ELSEVIER

Computer Vision and Image Understanding
journal homepage: www.elsevier.com

High-speed Multi-person Pose Estimation with Deep Feature Transfer

Ying Huang[a], Hubert P. H. Shum[a,**], Edmond S. L. Ho[a], Nauman Aslam[a]

[a] *Department of Computer and Information Sciences, Northumbria University, Newcastle Upon Tyne, NE1 8ST, UK*

ABSTRACT

Recent advancements in deep learning have significantly improved the accuracy of multi-person pose estimation from RGB images. However, these deep learning methods typically rely on a large number of deep refinement modules to refine the features of body joints and limbs, which hugely reduce the run-time speed and therefore limit the application domain. In this paper, we propose a feature transfer framework to capture the concurrent correlations between body joint and limb features. The concurrent correlations of these features form a complementary structural relationship, which mutually strengthens the network's inferences and reduces the needs of refinement modules. The transfer sub-network is implemented with multiple convolutional layers, and is merged with the body part detection network to form an end-to-end system. The transfer relationship is automatically learned from ground-truth data instead of being manually encoded, resulting in a more general and efficient design. The proposed framework is validated on the multiple popular multi-person pose estimation benchmarks - MPII, COCO 2018 and PoseTrack 2017 and 2018. Experimental results show that our method not only significantly increases the inference speed to 73.8 frame per second (FPS), but also attains comparable state-of-the-art performance.

1. Introduction

Human pose estimation is a computer vision problem that aims at recovering the posture of a person via localising joints and rigid parts from images. The obtained pose information can be used to inform other computer vision problems such as abnormal behaviour detection, human behaviour analysis and action recognition. It can also be used in a variety of applications such as smart environments, human-computer interaction, augmented reality and virtual reality.

Over the past few years, with the development of deep learning, human pose estimation from RGB images has made significant progress. For scenes containing multiple people, the estimation accuracy has increased by 35% (Andriluka et al., 2018b). Typical deep learning-based methods for pose estimation consists of two modules - joint detection and refinement (Wei et al., 2016). The joint detection module generates joint candidates for each joint type. Since different body joints may have similar appearances, such as the left and the right knees, using the joint detection module alone is not enough to distinguish all the joints. The refinement module is therefore implemented, which takes the output from the joint detection module and introduces a higher-level context to improve the decision process. Such a module usually consists of multiple stacked convolutional layers to increase the receptive field, thereby obtaining more contextual information. In Cao et al. (2017), 6 stages of refinement modules are implemented, with each stage consisting of 7 convolutional layers. In Newell et al. (2017), 4 stages of stacked hourglass-like refinement modules are used, with each hourglass consisting of 45 convolutional layers. As indicated in their experiments, the improvement does not increase linearly with the number of refinement modules. In general, the sub-sequence modules contribute smaller accuracy improvements. Additionally, if further integrating with other extension modules, such as 3D pose estimation (Tome et al., 2017), video-based pose estimation and tracking (Andriluka et al., 2018a), such a large number of refinement modules, together with the backbone network, will constrain the application areas due to speed and memory limitations, causing difficulties in training an end-to-end system. This motivates us to improve the potential of the backbone network and to propose

[**] Corresponding author.
e-mail: hubert.shum@northumbria.ac.uk (Hubert P. H. Shum)

a more efficient form of convolutional feature utilisation.

Our studies indicate that with powerful backbone networks, such as VGG-19 (Simonyan and Zisserman, 2015) and ResNet-50 (He et al., 2016), joint detection can already obtain acceptable localisation performance. The major source of error is the confusion between the left and right joints of the same type, as illustrated in Fig. 1(a). In order to address this, additional information that can strengthen the identification of human body parts is required. As analysed by Chu et al. (2016), different joints and limbs are highly correlated at lower feature-levels. In theory, the feature maps of neighbouring joints and limbs should be concurrently activated. For example, the left thigh, left hip and left knee typically have coincident activation responses, and any two of these three parts can sustain the discrimination of the third one. While this kind of clues should be capable of constructing a complementary relationship for inference, previous methods of multi-person pose estimation consider the joints independently, failing to harness the power of the concurrent feature information for pose estimation reinforcement.

In this paper, we propose a feature transfer structure to exploit the complementary feature information between joints and limbs, which effectively strengthens the recognition of human body parts. The proposed feature transfer structure has two major functions. First, it translates the activated region of one joint to that of the next adjacent joint in their respective feature maps. Such translated features can then be formed as complementary features. Second, it converts the features between the joint type and the limb type in both directions, facilitating feature fusion. Since the feature translation can be regarded as a matrix translation operation, and the converted features are represented as convolutional features, the feature transfer structure can be implemented with convolutional layers and its parameters can be learned effectively by backpropagation (Goodfellow et al., 2016a). This allows us to merge the feature transfer structure into the pose estimation network using convolutional layers, forming an end-to-end system. On top of this, our network automatically learns transfer relationships from supervisory data, and does not require manually defined neighbour joint information as in Yang et al. (2016) and Chu et al. (2016). This enables a more general design with fewer network layers.

We perform experiments on three popular multi-person pose estimation datasets, MPII (Andriluka et al., 2014), COCO 2017 (Lin et al., 2014) and PoseTrack 2017 and 2018 (Andriluka et al., 2018a). With only 4 stacked convolutional modules for feature transfer and 1 refinement module for combining context information, our method achieves comparable performance to existing approaches with two times more weighting parameters. This indicates the effectiveness of feature transfer in improving the accuracy of pose estimation. With the benefit of the decreased number of parameters, the forward inference speed of the whole network achieves 42.2 FPS, and further attains 73.8 FPS by optimising the implementation, when using an input size of 368×432 on a single NVIDIA Tesla P40 GPU. This enables real-time applications with consumer-level hardware. We open our source code for further research and development in the field.

The contributions of this work are summarised as follows:

- We propose a new design of a multi-person pose estimation network by introducing feature transfer, which utilises the complementary features of joints and limbs to strengthen the identification of human body parts. This reduces the needs of deep refinement modules.
- We propose and validate an implementation strategy of the feature transfer sub-network. First, the sub-network is implemented with convolution layers and therefore is mergeable with the backbone pose estimation network to form an end-to-end system. Second, it learns the transfer relationship automatically from ground-truth information, resulting in a more general and efficient implementation.
- Using our open-source implementation, we perform extensive experiments on three popular datasets - MPII, COCO 2018 and PoseTrack 2017 and 2018, which demonstrate that our system significantly improves the run-time speed while attaining comparable state-of-the-art performance, enabling real-time applications with consumer-level hardware.

The source code of our system can be downloaded at: http://hubertshum.com/CVIUSourceCode.zip

The rest of the paper is organised as follows. Section 2 introduces the related work. Section 3 describes our proposed approach. The experiment dataset, experiment results and analysis are presented in Section 4. The work is concluded in Section 5.

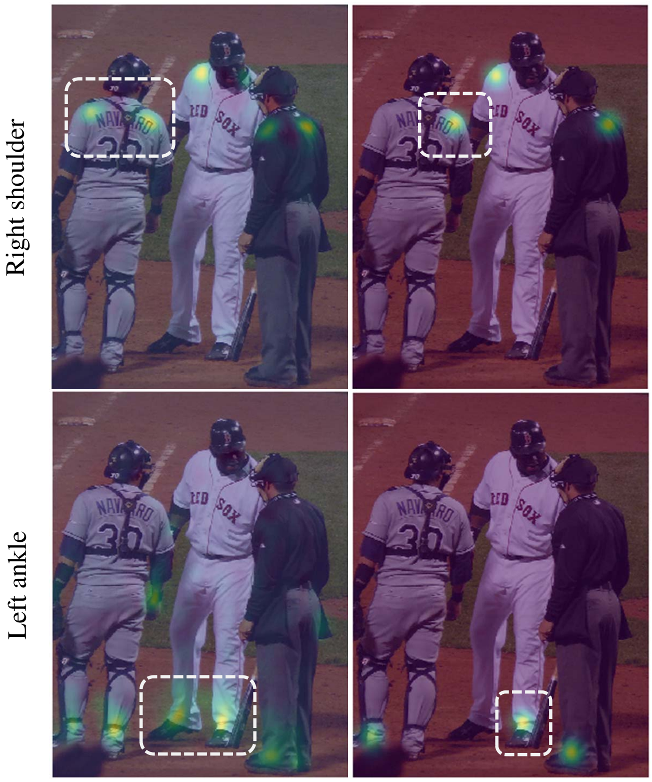
2. Related Work

The problem of human pose estimation has a long history in computer vision. Before the discovery of deep convolutional features, pose estimation research mainly focused on building local or global human body descriptive models (Andriluka et al., 2009; Yang and Ramanan, 2011; Gkioxari et al., 2013; Pishchulin et al., 2013; Sapp and Taskar, 2013) and predicting body parts based on predefined hand-crafted features. Recent advancement in deep learning significantly improves the performance. Since a comprehensive review of pose estimation approaches is beyond the scope of this paper, here, we focus on some recent work based on deep neural networks.

2.1. *Single Person Pose Estimation*

Earlier human pose estimation research mainly focuses on the case of a single person. It considers a cropped region that contains the person in order to predict the corresponding joint locations. In this case, prediction does not involve joint grouping.

Convolutional neural networks (CNNs) play an important role in pose estimation. Toshev and Szegedy (2014) use CNNs and a cascaded structure to predict and refine the joint locations. Tompson et al. (2014) combine CNNs with a Markov random field to explore the spatial constraints of body parts. Wei et al. (2016) propose stacking multi-stage refinement modules of fully convolutional networks to expand the receptive field, thereby obtaining the context information that supports



(a) Before transfer (b) After transfer

Fig. 1: Two examples of activation maps before and after feature transfer. (a) The backbone network can localise (upper) the shoulder joints and (lower) the ankle joints, but cannot distinguish the left/right joint type. (b) After adding the feature transfer sub-network, the discrimination power of the networks is strengthened and the joint detector can recognise (upper) the right shoulder and (lower) the left ankle.

the inference of joints. Different from Wei et al. (2016), Newell et al. (2016) design stacked hourglass-like networks to recover the spatial resolution of the output heatmaps while maintaining the high-level semantic features. Based on Newell et al. (2016), Chu et al. (2017) introduce the conditional random field and visual attention models to capture different granularity information. Yang et al. (2017) introduce feature pyramids to obtain multi-scale joint feature information. Kawana et al. (2018) propose clustering different poses before training the networks. Hong et al. (2015) and Hong et al. (2016) propose 3D pose recovery methods based on hypergraph learning. Hong et al. (2014) adopt silhouette-based locality-sensitive sparse coding to recover 3D human pose.

Structured models are further introduced to capture the human hierarchy structure. Chang and Lee (2018) propose a conditional random field model to measure the plausibility of human poses. Yang et al. (2016) construct message passing layers for describing the pair-wise relationship among body parts. In comparison, Chu et al. (2016) build a tree-structured model to translate the feature information using convolutional layers from one joint to another neighbouring joint.

Our method also develops a structured model for human pose. However, instead of manually defining the paths of message passing for different joints as in Yang et al. (2016) and Chu et al. (2016), our method learns the joint-limb relationships autonomously from ground-truth information in an end-to-end configuration. This results in a more general and efficient implementation.

2.2. *Multi-person Pose Estimation*

Estimating the poses of multiple people in a scene is more practical in real-world applications. Apart from challenges such as the complexity of appearance, the variety of gestures, occlusions and multiple scales, the inclusion of multiple people in the same image introduces another challenge — the system has to distinguish not only the type of the body part but also the person that the part belongs to. There are two main categories of solutions: top-down methods and bottom-up methods.

2.2.1. *Top-down Methods*

Top-down methods can be seen as a two-stage pipeline from global (i.e. the bounding box) to local (i.e. joints). The first stage is to perform human detection and to obtain their respective bounding boxes in the image. The second stage is to perform single person pose estimation for each of the obtained human regions. Fang et al. (2017) propose a symmetric spatial transformer network and a parametric pose non-maximum suppression to handle the inaccurate and redundant human detection bounding boxes. He et al. (2017) deploy an interpolation approach in the region pooling process to generate more accurate feature maps, which compensates for the effect of region pooling caused by missing feature information. Papandreou et al. (2017) directly increase the output scale of the region pooling by up-sampling the output to a much larger size than that of He et al. (2017) (257×353 *vs.* 56×56). Experiments show that this simple strategy is very effective in improving the accuracy. Similarly, Xiao et al. (2018) deploy a few deconvolutional layers on a backbone network to increase the resolution of feature maps and results in obtaining higher accuracy. Sun et al. (2019) propose a high-resolution network to obtain stronger representations and achieve top results on a wide range of vision tasks. These results indicate that the localisation accuracy of pose estimation greatly relies upon high-resolution feature information due to the generally small size of the body parts. Based on He et al. (2017), Alp Güler et al. (2018) design a structure that merges the feature information from other tasks such as 3D pose detection, for better 2D pose estimation during the training and forward inference.

For top-down methods, high-precision feature maps of human regions are important to maintain the accuracy of joint localisation. We also notice that the run time of these methods is affected by both the speed of human detection and that of single person pose estimation. The latter is proportional to the number of people in the view, and is problematic for applications requiring a consistent frame rate.

2.2.2. *Bottom-up Methods*

In order to handle the problem of processing speed, bottom-up methods have received increasing attention from researchers in recent years. They approach this problem from the opposite direction, in which they detect all of the body parts before grouping and associating the parts with the relevant person.

Pishchulin et al. (2016) modify a general object detector to detect body parts and partition body part candidates into person clusters by solving an integer linear program problem. Based on Pishchulin et al. (2016), Insafutdinov et al. (2016) introduce an incremental optimisation strategy to improve the grouping speed. Belagiannis and Zisserman (2017) validate that CNNs can generate heatmaps for both body joints and limbs using continuous regression. Levinkov et al. (2017) consider the articulated human body pose estimation as a combinatorial optimisation problem and propose two local search algorithms that offer a feasible solution at given time constraint. Varadarajan et al. (2018) exploit the inherent structure of the human body to decrease the complexity of the body part grouping model. Newell et al. (2017) propose learning both body part detection and grouping in the CNNs simultaneously. They also indicate that the accuracy bottleneck of pose estimation is not joint grouping but joint prediction. They find that by replacing the joint predictions with the groundtruth, the final pose estimation results can be improved from 60% to 94%, which is close to saturation. Cao et al. (2017) propose a limb descriptor known as the part affinity fields (PAFs), which provides additional limb information including the type, direction and length for better group assignments. They further deploy multiple very deep refinement modules to obtain accurate joint and limb information.

Our paper follows the bottom-up approaches as they have the advantage of a consistent frame speed by detecting all body parts of all people in a single shot. Motivated by Cao et al. (2017), we also employ a similar limb descriptor as it has high-efficiency in body part grouping. However, we introduce the mutually supportive relationship between joints and limbs for pose estimation. Experiments show that this structure effectively improves the accuracy of body part detection as shown in Sec. 4.6.2.

2.2.3. *Multi-task Learning*

Multi-task learning is an approach to solve multiple learning tasks at the same time, while exploiting their shared representations and differences (Caruana, 1997). Knowledge (feature) transfer is a related concept to multi-task learning. However, shared representations are developed concurrently in multi-task learning while feature transfer is to learn a sequentially shared representation. There are three equivalent ways to learn a shared representation, which are via a regulariser (Ciliberto et al., 2015), an output metric (Dai et al., 2016) and an output mapping (Kim and Xing, 2010), respectively. In the framework of CNNs, we use an output metric and learning the structure of feature transfer by the network outputs and loss to facilitate the learning process. In our experiments, we validate that joint and limb's feature can be transferred into the type of each other.

3. The Proposed Deep Feature Transfer Network

The proposed method consists of four components, which are visualised in Fig. 2. The first one is the body joint and limb detector, which employs fully convolutional layers to regress the location of each body part in the image (Sec. 3.1). The second component is the feature transfer networks, where the feature information of body joints and limbs are transferred into the representations of each another to strengthen their feature discrimination power (Sec. 3.2). The third component is a single refinement module (Sec. 3.3). Since both body part detector and feature transfer networks only handle the locally interested body parts, such a module helps to capture global context information and improve prediction accuracy. The refined feature information is used to produce the body joint and limb heatmaps, which are input into the final component - the body part grouping model (Sec. 3.4). It matches each pair-wise body joint according to the limb information and the matched pair-wise joints are assembled into a full-body group for each person in the image.

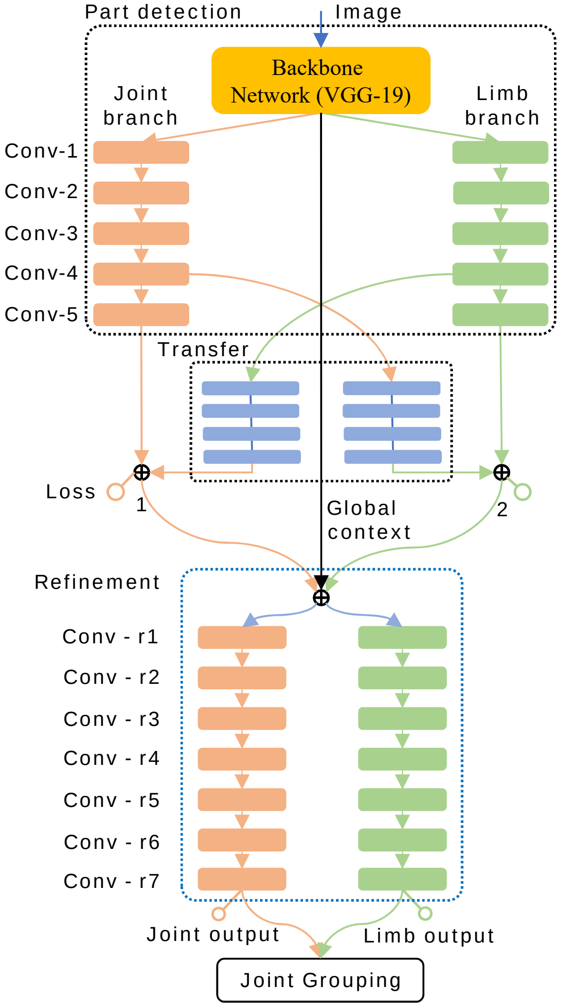


Fig. 2: The architecture of our proposed method includes 4 components, which are a part detection network, a feature transfer sub-network, a refinement module and joint grouping. The backbone network takes an image as input and outputs the abstract features. Then the feature information enters the two detection branches, which are coloured in orange and green, respectively. The transfer sub-network (blue blocks) extracts features from Conv-4 and outputs the transferred features to merge with the features of the detection branch. The merged features produce the score maps for both joint and limb branches. At last, the refinement module is used to capture context information to refine the results. (Best viewed in colour)

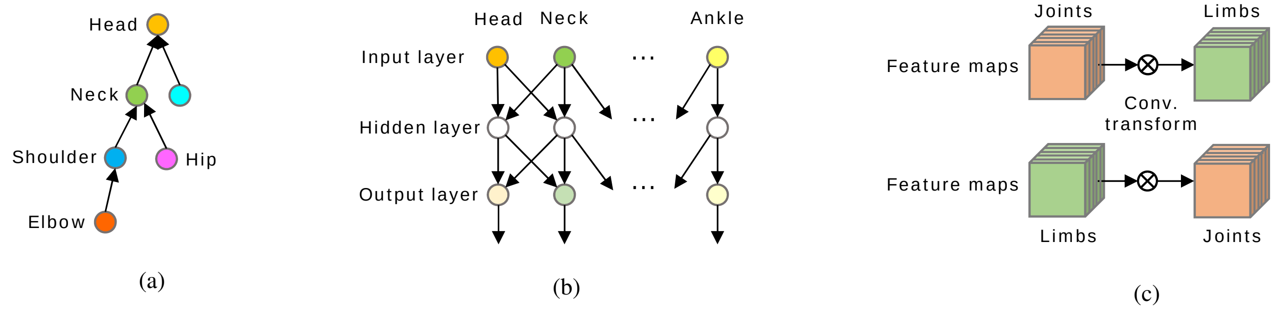


Fig. 3: Comparisons of different feature transfer models. (a) Tree model (Chu et al., 2016), which has a predefined tree structure. (b) Loopy model (Yang et al., 2016), which includes predefined loop feature flows. (c) Our one-shot transfer model, which requires no predefinition.

3.1. Body Joint and Limb Detection

Since deep CNNs possess the capacity to effectively deal with a wide variety of objects and have been validated in many vision-related tasks, such as object detection (Liu et al., 2016) and segmentation (Long et al., 2015), we deploy deep CNN in our method for pose estimation. To avoid interference, we set a two-branch head network for multi-task learning. Referring to Fig. 2, we use the first ten layers of VGG-19 (Simonyan and Zisserman, 2015) as the backbone network to extract general low-level convolutional features. Then, the feature streams enter two branches, namely the joint branch and the limb branch, for the respective high-level detection task. Each branch consists of 5 convolutional layers. Each of the first four convolutional layers is followed by a ReLU layer (Nair and Hinton, 2010). The only exception is the fifth convolutional layer, Conv-5, as it is the output layer of the body part score maps and does not require nonlinear rectification.

There are two possible methods for body parts detection. The first one considers the localisation of body parts as a problem of discrete classification, which adopts a univariate loss function $\ell(h,(x,y)) = \Pi_{[h(x)\neq y]}$ to learn a hypothesis $h : x \to y$ that assigns a body part label y for each pixel in the input image x (e.g., softmax (Goodfellow et al., 2016b)). The second method is to fit a continuous regression function $\ell(h,(x,y)) = (h(x)-y)^2$ to generate predictions. In CNNs, this corresponds to training the network that produces the confidence values of different object types at each pixel position. Since the classification method cannot provide a smooth transition for the pixels near the annotated joints, we adapt the regression method in predicting the confidence maps of body parts.

We use $\mathbf{J}$ and $\mathbf{L}$ to represent the joint and limb confidence maps, where $\mathbf{J} = (\mathbf{J}_1, ..., \mathbf{J}_i, ..., \mathbf{J}_m)$, $\mathbf{L} = (\mathbf{L}_1, ..., \mathbf{L}_j, ..., \mathbf{L}_n)$, $i \in [1, m]$, $j \in [1, n]$, m and n are the numbers of the predefined joint and limb types respectively, $\mathbf{J}_i \in \mathbb{R}^{w\times h}$, $\mathbf{L}_j \in \mathbb{R}^{w\times h\times 2}$, w and h are the width and height of the confidence maps. For the joint ground-truth confidence maps, to smooth the training loss, a Gaussian distribution is generated around the location of each annotated visible joint $\mathrm{p}^*_{i,c} \in \mathbb{R}^2$, $c \in [1, k]$, and k is the number of visible joints of type i. The ground-truth value $\mathbf{J}^*_i(\mathrm{p})$ at position $\mathrm{p} \in \mathbb{R}^2$ on the ground-truth confidence maps is defined as:

$$\mathbf{J}^*_i(\mathrm{p}) = \max_{\mathrm{c}\in[1,\mathrm{k}]} \exp\left(-\frac{\|\mathrm{p} - \mathrm{p}^*_{\mathrm{i,c}}\|_2^2}{\sigma^2}\right) \tag{1}$$

where σ is the variance. For the limb ground-truth score maps, there are several similar representation methods that can be used to describe the limb such as the limb spot (Belagiannis and Zisserman, 2017) and the part affinity fields (PAF) (Cao et al., 2017). To facilitate the comparison with PAF, we select PAF as the limb descriptor in our system. This method represents a limb with an ellipse between two neighbouring joints. The pixels within the ellipse are considered to be the limb region. Each of the pixels has a unit vector that points to the next joint. For the pixels outside the limb region, the vector is zero-valued. The limb ground-truth confidence maps are defined as:

$$\mathbf{L}^*_j(\mathrm{p}) = \begin{cases} \dfrac{(\mathrm{p}^*_{i_1,\mathrm{h}} - \mathrm{p}^*_{i_2,\mathrm{h}})}{\|\mathrm{p}^*_{i_1,\mathrm{h}} - \mathrm{p}^*_{i_2,\mathrm{h}}\|_2}, & \text{if pixel p on the limb of person h} \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

The loss functions of the branches are defined as:

$$e_{\mathbf{J}} = \sum_{i=1}^{m} \sum_{\mathrm{p}} \mathrm{W}(\mathrm{p}) \|\mathbf{J}_i(\mathrm{p}) - \mathbf{J}^*_i(\mathrm{p})\|_2^2 \tag{3}$$

$$e_{\mathbf{L}} = \sum_{j=1}^{n} \sum_{\mathrm{p}} \mathrm{W}(\mathrm{p}) \|\mathbf{L}_j(\mathrm{p}) - \mathbf{L}^*_j(\mathrm{p})\|_2^2 \tag{4}$$

where W(p) is the binarized mask to ignore unannotated people in the loss computation.

3.2. Feature Transfer

As mentioned in the introduction, the features of body joints and limbs should be concurrently activated. This complementary information can be used to mutually support the inference of both joints and limbs. Thus, we design a transfer sub-network to cross-transfer the features from one branch to another, which is visualised as the blue blocks in Fig. 2.

Let $\mathrm{A_b}$ be the feature maps of the branch b, the transferred feature maps, $\mathrm{A_b^T}$, is calculated as:

$$\mathrm{A_b^T} = F(\mathrm{A_b} \otimes f^{\mathrm{T}}) \tag{5}$$

where f^{T} is the filter bank for feature transfer, $\otimes$ is a convolution operation, and F is the rectified linear unit.

We observed that the largest distance between adjacent joints on the input training images of networks is within 100 pixels. Therefore, we implement the transfer sub-network with

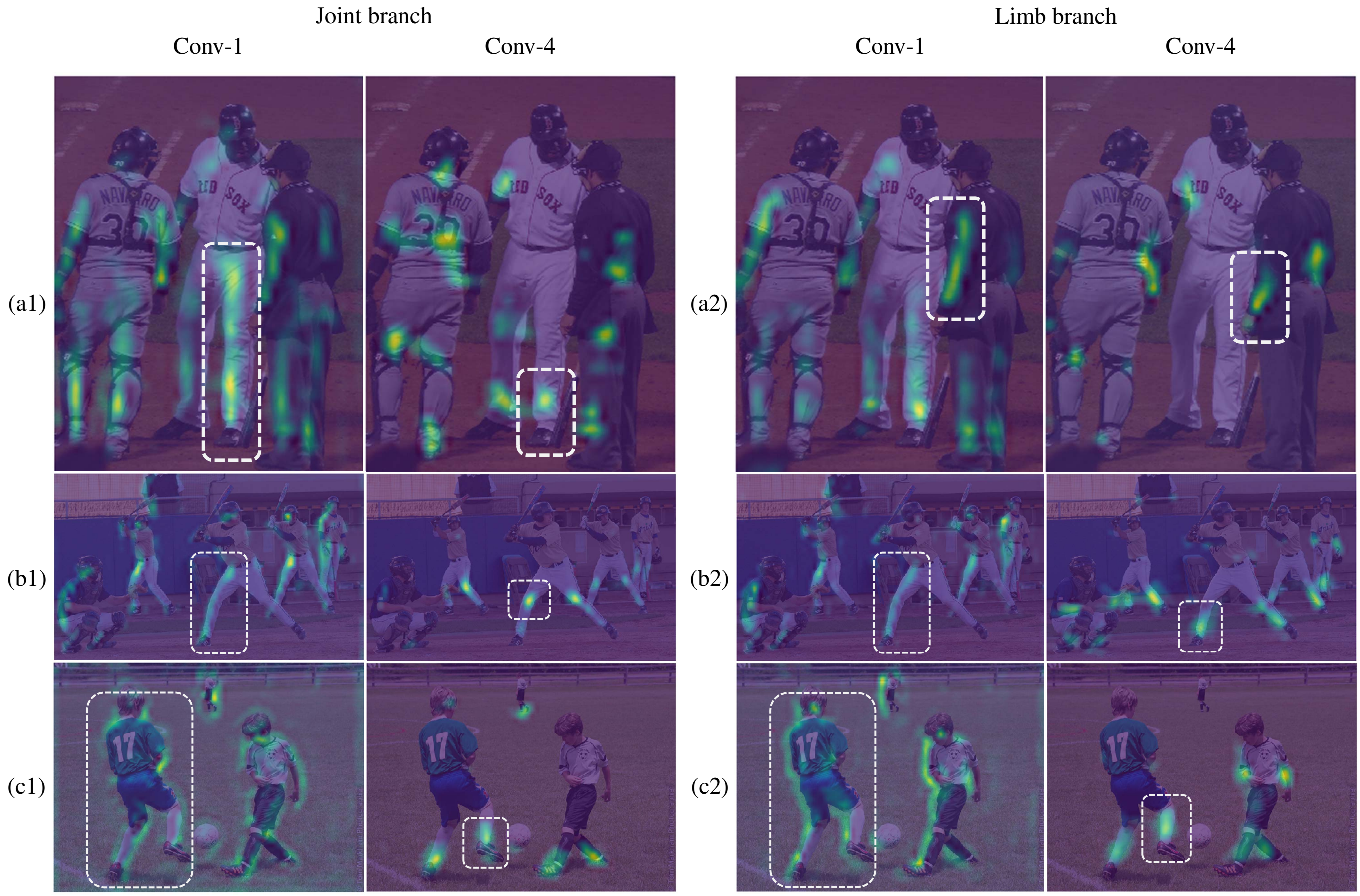


Fig. 4: Examples of feature maps extracted from Conv-1 and Conv-4 of the joint and limb branches respectively. (a1), (b1), and (c1) show example feature maps of Conv-1 and Conv-4 of joint branch. (a2), (b2), and (c2) show example feature maps of Conv-1 and Conv-4 of limb branch. For both branches, the features of Conv-1 are more abstract, while that of Conv-4 are more distinctive, as highlighted.

4 transfer blocks to provide the corresponding receptive field, where each block contains 3 convolutional layers with a kernel size of 3×3 and the output features of them are aggregated to strengthen feature propagation.

We design a more effective feature transfer scheme compared to existing works. As illustrated in Fig 3, Chu et al. (2016) utilise a predefined tree structure with over 100 convolutional layers, and Yang et al. (2016) utilise a predefined loopy model, to transfer the feature maps between neighbour joints for a single person. Both models explicitly encode the one-to-one relationship between joints. Adapting such single person models to deal with multiple people implies that a much larger network structure is required. As a solution, we transfer the feature maps of all joints and that of all limbs in a single shot using four convolution modules, in which the feature maps of different joints and that of different limbs are represented as two respective stacked blocks. This design allows the network to autonomously learn the complementary features while significantly reducing the number of layers required, thereby saving the computation cost.

Here, we explain how we select the suitable convolutional layer for feature transfer. We analyse the feature maps extracted from Conv-1 and Conv-4 of the joint branch and the limb branch respectively. Some examples are shown in Fig. 4. We observe that the features in Conv-1 of both branches are abstract and less distinctive. This shows that the feature maps cannot effectively represent the joints and the limbs. In contrast, Conv-4 of the joint and the limb branches shows more distinctive and concrete joint and limb features respectively. We can observe that the concreteness of the feature maps increases with the convolution depth, and that Conv-4 of each branch produces results that are distinctive enough for feature transfer. Therefore, we connect Conv-4 to the transfer sub-network for feature transfer. Notice that Conv-5 is the output layer which has only the confidence and location information, and is not suitable for feature transfer. In Sec. 4, we give quantitative comparisons to validate the selection of the transfer layer.

Through the transfer sub-network, the features are transferred into the feature type of the other branch, and are merged with the features of the other branch. Because the transfer sub-network is composed of convolutional layers, it can be combined into the backbone network to form an end-to-end system. Qualitative and quantitative results are provided in Sec. 4 to show that the transfer sub-network can convert and translate the features between the branches. We also show that combining the transferred features with the detection features effectively improves the performance of pose estimation as shown in Sec. 4.6.2.

3.3. Refinement with Context Information

Since the previous body-part detector and feature transfer networks only handle the locally interested body parts, it is necessary to include global context information to further improve the prediction performance. As analysed in Sec. 3.2, the features before/on Conv-1 are more abstract and global. Therefore we extract global context from the bifurcation layer of the backbone network. This approach is also similar to the methods employed by Wei et al. (2016) and Newell et al. (2016).

The architecture of the refinement module is shown in Fig. 2. It concatenates the heatmaps of joint and limb detection with the feature maps of the backbone network and takes them as the input. The input then enters two refinement branches to refine joint and limb detections separately. Both branches use the same network configuration, consisting of 7 modules which have the same layer structure as used in the transfer subnetwork. Each convolutional layer in the module has 128 channels with each followed by a ReLU layer except for the last output layer. The 7 stacked modules increase the receptive field of the network and enable the network to capture context information around the predictions to refine them.

3.4. Group Assignments

Here, we assemble a full body group of joints and limbs for each person. The outputs of body joint and limb branches are the confidence maps of the respective type. We perform a non-maximal suppression of 4-neighbourhoods over each score map and choose the pixels with the largest score in every search as the corresponding candidate body part.

Given two pair-wise candidate joints, $\mathbf{J}^+_{i_1}$ and $\mathbf{J}^+_{i_2}$, from a predefined kinematic chain, their matching score is computed by the cosine similarity between their line segment and the limb unit vector. More specifically, the matching score is approximated by:

$$s = \sum_{d=1}^{D} \mathbf{L}^+_j(\mathrm{p}(d)) \frac{(\mathbf{J}^+_{i_1} - \mathbf{J}^+_{i_2})}{\|\mathbf{J}^+_{i_1} - \mathbf{J}^+_{i_2}\|_2} \tag{6}$$

where D is the total number of equidistant line segments between two joints, and is set as 10 following existing works.

After computing the matching scores of all the candidate joint pairs, we search for the definite connections of the human skeletons according to the matching score. We follow the search approach of Cao et al. (2017) due to its high efficiency. For a predefined skeleton connection, the search starts from the connection with the highest score. The obtained connection is considered a definite connection. Then, it finds the next connection with the second highest score. If such a connection has no duplicate joints with the previous definite connection, it will be preserved. Otherwise, it will be removed. The system repeats the search until no candidate connections can be found. This process allows us to obtain the definite connections of all the predefined skeleton connections. Finally, we assemble the definite connections that share the same joint to form the complete human skeletons of multiple people.

4. Experimental Results

4.1. Datasets

We perform qualitative and quantitative experiments on the three most popular multi-person pose estimation datasets: MPII Human Pose (Andriluka et al., 2014), MS-COCO 2018 Keypoints Challenge dataset (Lin et al., 2014) and PoseTrack 2017 and 2018 dataset (Andriluka et al., 2018a).

The MPII Human pose contains 24,589 images, in which 17,408 images are split as the training set with 28,883 annotated people. During the testing stage, the evaluation focuses on different regions in an image, and one image may include one or more regions that consist of a non-identical number of people. Pishchulin et al. (2016) defines a set of 1,758 regions with rough position and scale information as the test set and provides an evaluation tool to calculate mean Average Precision (mAP) of the whole body joint prediction. The accuracy results are evaluated and returned by the staff members of the MPII dataset.

MS-COCO 2018 keypoint detection dataset (Lin et al., 2014) consists of training, validation and testing sets. On the COCO 2018 training and validation sets, there are 118,287 and 5000 images respectively, totally containing over 150,000 people with around 1.7 million labelled keypoints. For open testing, the testing set has two splits: test-dev and test-challenge. Each split contains roughly 20,000 images. We train our models on the training set and perform ablation experiments on the validation set. The model is evaluated on the test-dev set and the accuracy results are obtained from the online evaluation server for public comparisons.

PoseTrack 2018 dataset (Andriluka et al., 2018a) is split into 593, 74 and 375 videos for training, validation and testing, respectively. The videos in the training set consist of 18,064 image frames. After filtering out some bad cases according to our defined rules, the number of effective annotated human instances in the training set is 85,967. PoseTrack 2017 dataset is split into 300, 50 and 214 videos for training, validation and testing, respectively. The annotation has defined 15 body keypoints. The dataset contains three challenges, single-frame, multi-frame pose estimation and pose tracking. Here we focus on the first challenge, i.e. single-frame multi-person pose estimation.

4.2. Evaluation Protocols

Both MPII and PoseTrack multi-person pose estimation datasets use the mean average precision (mAP) as the evaluation metric, similar to Yang and Ramanan (2011). First, multiple people's pose predictions are generated and are assigned to the groundtruth according to the highest $\mathrm{PCK_h}$ matching score (Andriluka et al., 2014). Each groundtruth can possess only one prediction. Unassigned predictions are counted as false positives. Furthermore, the average precision (AP) for each body part type is computed over all person instances and the mAP is reported over all body part types.

On the COCO keypoint dataset, 5 metrics are used to describe the performance of a model. They are AP (i.e. average precision), $\mathrm{AP}^{0.5}$, $\mathrm{AP}^{0.75}$, AP^{M}, AP^{L}, as illustrated in Table 1.

Table 1: Evaluation metrics on the COCO dataset

Metric	Description
AP	AP at OKS*=0.50:0.05:0.95 (primary metric)
$AP^{0.5}$	AP at OKS=0.50
$AP^{0.75}$	AP at OKS=0.75
AP^{M}	AP for medium objects: 32^2 < *area* < 96^2
AP^{L}	AP for large objects: *area* > 96^2

*OKS–Object Keypoint Similarity, same role as IoU

In order to assign predictions to groundtruth, an object keypoint similarity (OKS) is defined to compute the overlapping ratio between groundtruth and predictions in terms of point distribution (Lin et al., 2014). Here the OKS plays the same role as the intersection over union (IoU) in the case of object detection. Thresholding the OKS adjusts the matching criterion. All metrics computed allow a maximum of 20 top-scoring predictions per image. Notice that in general applications, $AP^{0.5}$ gives good accuracy already. AP (averaged across all 10 OKS thresholds) is a stricter metric in which 6 of the OKS matching thresholds exceed 0.70.

4.3. *Implementation Details*

We train the network with an input size of 368×368 and an output scale of 46x46. The ratio of the network input to output size is 8.0. We utilise the SGD method to optimise the network weights. Optimisation super-parameters are selected as: 4e-5 initial learning rate, 0.9 momentum, 0.0005 weight decay, and a batch size of 28. During training, we use a person-centric sampling strategy. The augmentation of each sample in a batch is focused on one person-instance. For example, an image is first scaled so that the height of the selected person in the image is around 220 pixels (a ratio of 0.6), then the image is randomly augmented by rotating, scaling and flipping using the centre of the selected person as the centre of transformation. Lastly, a patch of 368×368 is centred on the selected instance and cropped from the image. The regions out of the image are padded with a value of 128. In order to learn a detection confidence within the range of [0, 1] and smooth the training gradients, the pixel values of the cropped patch are normalised by 256 and are subtracted by 0.5. The implementation is built on the open-sourced Caffe framework (Jia et al., 2014). During testing, for the single scale evaluation, an input image is scaled to the height of 368 with the length-to-width ratio is maintained. For the multi-scale evaluation, an input image is scaled to four sizes with a gap of 0.25 and the heatmaps of joints and limbs are averaged across sizes.

4.4. *Comparison with State-of-the-art Methods on Accuracy*

For the MPII test subset, our approach outperforms other methods in computational time and achieves comparable performance in accuracy (within 1.2%) with PAF(Cao et al., 2017), as illustrated in Table 2. In this case, we use the network's depth of 26 compared to PAF's network depth of 50, resulted in a significantly increased frame speed. Specifically, the precision of head detection achieves a very high value of 92.7%, this being the case due to appearance variation and occlusion not affecting the human head as prominently. For the remaining body parts, the accuracy of the upper-body is higher than that of the lower body due to an increased chance of occlusion in the lower body areas of the dataset. In addition, the upper body and lower body show a great class-imbalance in the dataset. We find that the number of visible human ankle joints is lower than that of the upper body joints (e.g. shoulder, wrist and elbow) by about 25%. Therefore the lower precision values for ankle joints is not unexpected. Note that the accuracy of ankle identification found in our method is higher than PAF.

For the COCO test-dev set, we achieved the same performance with PAF (Cao et al., 2017) while attaining a 2 times faster speed of 42.2 FPS, as illustrated in Table 3. From our observation, the large human instances have higher precision and recall than medium-sized human instances. For the $AP^{0.5}$ metric our method achieves a very high value of 0.821. In Table 6, we show that our model outperforms PAF by 1.4% in the single-scale evaluation and has the same performance to PAF in the multi-scale evaluation.

For the PoseTrack dataset, the comparison results are presented in Table 4. Our method outperforms Detect&Track (Girdhar et al., 2018) and AlphaPose (Xiu et al., 2018) on the PoseTrack 2017 validation and testing sets. On the 2017 validation set, for the parts of shoulder and hip, the accuracy of our method is higher than or comparable to Xiao et al. (2018). On the 2017 testing set, our method is ranked 2nd for most of the body parts. On the 2018 validation, our method outperforms Xiao et al. (2018) in the detection accuracy of several body parts, such as shoulder, elbow and hip, by 5%, 2.1% and 1%, respectively. In addition, the speed of our method is much faster than the other algorithms we compared in this experiment.

4.5. *Comparison with State-of-the-art Methods on Computational Complexity*

Our networks consist of a backbone, a feature transfer module and a refinement module. The parameter number of the whole network is 21,278,912. In contrast, the total amount of parameters in the PAF(Cao et al., 2017) is 52,298,816, which is two times more than our network. The parameter number of each module of our network is shown in Fig. 5. For the run time of our approach, we record the inference time on a desktop with one NVIDIA Tesla P40 GPU over 1000 images, which include different numbers of people from 1 to 20. The whole network with 368×432 sized inputs only costs 22.71 ms on average (i.e. 44.0 FPS). Group assignment takes 0.2 ms for 2 people and 0.6 ms for 10 people. This shows that our network has higher inference efficiency due to the contribution of feature transfer.

We also use the same environment for speed comparisons with the state-of-the-art, except Papandreou et al. (2017) who have not released the source code, as illustrated in Table 2 and Table 3. In Table 5, we compare our method with two typical top-down and bottom-up methods in terms of computational complexity. We observe that our method is 9.6 times faster than the top-down one, and 2.1 times faster than the bottom-up method. In addition, our model has smaller model size, number of parameters and FLOPs than other models, which accelerates

Table 2: Comparisons of different methods on the MPII test subset of 288 images. Bold: the best performance. Bold-Italic: comparable or better performance than PAF.

Method	Head	Shoulder	Elbow	Wrist	Hip	Knee	Ankle	**mAP**	FPS
Iqbal and Gall (2016)	70.0	65.2	56.4	46.1	52.7	47.9	44.5	54.7	0.1
DeeperCut(Insafutdinov et al., 2016)	87.9	84.0	71.9	63.9	68.8	63.8	58.1	71.2	0.005
AE(Newell et al., 2017)	91.5	87.2	75.9	65.4	72.2	67.0	62.1	74.5	6.5
PAF(Cao et al., 2017)	**92.9**	**91.3**	**82.3**	72.6	**76.0**	70.9	66.8	79.0	20.4
AlphaPose(Fang et al., 2017)	89.3	88.1	80.7	**75.5**	73.7	**76.7**	**70.0**	**79.1**	3.4
Our method	***92.7***	89.3	80.0	71.3	73.8	***70.0***	***67.6***	77.8	**42.2**

Table 3: Comparisons of different approaches on the COCO 2018 test-dev set. Bold: the best performance. Bold-Italic: comparable or better performance than PAF. Papandreou et al. (2017) have not released source code.

Method	FPS	**AP**	$\mathrm{AP}^{0.5}$	$\mathrm{AP}^{0.75}$	AP^{M}	AP^{L}
Papandreou et al. (2017)	-	0.605	0.822	0.662	0.576	0.666
MaskRCNN(He et al., 2017)	4.4	0.627	**0.870**	0.684	0.574	0.711
AE(Newell et al., 2017)	6.5	**0.655**	0.868	**0.723**	**0.606**	**0.726**
PAF(Cao et al., 2017)	20.4	***0.584***	0.815	0.626	0.544	0.651
Our method	**42.2**	***0.584***	0.821	0.626	0.537	0.658

Table 4: Comparisons of different methods on the PoseTrack 2017 and 2018 dataset. Bold: the best performance. The results of Xiao et al. (2018) is trained and tested by us since the original implementation has not performed training on the PoseTrack 2018 dataset.

Method	Head	Shoulder	Elbow	Wrist	Hip	Knee	Ankle	**mAP**	FPS
PoseTrack 2017 Validation									
Detect&Track(Girdhar et al., 2018)	67.5	70.2	62	51.7	60.7	58.7	49.8	60.6	4.4
AlphaPose-PoseFlow(Xiu et al., 2018)	66.7	73.3	68.3	61.1	67.5	67.0	61.3	66.5	3.4
JointFlow(Doering et al., 2018)	-	-	-	-	-	-	-	69.3	0.2
Xiao et al. (2018), ResNet50	**79.1**	80.5	**75.5**	**66.0**	**70.8**	**70.0**	**61.7**	**72.4**	9.1
STAF-SS(Raaj et al., 2019)	-	-	-	55.0	-	-	53.5	64.6	27
Our method	65.0	**81.6**	72.8	60.8	69.2	63.3	54.7	66.6	**42.2**
PoseTrack 2017 Testing									
BUTD(Jin et al., 2017)	74.7	71.9	65.6	56.4	62.2	57.5	51.0	63.6	-
Detect&Track(Girdhar et al., 2018)	-	-	-	-	-	-	-	59.6	4.4
AlphaPose-PoseFlow(Xiu et al., 2018)	64.9	67.5	65.0	59.0	62.5	62.8	57.9	63.0	3.4
JointFlow(Doering et al., 2018)	-	-	-	53.1	-	-	50.4	63.3	0.2
Xiao et al. (2018), ResNet50	**76.4**	**77.2**	**72.2**	**65.1**	**68.5**	**66.9**	**60.3**	**70.0**	9.1
STAF-MS(Raaj et al., 2019)	-	-	-	62.8	-	-	59.5	69.4	7
Our method	65.5	75.9	68.1	58.9	63.1	59.0	52.1	63.4	**42.2**
PoseTrack 2018 Validation									
Xiao et al. (2018), ResNet50	**74.4**	76.9	72.2	**65.2**	69.2	**70.0**	**62.9**	**70.4**	9.1
STAF-SS(Raaj et al., 2019)	-	-	-	56.2	-	-	54.2	63.7	27
Our method	66.2	**81.9**	**74.3**	62.8	**70.1**	66.2	57.5	68.3	**42.2**

Table 5: Comparisons of properties of different models. FPS is tested on a single NVIDIA Tesla P40. MaskRCNN(He et al., 2017) correspond to the configuration of ResNet-50 with feature pyramid network. Our method is 10 times faster than MaskRCNN(He et al., 2017), and 2 times quicker than PAF(Cao et al., 2017).

Type	Method	Model Size (MB)	# Parameter	FLOPs	AP	FPS
Top-down	MaskRCNN(He et al., 2017)	480.8	62.4×10^6	536.6×10^9	**0.627**	4.4
Bottom-up	PAF(Cao et al., 2017)	209.3	52.3×10^6	159.6×10^9	0.584	20.4
Bottom-up	Our method	**85.2**	$\mathbf{21.2\times10^6}$	$\mathbf{82.5\times10^9}$	0.584	**42.2**

the training speed and reduces the requirements of the processor's memory, frequency, etc.

4.6. Detailed Analysis

In this section, we carry out ablation experiments to validate the design of the network architecture. These include determining the position of extracted features for transfer, testing with/without the feature transfer sub-networks to determine their effect on performance, testing with/without the refinement module to determine its effect on performance, and evaluating the effect of multi-scale evaluation. Finally, we show the re-

Table 6: Comparisons of multi-scale evaluation on the COCO 2018 test-dev set

Method	**AP**	$\text{AP}^{0.5}$	$\text{AP}^{0.75}$	AP^{M}	AP^{L}
PAF(Cao et al., 2017), single-scale	0.469	0.737	0.493	0.403	0.561
PAF(Cao et al., 2017), multi-scale	0.584	0.815	0.626	0.544	0.651
Our method, single-scale	0.483	0.751	0.503	0.462	0.515
Our method, multi-scale	0.584	0.821	0.626	0.537	0.658

Table 7: The comparisons of transfer from different layers on the COCO 2018 validation set

Method	**AP**	$\text{AP}^{0.5}$	$\text{AP}^{0.75}$	AP^{M}	AP^{L}
Transfer from Conv-1	0.452	0.740	0.456	0.387	0.549
Transfer from Conv-2	0.463	0.741	0.475	0.401	0.557
Transfer from Conv-3	0.475	0.741	0.493	0.415	0.565
Transfer from Conv-5	0.459	0.742	0.454	0.433	0.501
Transfer from Conv-4	0.484	0.741	0.512	0.429	0.573

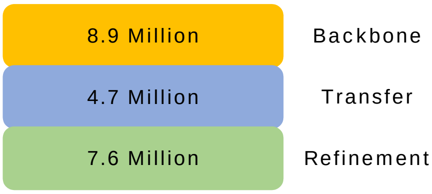


Fig. 5: The number of parameters for each module of our network.

Table 8: The comparisons of with and without transfer sub-networks on the COCO 2018 validation set

Method	**AP**	$\text{AP}^{0.5}$	$\text{AP}^{0.75}$	AP^{M}	AP^{L}
Ours no transfer	0.444	0.737	0.440	0.391	0.530
Ours	0.484	0.741	0.512	0.429	0.573

Table 9: The comparisons of with and without refinement module on the COCO 2018 validation set

Method	**AP**	$\text{AP}^{0.5}$	$\text{AP}^{0.75}$	AP^{M}	AP^{L}
Ours no refine	0.467	0.751	0.453	0.400	0.576
Ours	0.484	0.741	0.512	0.429	0.573

sults of error analysis to suggest future modification directions. Since the COCO dataset provides standard validation set and performance analysis tools, we undertake all ablation experiments on the COCO 2018 validation set using single scale input.

4.6.1. *The Effect of Features Extraction Layer Placement*

We compare the effect of extracting features from different layers to the accuracy by quantitative evaluations. Table 7 presents the results of extracting features from Conv-1 to Conv-5. We can see that the accuracy (AP metric) is increased progressively by extracting features from higher layers until Conv-5. Extracting features from the Conv-5 layer has the effect of lowering accuracy, suggesting that this layer is not as suitable as an output layer for feature transfer.

4.6.2. *The Effect of Feature Transfer Sub-network*

Here, we determine the effect that the feature transfer sub-networks have on performance. We do this by removing the relevant feature transfer layers (i.e. the blue blocks in Fig. 2) while keeping all other structures and parameters the same for the equality of experiments. The results of the experiments are presented in Table 8. We find that the feature transfer sub-network accounts for up to 4.0% of improvement in estimation accuracy (AP metric). This result indicates that feature transfer has a significant effect on network performance.

4.6.3. *The Effect of Refinement Module*

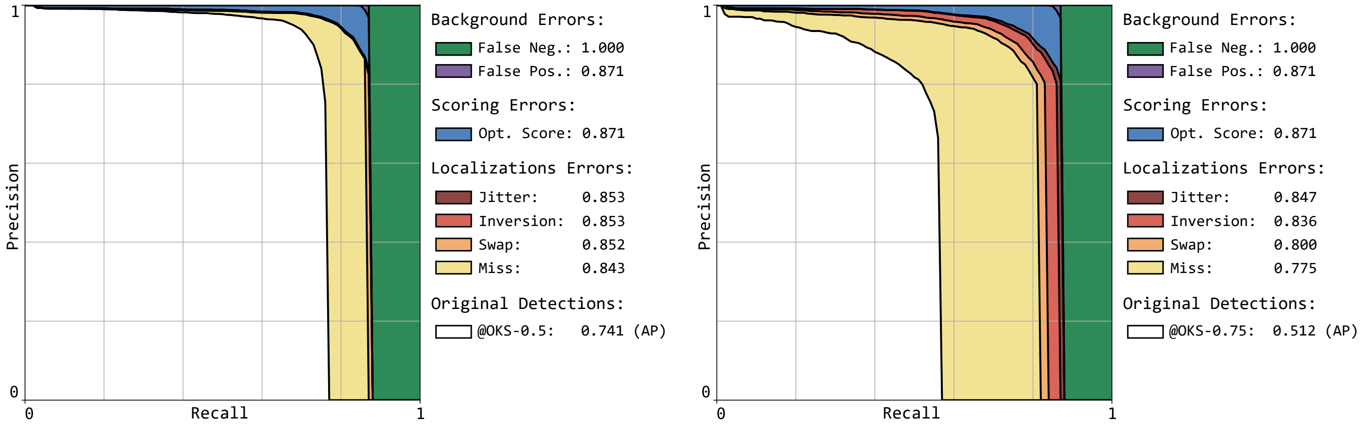
We determine the effect that the refinement module has on network performance. We do this by removing the entire refinement module (i.e. the blocks surrounded by the dotted blue line in Fig. 2) and set the nodes of 1 and 2 (Fig. 2) as the outputs of joint and limb branches. In Table 9, we can see that the refinement module contributes a 1.7% improvement in estimation accuracy (AP metric). This value reveals that refining the score maps with context information is an effective strategy for improving accuracy.

4.6.4. *The Effect of Multi-scale Evaluation*

In order to analyse the effect of multi-scale evaluation, we report the results of single-scale and multi-scale evaluation on the PAF(Cao et al., 2017) and our method on the COCO test-dev set in Table 6. We observe that our single-scale model outperforms the single-scale PAF model by about 1.4% in accuracy. We also notice that both single-scale models using the $\text{AP}^{0.5}$ metric already achieve a high accuracy of around 0.75. Multi-scale evaluation mainly compensates for the precision at extremely strict OKS matching thresholds from $\text{AP}^{0.75}$ to AP^{95}.

4.6.5. *Performance Analysis on COCO 2018 Validation Set*

We use the evaluation tools of Ronchi and Perona (2017) to analyse the error constitutions of our model. Ronchi and Perona (2017) define 3 error types including background error, scoring error, and localisation error. *Background error* includes false positives (FP) and false negatives (FN). *Scoring error* occurs when one prediction with a high confidence score has a low OKS value. *Localisation error* contains four specific types of



(a) OKS=0.5

(b) OKS=0.75

Fig. 6: Error distribution and sensitivity analysis. The plot on the left shows the effect of progressively rectifying errors of each type on the accuracy of our method at the OKS evaluation threshold of 0.5. The legend indicates the corresponding AP values. The plot on the right shows the results using the same sensitivity analysis method at the OKS threshold of 0.75.

error - jitter, inversion, swap and miss. *Jitter* is when the predictions have a small error around the correct keypoint location. *Inversion* is defined as the errors of inversions between the left and right parts of the body. *Swap* denotes the predictions with the same part type on incorrect instances. This metric is useful for overlapping people. *Miss* is used when the predictions have large localisation errors which exceed the defined keypoint similarity thresholds.

The impact of all types of error above on the accuracy of our approach is summarised in Fig. 6 where the OKS threshold is at 0.5 and 0.75. Each plot consists of a set of Precision-Recall (PR) curves where each curve is strictly larger than the previous as the method's errors are progressively rectified. For the OKS threshold of 0.5 (Fig. 6(a)), we can see that the overall AP is 0.741. Rectifying all the *miss* errors obtains a large improvement of the AP to 0.843. Correcting *swap*, *inversion*, and *jitter* have almost no change to the AP (0.853). When localisation is correct, revising the *confidence score* can contribute a small AP improvement of about 1.8% (0.871). With the optimal confidence score, correcting *background false positives* has a trivial effect on the AP as predictions barely remain unmatched. Finally, eliminating *background false negatives* results in perfect performance. In contrast, using a more strict OKS threshold of 0.75 enlarges the impact of *localisation error* but has no effect on *scoring error* and *background error*, as shown in Fig. 6(b). From here we know that the errors of our method are dominated mostly by *localisation error* and *background false negatives*.

Finally, we give some qualitative results in Fig. 7, which includes the cases of scale, appearance and viewpoint variation, occlusion and crowding.

5. Deployment Acceleration

The existing deep learning frameworks provide several basic layers or operators (OP) to support specific computations due to the consideration of flexibility during model designing and training. However, if the network structure and weights are fixed, some layers or operators could be merged as one operation and model forward inference can be further accelerated. For example, one convolutional layer followed by one bias layer have to deploy memory operation twice. Actually, these two layers could be merged into one layer according to the mathematical derivation. In addition, some layers could be ignored in the implementation, such as concatenation operation as multiple corresponding tensors can be utilised directly by the next layer. Currently, there are some open source libraries providing network optimisation and acceleration. Here, we prototype the acceleration solution using TensorRT$^{\text{TM}}$ (NVIDIA, 2019), which is the state-of-the-art library that facilitates optimisation on NVIDIA GPUs. The optimised inference implementation does not affect the prediction accuracy since the network weights and parameter precision are not changed. The acceleration results are shown in Table 10. The inference time of the network is decreased from 22.71 ms to 12.45 ms on one NVIDIA Tesla P40 GPU. The last pose estimation speed achieves 73.8 FPS.

Table 10: The comparisons of network's inference speed before and after using TensorRT$^{\text{TM}}$ library

Optimisation	Before	After
Inference time (ms)	22.71	12.45

6. Conclusion

In this work, we have proposed a deep feature transfer network that captures concurrently activated joint and limb features to form a complementary inference architecture for multi-person pose estimation. Experiments are performed on the three most popular multi-person pose estimation benchmarks. Results show that the proposed structure effectively improves the accuracy. In addition, our method achieved comparable state-of-the-art accuracy with speeds exceeding 42.2 FPS, which is between 2 and 10 times faster than existing works. We further

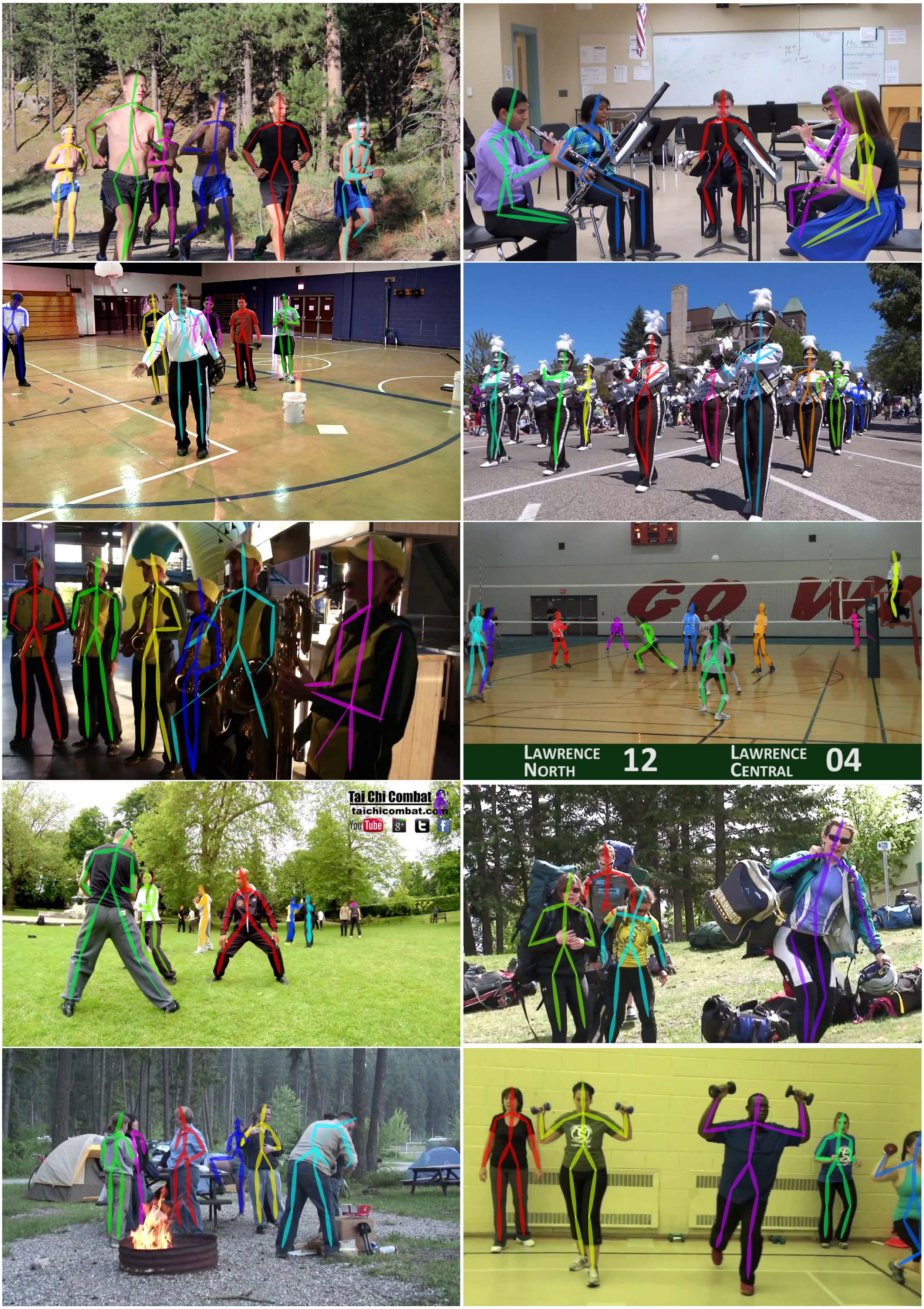


Fig. 7: Qualitative results of our method on the MPII dataset. Each color corresponds to a human instance.

accelerate the inference speed to 73.8 FPS by using the deep learning optimisation library of TensorRT.

As a future direction, we would like to further improve the method by combining other strategies from existing literature, such as the feature pyramid by Yang et al. (2017) and the attention mechanism by Chu et al. (2016), to reduce localisation and background false negative errors. Various network architectures for structural relation inference tasks in existing works Chen et al. (2018) could also provide guidance to improve body part detection. In addition, using light-weight and low-bit network to obtain high speed while maintaining the accuracy is a promising future research direction.

Acknowledgement

This work was supported in part by the Erasmus Mundus Action 2 Programme (Ref: 2014-0861/001-001), the Royal Society (Ref: IES\R2\181024 and IES\R1\191147). We would like to thank Kevin Mccay for polishing the paper.

References

Alp Güler, R., Neverova, N., Kokkinos, I., 2018. Densepose: Dense human pose estimation in the wild, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 7297–7306.

Andriluka, M., Iqbal, U., Milan, A., Insafutdinov, E., Pishchulin, L., Gall, J., Schiele, B., 2018a. Posetrack: A benchmark for human pose estimation and tracking, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 5167–5176.

Andriluka, M., Pishchulin, L., Gehler, P., Schiele, B., 2014. 2d human pose estimation: New benchmark and state of the art analysis, in: Proceedings of the IEEE Conference on computer Vision and Pattern Recognition (CVPR), pp. 3686–3693.

Andriluka, M., Pishchulin, L., Gehler, P., Schiele, B., 2018b. Mpii human pose database. URL: `http://human-pose.mpi-inf.mpg.de/#results`.

Andriluka, M., Roth, S., Schiele, B., 2009. Pictorial structures revisited: People detection and articulated pose estimation, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1014–1021.

Belagiannis, V., Zisserman, A., 2017. Recurrent human pose estimation, in: 2017 12th IEEE International Conference on Automatic Face Gesture Recognition (FG), pp. 468–475.

Cao, Z., Simon, T., Wei, S.E., Sheikh, Y., 2017. Realtime multi-person 2d pose estimation using part affinity fields, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 7291–7299.

Caruana, R., 1997. Multitask learning. Machine learning 28, 41–75.

Chang, J.Y., Lee, K.M., 2018. 2d 3d pose consistency-based conditional random fields for 3d human pose estimation. Computer Vision and Image Understanding (CVIU) 169, 52 – 61. URL: `http://www.sciencedirect.com/science/article/pii/S107731421830016X`, doi:`https://doi.org/10.1016/j.cviu.2018.02.004`.

Chen, X., Li, L.J., Fei-Fei, L., Gupta, A., 2018. Iterative visual reasoning beyond convolutions, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 7239–7248.

Chu, X., Ouyang, W., Li, H., Wang, X., 2016. Structured feature learning for pose estimation, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 4715–4723.

Chu, X., Yang, W., Ouyang, W., Ma, C., Yuille, A.L., Wang, X., 2017. Multi-context attention for human pose estimation, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1831–1840.

Ciliberto, C., Mroueh, Y., Poggio, T., Rosasco, L., 2015. Convex learning of multiple tasks and their structure, in: Proceedings of the 32nd International Conference on Machine Learning (ICML), pp. 1548–1557.

Dai, J., He, K., Sun, J., 2016. Instance-aware semantic segmentation via multi-task network cascades, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3150–3158.

Doering, A., Iqbal, U., Gall, J., 2018. Jointflow: Temporal flow fields for multi person pose estimation, in: Proceedings of the British Machine Vision Conference (BMVC), p. 261.

Fang, H.S., Xie, S., Tai, Y.W., Lu, C., 2017. Rmpe: Regional multi-person pose estimation, in: Proceedings of the IEEE International Conference on Computer Vision (ICCV), pp. 2334–2343.

Girdhar, R., Gkioxari, G., Torresani, L., Paluri, M., Tran, D., 2018. Detect-and-track: Efficient pose estimation in videos, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 350–359.

Gkioxari, G., Arbelaez, P., Bourdev, L., Malik, J., 2013. Articulated pose estimation using discriminative armlet classifiers, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3342–3349.

Goodfellow, I., Bengio, Y., Courville, A., Bengio, Y., 2016a. Deep learning. volume 1. MIT press Cambridge.

Goodfellow, I., Bengio, Y., Courville, A., Bengio, Y., 2016b. Deep learning. volume 1. MIT press Cambridge.

He, K., Gkioxari, G., Dollár, P., Girshick, R., 2017. Mask r-cnn, in: Proceedings of the IEEE International Conference on Computer Vision (ICCV), pp. 2980–2988.

He, K., Zhang, X., Ren, S., Sun, J., 2016. Deep residual learning for image recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 770–778.

Hong, C., Chen, X., Wang, X., Tang, C., 2016. Hypergraph regularized autoencoder for image-based 3d human pose recovery. Signal Processing 124, 132–140.

Hong, C., Yu, J., Tao, D., Wang, M., 2014. Image-based three-dimensional human pose recovery by multiview locality-sensitive sparse retrieval. IEEE Transactions on Industrial Electronics 62, 3742–3751.

Hong, C., Yu, J., Wan, J., Tao, D., Wang, M., 2015. Multimodal deep autoencoder for human pose recovery. IEEE Transactions on Image Processing (TIP) 24, 5659–5670.

Insafutdinov, E., Pishchulin, L., Andres, B., Andriluka, M., Schiele, B., 2016. Deepercut: A deeper, stronger, and faster multi-person pose estimation model, in: Proceedings of the European Conference on Computer Vision (ECCV), pp. 34–50.

Iqbal, U., Gall, J., 2016. Multi-person pose estimation with local joint-to-person associations, in: Proceedings of the European Conference on Computer Vision (ECCV), pp. 627–642.

Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., Guadarrama, S., Darrell, T., 2014. Caffe: Convolutional architecture for fast feature embedding. arXiv preprint arXiv:1408.5093 .

Jin, S., Ma, X., Han, Z., Wu, Y., Yang, W., Liu, W., Qian, C., Ouyang, W., 2017. Towards multi-person pose tracking: Bottom-up and top-down methods, in: Proceedings of the IEEE International Conference on Computer Vision (ICCV) PoseTrack Workshop, p. 7.

Kawana, Y., Ukita, N., Huang, J.B., Yang, M.H., 2018. Ensemble convolutional neural networks for pose estimation. Computer Vision and Image Understanding (CVIU) 169, 62 – 74. URL: `http://www.sciencedirect.com/science/article/pii/S1077314217302308`, doi:`https://doi.org/10.1016/j.cviu.2017.12.005`.

Kim, S., Xing, E.P., 2010. Tree-guided group lasso for multi-task regression with structured sparsity, in: Proceedings of the 27th International Conference on Machine Learning (ICML), p. 1.

Levinkov, E., Uhrig, J., Tang, S., Omran, M., Insafutdinov, E., Kirillov, A., Rother, C., Brox, T., Schiele, B., Andres, B., 2017. Joint graph decomposition & node labeling: Problem, algorithms, applications, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 6012–6020.

Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L., 2014. Microsoft coco: Common objects in context, in: Proceedings of the European conference on computer vision (ECCV), pp. 740–755.

Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.Y., Berg, A.C., 2016. Ssd: Single shot multibox detector, in: Proceedings of the European conference on computer vision (ECCV), pp. 21–37.

Long, J., Shelhamer, E., Darrell, T., 2015. Fully convolutional networks for semantic segmentation, in: Proceedings of the IEEE Conference on computer Vision and Pattern Recognition (CVPR), pp. 3431–3440.

Nair, V., Hinton, G.E., 2010. Rectified linear units improve restricted boltzmann machines, in: Proceedings of the 27th International Conference on

Machine Learning (ICML), pp. 807–814.

Newell, A., Huang, Z., Deng, J., 2017. Associative embedding: End-to-end learning for joint detection and grouping, in: Proceedings of the Neural Information Processing Systems (NIPS), pp. 2277–2287.

Newell, A., Yang, K., Deng, J., 2016. Stacked hourglass networks for human pose estimation, in: Proceedings of the European Conference on Computer Vision (ECCV), pp. 483–499.

NVIDIA, 2019. Tensorrt. URL: `https://developer.nvidia.com/tensorrt`.

Papandreou, G., Zhu, T., Kanazawa, N., Toshev, A., Tompson, J., Bregler, C., Murphy, K., 2017. Towards accurate multi-person pose estimation in the wild, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR).

Pishchulin, L., Andriluka, M., Gehler, P., Schiele, B., 2013. Poselet conditioned pictorial structures, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 588–595.

Pishchulin, L., Insafutdinov, E., Tang, S., Andres, B., Andriluka, M., Gehler, P.V., Schiele, B., 2016. Deepcut: Joint subset partition and labeling for multi person pose estimation, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 4929–4937.

Raaj, Y., Idrees, H., Hidalgo, G., Sheikh, Y., 2019. Efficient online multi-person 2d pose tracking with recurrent spatio-temporal affinity fields, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 4620–4628.

Ronchi, M.R., Perona, P., 2017. Benchmarking and error diagnosis in multi-instance pose estimation, in: Proceedings of the IEEE International Conference on Computer Vision (ICCV), pp. 369–378.

Sapp, B., Taskar, B., 2013. Modec: Multimodal decomposable models for human pose estimation, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3674–3681.

Simonyan, K., Zisserman, A., 2015. Very deep convolutional networks for large-scale image recognition, in: Proceedings of the International Conference on Learning Representations (ICLR).

Sun, K., Zhao, Y., Jiang, B., Cheng, T., Xiao, B., Liu, D., Mu, Y., Wang, X., Liu, W., Wang, J., 2019. High-resolution representations for labeling pixels and regions, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR).

Tome, D., Russell, C., Agapito, L., 2017. Lifting from the deep: Convolutional 3d pose estimation from a single image, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2500–2509.

Tompson, J.J., Jain, A., LeCun, Y., Bregler, C., 2014. Joint training of a convolutional network and a graphical model for human pose estimation, in: Proceedings of the Neural Information Processing Systems (NIPS), pp. 1799–1807.

Toshev, A., Szegedy, C., 2014. Deeppose: Human pose estimation via deep neural networks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1653–1660.

Varadarajan, S., Datta, P., Tickoo, O., 2018. 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 (WACV), pp. 418–428.

Wei, S.E., Ramakrishna, V., Kanade, T., Sheikh, Y., 2016. Convolutional pose machines, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 4724–4732.

Xiao, B., Wu, H., Wei, Y., 2018. Simple baselines for human pose estimation and tracking, in: Proceedings of the European Conference on Computer Vision (ECCV), pp. 466–481.

Xiu, Y., Li, J., Wang, H., Fang, Y., Lu, C., 2018. Pose flow: Efficient online pose tracking, in: Proceedings of the British Machine Vision Conference (BMVC), pp. 1–12.

Yang, W., Li, S., Ouyang, W., Li, H., Wang, X., 2017. Learning feature pyramids for human pose estimation, in: Proceedings of the IEEE International Conference on Computer Vision (ICCV).

Yang, W., Ouyang, W., Li, H., Wang, X., 2016. End-to-end learning of deformable mixture of parts and deep convolutional neural networks for human pose estimation, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3073–3082.

Yang, Y., Ramanan, D., 2011. Articulated pose estimation with flexible mixtures-of-parts, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1385–1392.