.org/#home.

[32] T. Simon, H. Joo, I. Matthews, Y. Sheikh, “Hand keypoint detection in single images using multiview bootstrapping,” in Proc. the IEEE Conf. Computer Vision and Pattern Recognition: 1145–1153, 2017.

[33] S. Kreiss, L. Bertoni, A. Alahi, “Pifpaf: Composite fields for human pose estimation,” in Proc. the IEEE/CVF Conference on Computer Vision and Pattern Recognition: 11977–11986, 2019.

[34] N. Nakano, T. Sakura, K. Ueda, L. Omura, A. Kimura, Y. Iino, S. Fukashiro, S. Yoshioka, “Evaluation of 3d markerless motion capture accuracy using openpose with multiple video cameras,” Fron. sports Active Living, 2(50), 2020.

[35] N. D. Reddy, L. Guigues, L. Pishchulin, J. Eledath, S. G. Narasimhan, “Tessetrack: End-to-end learnable multi-person articulated 3d pose tracking,” in Proc. the IEEE/CVF Conference on Computer Vision and Pattern Recognition: 15190–15200, 2021.

[36] N. D. Reddy, M. Vo, S. G. Narasimhan, “Occlusion-net: 2d/3d occluded keypoint localization using graph networks,” in Proc. the IEEE/CVF Conference on Computer Vision and Pattern Recognition: 7326–7335, 2019.

[37] Y. Cheng, B. Wang, B. Yang, R. T. Tan, “Monocular 3d multi-person pose estimation by integrating top-down and bottom-up networks,” in Proc. the IEEE/CVF Conference on Computer Vision and Pattern Recognition: 7649–7659, 2021.

[38] H. Tu, C. Wang, W. Zeng, “Voxelpose: Towards multi-camera 3d human pose estimation in wild environment,” in Proc. 16th European Conference on Computer Vision–ECCV 2020, Part I 16: 197–212, 2020.

[39] G. Zhang, J. Liu, H. Li, Y. Q. Chen, L. S. Davis, “Joint human detection and head pose estimation via multistream networks for rgb-d videos,” IEEE Signal Process. Lett., 24(11): 1666–1670, 2017.

[40] M. Schwarz, H. Schulz, S. Behnke, “Rgb-d object recognition and pose estimation based on pre-trained convolutional neural network features,” in Proc. 2015 IEEE International Conference on Robotics and Automation (ICRA): 1329–1335, 2015.

[41] A. Krull, E. Brachmann, F. Michel, M. Y. Yang, S. Gumhold, C. Rother, “Learning analysis-by-synthesis for 6d pose estimation in rgb-d images,” in Proc. the IEEE International Conference on Computer Vision: 954–962, 2015.

[42] Y. Gal, “Uncertainty in deep learning,” University of Cambridge 1(3), 2016.

[43] A. Kendall, Y. Gal, “What uncertainties do we need in bayesian deep learning for computer vision?” in Proc. Advances in Neural Information Processing Systems: 5574–5584, 2017.

[44] F. K. Gustafsson, M. Danelljan, T. B. Schon, "Evaluating scalable bayesian deep learning methods for robust computer vision," in Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops: 318–319, 2020.

[45] I. Misra, A. Shrivastava, A. Gupta, M. Hebert, "Cross-stitch networks for multi-task learning," in Proc. the IEEE Conference on Computer Vision and Pattern Recognition: 3994–4003, 2016.

[46] Z. Chen, V. Badrinarayanan, C. Y. Lee, A. Rabinovich, "Gradnorm: Gradient normalization for adaptive loss balancing in deep multitask networks," in Proc. International Conference on Machine Learning: 794–803, 2018.

[47] A. Kendall, Y. Gal, R. Cipolla, "Multi-task learning using uncertainty to weigh losses for scene geometry and semantics," in Proc. the IEEE Conference on Computer Vision and Pattern Recognition: 7482–7491, 2018.

[48] T. Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, C.L. Zitnick, "Microsoft coco: Common objects in context. In Proc. European Conf. Computer Vision: 740–755, 2014.

[49] M. Ruggero Ronchi, P. Perona, "Benchmarking and error diagnosis in multi-instance pose estimation," in Proc. the IEEE International Conference on Computer Vision: 369–378, 2017.

[50] A. Newell, Z. Huang, J. Deng, "Associative embedding: End-to-end learning for joint detection and grouping," in Proc. Advances in Neural Information Processing Systems: 2278–2288, 2017.

[51] M. Kocabas, S. Karagoz, E. Akbas, "Multiposenet: Fast multi-person pose estimation using pose residual network," in Proc. the European Conference on Computer Vision (ECCV): 417–433, 2018.

Biographies

Zeinab Ghasemi-Naraghi is a Ph.D. student in Artificial Intelligence at the Computer Engineering Department, Amirkabir University of Technology, Tehran, Iran. She earned a M.S. degree in Artificial Intelligence from Sharif University of Technology and received the B.S. degree in Computer Engineering from Shahid Beheshti University in 2013 and 2009, respectively. Her research interests are mainly in computer vision, deep learning and probabilistic graphical models.

- Email: z_naraghi@aut.ac.ir
- ORCID: 0009-0008-9545-8706
- Web of Science Researcher ID: NA
- Scopus Author ID: NA
- Homepage: NA

Ahmad Nickabadi received the B.S. degree in Computer Engineering and the M.S. and Ph.D. degrees in Artificial Intelligence from the Amirkabir University of Technology (AUT), Tehran, Iran, in 2004, 2006, and 2011, respectively. Since 2012, he has been an Assistant Professor with the Computer Engineering Department, AUT. His research interests include the analysis of image and video content using deep learning and probabilistic graphical models with a special focus on activity recognition, face recognition, and face synthesis.

- Email: nickabadi@aut.ac.ir
- ORCID: 0000-0003-3709-1041
- Web of Science Researcher ID: NA
- Scopus Author ID: NA
- Homepage: https://aut.ac.ir/cv/2387/Ahmad%20Nikabadi

Reza Safabakhsh received the B.S. degree in Electrical Engineering from Sharif University of Technology, Tehran, Iran, in 1976 and the M.S. and Ph.D. degrees in Electrical and Computer Engineering from the University of Tennessee, Knoxville, in 1980 and 1986, respectively. He worked at the Center of Excellence in Information Systems, Nashville, TN, USA, from 1986 to 1988. Since 1988, he has been with the Computer Engineering Department, Amirkabir University of Technology, Tehran, Iran, where he is currently a professor and the director of the Computer Vision Laboratory. His current research interests include neural networks, computer vision, and deep learning. Dr. Safabakhsh is a member of the IEEE and several honor societies, including Phi Kappa Phi and Eta Kapa Nu. He was the founder and a member of the Board of Executives of the Computer Society of Iran, and was the President of this society for the first 4 years.

- Email: safa@aut.ac.ir
- ORCID: 0000-0002-4937-8026
- Web of Science Researcher ID:
- Scopus Author ID
- Homepage: https://aut.ac.ir/cv/2455/REZA%20SAFABAKHSH

How to cite this paper:
Z. Ghasemi-Naraghi, A. Nickabadi, R. Safabakhsh, "Multi-Task learning using uncertainty for realtime multi-person pose estimation," J. Electr. Comput. Eng. Innovations, 12(1): 147-162, 2024.

DOI: 10.22061/jecei.2023.9848.657

URL: https://jecei.sru.ac.ir/article_1985.html