Ph.D. DISSERTATION

2D Human Pose Estimation and Tracking with spatial and temporal features

2D 영상에서 시간 특징과 지역적 특징의 분석을 통한 사람 포즈 검출 및 추적

BY

JIHYE HWANG

FEBRUARY 2021

Intelligent Systems
Department of Transdisciplinary Studies
Graduate School of Convergence Science and Technology
SEOUL NATIONAL UNIVERSITY

Ph.D. DISSERTATION

2D Human Pose Estimation and Tracking with spatial and temporal features

2D 영상에서 시간 특징과 지역적 특징의 분석을 통한 사람 포즈 검출 및 추적

BY

JIHYE HWANG

FEBRUARY 2021

Intelligent Systems
Department of Transdisciplinary Studies
Graduate School of Convergence Science and Technology
SEOUL NATIONAL UNIVERSITY

2D Human Pose Estimation and Tracking with spatial and temporal features

2D 영상에서 시간 특징과 지역적 특징의 분석을 통한 사람 포즈 검출 및 추적

지도교수 곽노준

이 논문을 공학박사 학위논문으로 제출함

2021년 2월

서울대학교 대학원

융합과학부 지능형융합시스템전공

황지혜

황지혜의 공학박사 학위 논문을 인준함

2021년 2월

위원장: 이교구 (인)

부위원장: 곽노준 (인)

위원: 박재흥 (인)

위원: 최상일 (인)

위원: 이민식 (인)

Abstract

2D human pose estimation and tracking aim to detect the location of a person's parts and their trajectory. A pose is composed of parts of a person, and a person's part is an element of the body such as arms, legs and head. Pose estimation technique is being utilized both industrially and academically. For example, in a home training system, pose detection can detect the user's pose and help the user correct the posture. Also, in human action recognition research, human pose information can be exploited as a helpful supplementary information.

In order to apply human pose studies to real-world systems, the model is required to be of high performance and also light enough to run in a real-time manner. In this paper, we have focused on improving accuracy. We have considered how to utilize the feature values to achieve high accuracy using the spatial and temporal features.

Spatial feature means characteristic values such as textures, patterns, and postures that can be extracted from images. We have made better use of the spatial feature by dividing it into local and global features. The global feature is likely to include a large number of parts, while the local feature focuses on a relatively small number of parts.

First, we have proposed a structure that can use the global-local feature at the same time to improve the performance. The global network intensively learns the global feature, and the local network can learn various regional information from images. The local network performs as a function of refining the pose detected in the global network sequentially. To prove the efficiency of the proposed method, experiments have been conducted on the Leeds sports dataset

(LSP) data, which is one of the single-person pose estimation datasets.

Secondly, we define the rare pose using global feature and solve the imbalance in poses. First of all, the poses are classified using location information of the entire pose. Experiments have shown that the poses are distributed around certain poses (standing poses, upper body poses, etc.), and an imbalance between them apparently exists. We have proposed methods such as weighted loss, synthesizing rare pose data, etc. to resolve the imbalance. Experiments are conducted using MPII and COCO data, which are widely used in multi-person pose estimation.

The temporal feature refers to the varying information of poses along the time. It is usually recommended to use time information when analyzing objects in a video. Therefore thirdly, we have estimated and tracked the poses with a map that expresses the change of a person's movement. The network learns the spatial and temporal maps together to create synergy between each other. The experiment has been conducted in multi-person pose tracking data, Posetrack 2017 and 2018.

Even if the proposed three methods improve different issues, utilized together. For example, a new structure is a top-down approach and has parallel two deconvolutions for spatial (Heatmap) and temporal map (TML). Additionally, the rare pose data augmentation and the local network are applied to increase performance. Thus, adopting three methods is available to improve performance and more extensible in the pose estimation field.

keywords: 2D human pose estimation, 2D multi-person pose tracking, Spatial feature, Temporal feature, Rare pose, Temporal flow map, Deep learning, clustering

student number: 2015-31349

Contents

List of Tables

List of Figures

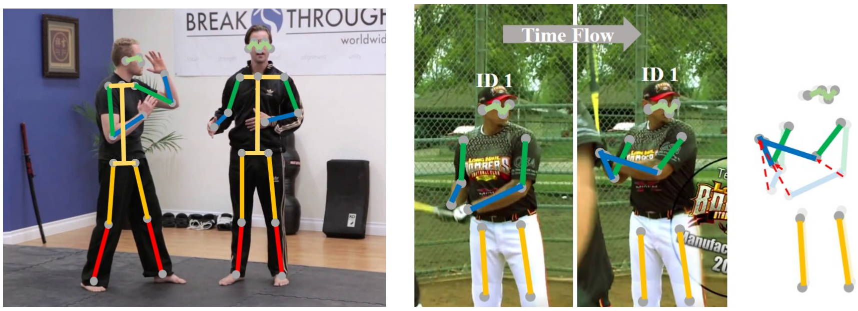
Chapter 1

Introduction

The 2D human pose estimation is to detect the pose of human in 2D images. The pose is similar to a person's skeleton and is composed of parts such as head, hand as shown in Figure 1.1. Although the number of parts constituting a pose is different in each pose dataset, parts are distributed mainly on the head, arms, and legs. The location of each part is represented as (x, y) coordinate in the image.

Because the pose indicates a person's shape and motion, pose information can be applied to various fields such as education and entertainment. Recently a home training system has been released by a well known company to help those who workout alone at home. When the user follows the instructor's motion shown in the screen, the home training system detects the user's pose and suggests correction on the posture. Also, other research fields have used pose feature as an additional information. For example, in action recognition area, action recognition accuracy has been enhanced drastically using human pose information [51, 75, 52].

In order to apply pose information to various fields, the essential compo-



(a)

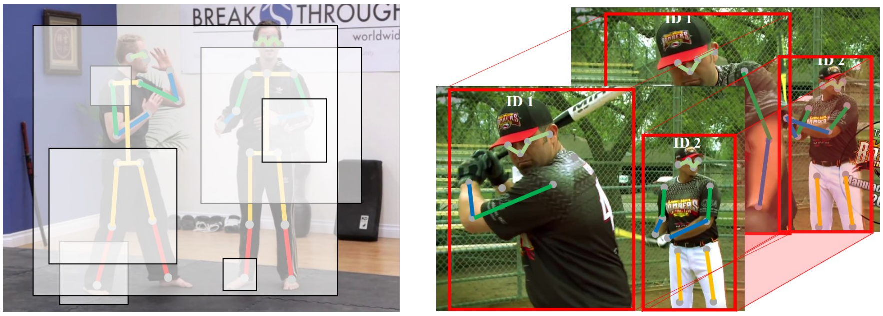
(b)

Figure 1.1: The example results of pose estimation and pose tracking. (a) shows the multi-person pose estimation result and (b) shows the multi-person pose estimation and tracking result. Left side of (b) is the pose overlapping image between two images. It shows the movement (red arrow) of the pose between the poses.

nents of the model are high performance and low complexity. With a high performance, a smaller number of poses will fail to be detected. With a low complexity, pose estimation will be done light enough to reduce the runtime and memory consumption in an actual application. Among them, we have focused on improving the accuracy of a pose estimation and tracking model.

There are various factors such as the model structure, the size of an image and the optimization method to improve the performance. Among them, we have considered how to use and combine the feature values from the model itself to increase pose accuracy. By designing the architecture of a pose estimation model, we induce the intermediate features to represent spatial and temporal features to harvest more information from an images or video.

The research on 2D human pose consists of pose estimation, dense pose estimation and pose tracking. This paper is mainly about pose estimation and



(a) Spatial feature

(b) Temporal feature

Figure 1.2: The visualization of spatial and temporal features.
(a) is the spatial features and (b) is the temporal features. In the (a), the area containing a person's whole means a global feature and an area including a partial part means a local feature.

tracking. The Figure 1.1 shows the example of (a) pose estimation and (b) pose tracking. The pose estimation is a study that detects the (x, y) coordinates of a person's part in the image. Depending on how many people are in the input image, it can be divided into a single-person pose estimation and a multi-person pose estimation. The pose tracking is a study of detecting and tracking poses. Tracking a pose, the model should assign every detected parts of a person to the identically corresponding parts of the same person in the next frame. More specifically, the pose tracking aims to give each person a unique id and correctly match each detected pose.

1.1 Spatial feature

Implicit connections between parts exist in the pose of a person. If we focus only on a single part, we cannot learn the overall relationship among parts, and the performance may degrade. Besides, if we focus on the entire pose, the accuracy of detecting each part having a wider range of movement may be reduced. Therefore, we need to use all the features globally and locally.

Literally, the spatial feature is a feature value that represents the space domain. We have divided the spatial feature into global features and local feature. The global feature stands for the overall property of the pose, and local feature embraces partially focused information. As shown in Figure 1.2 (a), the global feature is likely to extract information from area that contain a large number of parts while the local feature would rather interpret a smaller area from an image. Therefore, in order to detect poses effectively, both global and local features should be used appropriately.

1.1.1 Global-local network

Many studies have learned global and local information using the receptive field of traditional convolutional neural networks (CNN). The receptive field refers to a region where one neuron affects the pre-layer, and it usually lies on a square-shaped area in the convolution layer. Some researchers expand the receptive field by repeatedly stacking networks or accumulating layers deeply. In the expanded receptive field, the global feature can be extracted to includes the full body. Also, the receptive field is small at the bottom layer and it can be used as the local feature that contains a few parts.

Strided convolutional operation restricts the area of corresponding receptive field and offers a limited amount of information from the local feature of interest. To solve this problem, we have proposed a local network to get a better representation of local feature.

Proposed method consists of two parts: the global (general) network and the local (refine) network. In the global network module, we predict Heatmaps [7] of parts using the ResNet-101.

The feature maps of the final convolutional layer are concatenated with the Heatmaps of global network, which are inputted to the local network to refine the location. In the local network, position-sensitive score maps are created to explain the spatial information on region of interest (ROI) as in [43], where region-based fully convolutional networks was proposed for object detection. More details are in Chapter 3.

1.1.2 Exploring rare pose estimation using global pose information

Comparing to datasets of other research field such as object detection, the number of training images in pose estimation datasets are relatively smaller. Among them, rarely seen poses which act as outliers can be observed and this easily affects the performance of the model. To resolve this problem, we need to analyze the distribution of the dataset.

For a better understanding on poses, a criteria to classify them is required. Unfortunately, a pose is expressed as a combination of continuous values with a very wide range and this makes the criteria hard to define. Thus, we propose a method to classify poses using the global pose feature.

We use the (x, y) coordinates of parts as global information. All poses are grouped by clustering with global information that consist of part's location. As a result, poses are clustered around standard poses such as standing pose, left, right pose, etc. A large number of poses are close to the center pose and a small number of poses still remains far from the center. This infers that the diversity of poses in the data is insufficient and statistics of poses is imbalanced.

We have defined the imbalance part as ”rare pose” where the distance to the center position is higher than the threshold. A pose near the outlier, its predicted accuracy is likely to decrease. The poses corresponding to the outlier are relatively small in the dataset and, at the same time, difficult to detect. Thus, we define the rare pose and propose methods to improve the performance of rare poses. More details are available in Chapter 4

1.2 Temporal feature

The temporal feature is time-sequential information and feature value that can be extracted from the video. To acquire temporal feature, we can think of a relatively narrow temporal feature acquired from the neighboring two frames and a wide range of temporal feature obtained from the ongoing video. The temporal feature is a rich information that can be extracted from the video data and is essential to analyze the video data.

We focus on increasing the pose estimation and tracking performance in the video. We should make a proper use of the temporal and spatial information to track the poses effectively. We considered how we could simultaneously learn the temporal and spatial features. We thought that if both feature were learned together, it would create synergy. Thus, we propose a temporal flow map for limb (TML) representing the limb's movement and a network designed parallel to train both features.

Limb denotes the area that connects two parts. A pose is composed of parts and a part is a (x, y) coordinate. Tracking a single part of a pose equals to tracking a single coordinate. However, tracking only one single part may not contain enough temporal information due to lack of representation power and it may be vulnerable to occlusion of parts. Therefore, instead of using a single part point, a limb connecting two adjacent parts is tracked, which is expected to resolve the above mentioned problems.

The TML is designed to represent a temporal movement of a person by estimating the direction of limbs' movement. More specifically, we subdivide each limb into several sections equally in each frame. Then, 2D unit vectors that represent the direction of corresponding limb sections between two frames are calculated, which are used to build each limb's temporal maps. A huge amount of data is needed to train the TML because the maps have to learn extensive information. Thus, we develop a multi-stride method as a data augmentation method to learn various types of TML. In other words, we randomly take the two frames within a given time range. More detailed in Chapter 5

The proposed methods have solved different issues of human pose. Going one step forward, other issues can be solved by using combined methods. For example, a new pose estimator and tracker followed top-down manner is constructed using the combined methods to get higher accuracy. The top-down manner estimates the pose based on the detected bounding box. So, the accuracy of pose estimation is higher than a bottom-up manner and state-of-the-art methods follow the top-down manner.

Additionally, the Heatmap representing body parts can be learned using various local features by the proposed global-local network. And, the rare pose data augmentation helps to general learning. Above the three ways which are the top-down manner, the global-local network and the rare pose data augmentation help to improve a performance of pose estimation. For pose tracking, the temporal flow map for limbs (TML) is adopted. More detailed in Chapter 6. Moreover, if a method of reducing parameters is additionally applied, it will be possible to detect and track the pose in real time.

Chapter 2

Related work

2.1 Single-person pose estimation

Single-person pose estimation is the pose estimating problem of one person from an image with only one person. The single-person pose estimation problem has been researched for a long time in Computer Vision (CV). As Deep learning advances, the performance of pose estimation has increased significantly. [73, 11, 35, 79, 4, 57, 18, 23]

DeepPose [73] is the first paper to adopt Deep learning to human pose estimation. A network of DeepPose has consisted of 7 fully connected layers. The last layer of the network regresses the (x, y) coordinates of joints. The same network structure was applied to multiple stages to improve the position. DeepPose has been shown that Deep learning is effectively applied in pose estimation problems, with higher performance than conventional methods that do not use deep learning.

One of the reasons why pose estimation is difficult is that we need to detect the (x, y) coordinates. The range of coordinates is too broad because of the variety pose variation. Thus, a Heatmap to express the location of parts was proposed. The Heatmap is created as a Gaussian map centered on the (x, y) coordinate. The Heatmaps are created for each part, and all Heatmaps of parts are concatenated to a 2D tensor form.

[35] is the first paper to train the Convolution neural network (CNN) with the Heatmap. They have two significant parts as the part detector and the high-level spatial model. Part detecter trains the Heatmap through parallel convolution layers with different input image size. Based on the Part detector's output, the Spatial model trains the Heatmap using parts' relation. At the last time, they have merged the outputs of each part. After their work, most papers have adopted the Heatmap to regress the joints.

In order to solve the single-person pose estimation problem, many papers have built up networks deeply. The deeper network has a larger receptive field, and it is more efficient to detect global pose information.

[79] applied the multi-stage method to take a deep network. The multi-stage method is constructed with small networks of the same structure repeatedly. In each stage, they calculate a loss of Heatmap, and the calculated Heatmap has added the input of the next stage. Not only get the global information more broadly, but it also has the advantage of being able to view the attention part via the Heatmap.

Stacked-hourglass network [57] also applied the multi-stage method using a proposed Hourglass module. The Hourglass module has a symmetric structure between down-sampling and up-sampling. Repeatedly features down-sampling and up-sampling, the network learns global and local features.

Because parts' movements are diverse, it is also essential to use appropriate local features expressing partial characteristics. In the Convolutional neural network (CNN), a receptive field means a volume of input values that affect the output neuron. The feature of the receptive field explains local information. The deeper network has a larger receptive field and various sizes of receptive fields. Unfortunately, it is not easy to get the various location of the receptive field.

[18] proposed the Dual-source deep Convolutional neural networks (DS-CNN). They learn the pose to integrate both the local appearance and full body appearance. They get the local information through the part image patch, which has various scales and locations. Because the image patch is detected independently, it is possible to learn various local information better than the receptive field.

To efficiently use global and local information simultaneously, we proposed the global-local network. The global-local network concatenates with the baseline network to learn global features and the small network to learn local feature. Differently DS-CNN, we used the local information to refine the pose. Because we extract local information on the region of interest (ROI), the local information is an appropriate refining pose. More detail, in chapter 3.

2.2 Multi-person pose estimation

The multi-person pose estimation is to estimate poses of the multi-person in the image. The number of multi-person is different for each image. Multi-person pose estimation methods are divided into Bottom-up approach and Top-down approach. The most significant difference between the two methods is whether or not to have a person detection module. The top-down approach firstly detects a person in the image through the detection module and extracts that person's pose on the detected bounding box. The Bottom-up approach, on the other hand, detects a person's pose directly within the image. [53, 42, 55, 19, 61, 85, 39, 86, 87, 72, 37, 12, 62, 15, 14, 59]

Both approaches have pros and cons. The top-down approach has higher performance than the bottom-up approach method because it prioritizes human detection. So the state-of-the-art method in multi-person pose estimation is mostly top-down structures. In contrast, the bottom-up approach has a relatively faster inference speed than the top-down because there is no detection module part. So in real-life applications, the bottom-up method is more suitable.

In the bottom-up approach, detecting the pose is split mainly into detecting the part's location and tying the parts of the same person. To detect the part's location, previous research proposed a network to learn the Heatmap to represent parts or methods to directly detect the part. Also, they suggested ways to tie up parts of the same person to enhance pose detection performance.

Openpose[10] proposed a part affinity field (PAF) map to connect detected parts. PAF is the map that indicates the direction of the limb connecting parts. At the inference, the parts are detected through Heatmap. They calculate the association score between detected parts through the line segmentation method in the PAF and the distance between parts. Finally, match them together into one person's parts.

[63] proposed the DeepCut module to connect parts of the same person. [63] detects various parts candidates first through the part detector. Each part has a unary score and uses this score to calculate the association cost again. Based on the association score, they match part candidates. DeepCut[63] and DeepCut[29] have the same frame form, and DeepCut has proposed a better part detector and pairwise module.

The top-down methods estimate poses from objects detected by executing detection methods. Many researchers use various detection methods, of which Mask-RCNN [1] is the most commonly used one. Most previous works have done researches on utilizing multi-scale features to estimate poses for different situations and sizes. Simple [80] proposes a method to increase the scale of the output heatmaps through deconvolution layers. Upon a ResNet [24] structure, the work increases the scale of encoded features through newly added deconvolution layers. Although it is a network with a relatively simple extension, it had achieved a quite good accuracy.

Many recent studies have proposed network structures that can utilize features in various scales concurrently. For example, a network that exploits multi-scale features to maintain a high-resolution feature scale is proposed [71]. High- and low-resolution features are provided with separated inference paths with four stations to exchange information along the paths. On every last layer of each station, features are concatenated to be fed into following separated paths. For the concatenations, 1x1 upsampling has been applied for low-resolution features, and a 3x3 convolution layer with a stride size of 2 has been applied to downsample high-resolution features.

Based on the analysis that local and global features are respectively important in localization and classification problems, Cai et al. [8] proposed a method that seeks to integrate local and global features since pose estimation problems require estimation for joint locations of different body parts. The method had achieved state-of-the-art performance in COCO keypoint 2017 challenge with their proposed network structure. A convolution layer operates recursively with a single bottleneck, effectively extracting local and global features.

Localization or detection accuracy rate varies for different body parts, each of which is innately given with different ranges and degrees of movement freedom. Several kinds of research have labeled keypoints that are relatively more difficult to localize, such as ankle and wrist joints. As a similar approach to the object mining method, OHEM (online hard example mining), that tries to solve data imbalance issue from object detection tasks [68],an online hard keypoints mining (OHKM) loss is proposed to solve typical accuracy imbalance among keypoints of pose estimation problems [13]. In the work, a refine network is fed with features of a global network, and both networks are trained with L2-loss functions. The refined network is applied with an OHKM loss to be trained only with detectable parts less accurately. Another work[88] that assigns more weights on joints that are comparably more difficult to estimate, such as partially occluded body parts, is proposed with a generative adversarial network[22] (GAN). The work collects losses for each part calculated from a generator and applies larger weights on joints with larger loss values.

As mentioned, recent papers have focused on improving pose estimation methods by utilizing the feature scales and refinement of poses using local information. While such methods have improved overall performance to an extent, a more direct approach is needed to handle poses that cover a lot of complexity.

We looked at the problem from a different perspective. We have guessed that the imbalance of pose has existed in the pose dataset. If the pose dataset has an imbalance of pose, it has a limitation of improving the pose estimation's accuracy. Because the model couldn't be learned enough to estimate the unusual pose, there is a limit to performance improvement. Thus, we discover and define the pose imbalance and propose the methods to improve the imbalance. More detailed in chapter 4

2.3 Multi-person pose tracking

Multi-person pose tracking means tracking the pose of the same person in the video. The purpose of multi-person pose tracking is to follow the same person for a long time. The multi-person pose estimation and tracking are related because the pose can only be tracked if it is well estimated.

The multi-person pose tracking is divided into the bottom-up approach and the top-down approach, same as the multi-person pose estimation. The bottom-up approach estimates the poses and tracks the poses using temporal information. The top-down approach first detects the bounding boxes of humans and estimates one person's pose on the bounding box. Same as the bottom-up approach, the top-down approach tracks the poses using temporal information.

The main issue in pose tracking is how to use temporal information. Many researchers have conducted research using temporal information in various ways, such as a bounding box tracking algorithm, optical flow, and similarity. [16, 21, 28, 30, 33, 80, 81, 76, 70, 5, 67]

Xiu et al. [81] proposed a pose tracker based on a pose flow that is a flow structure indicating the same person in different frames by pose distance. [66] used a bi-directional long-short term memory (LSTM) framework to learn the consistencies of the human body shapes.

[16] and [65] proposed the temporal map to represent the movement of person. In the case of [16], they used the direction of joints for temporal map information. Their map is called a temporal flow fields (TFF). TFF indicates the transition between two frames. On the other hand, [65] used the direction of limbs to generate the temporal map because the movement of joints is too small. The way of generating temporal map is similar with [16]. Both methods used a

similarity measure to track the poses using the temporal map.

Recently, studies using the top-down approach have produced state-of-the-art results in the multi-person pose estimation. Multi-person pose tracking has also come up with a lot of top-down approaches accordingly.

[80] tracked the pose using the optical flow and the pose similarity. At the previous frame, the bounding box is shifted to the current frame using the optical flow generated between the previous and present frames. They adopted the greedy search to match the same pose. They calculated the similarity using object keypoint similarity (OKS) and selected the optimized association between the shifted bounding box and bounding box of the current frame.

[58] is the online pose tracking method for the top-down approach. They use a Re-ID module to track the pose. As the Re-ID module, A Siamese Graph Convolution Network (SGCN) is proposed. The SGCN calculates a similarity of the poses using graphical information of person parts.

2.4 Datasets and measurements

Various datasets express the pose of people. The most commonly used data are Leeds sports pose (LSP), MPII, COCO, and Pose Track. The annotation of each data is centered on the arms, legs, and head, but the order of each data is different. Also, the methods for measuring the accuracy of each dataset are different. Therefore, in this section, we explained the pose information and measurement metric of each data.

2.4.1 Leeds Sports Pose (LSP) dataset

The LSP dataset is typically used in single-person pose estimation. They gathered the dynamic sports pose images in Flickr, an online image and video hosting site (Figure 2.1). Images contain various sports poses such as Badminton, Baseball, Gymnastics, Tennis, and so on. Most images have only one person. The images were resized so that the person was about 150 pixels. The pose is annotated with 14 joints as shown in Figure 2.1 (a).

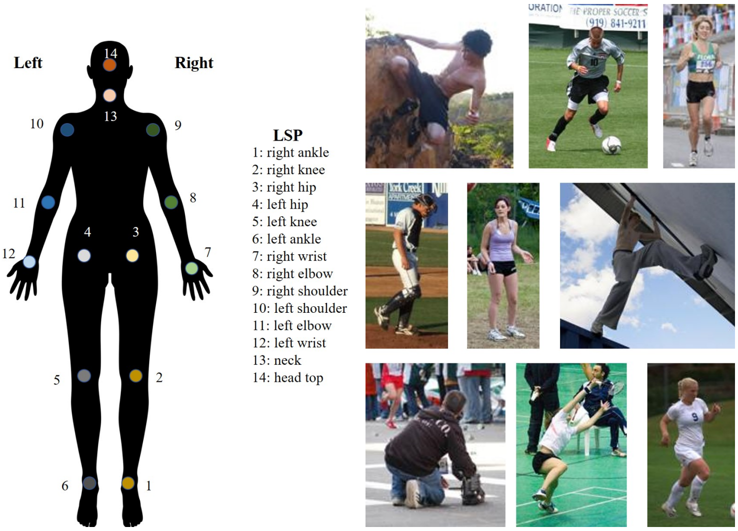
They have two sets, which are original LSP [35] and LSP extended (LSPe) [36]. The original LSP data has 2000 poses images, and LSPe has 11,000 pose images. Two sets have shared the same 1,000 test pose images. Except for the test image, the rest pose images are training images. The LSP has 1,000 training images, and The LSPe has 10,000 training images. The Figure 2.1 (b) shows the example images of LSP.

We use the percentage of correct keypoint (PCK) to evaluate the pose as bellow equation 2.1. $Dist$ means the distance between ground-truth joint (J_{gt}) and predicted joint (J_{pred}). If the $Dist$ is in the threshold ($0.2{*}d_{torso}$), the output of $bool$ is true, which means the joint is corrected. d_{torso} means a distance of torsor in person, and N is the number of people.

$$\frac{\sum_{n=1}^{N}(bool(Dist(J_{gt}^{n}, J_{pred}^{n}) < (0.2 * d_{torso}^{n})))}{N} \tag{2.1}$$

2.4.2 MPII dataset

MPII human pose dataset [3] has 25k images with poses of 40k people. The poses are annotated with 2D locations of 16 joint parts as shown in Figure 2.2 (a). The MPII pose dataset is collected based on 410 types of action categories



(a) (b)

Figure 2.1: The order of annotations (a) and example images (b) in Leeds sports dataset (LSP).

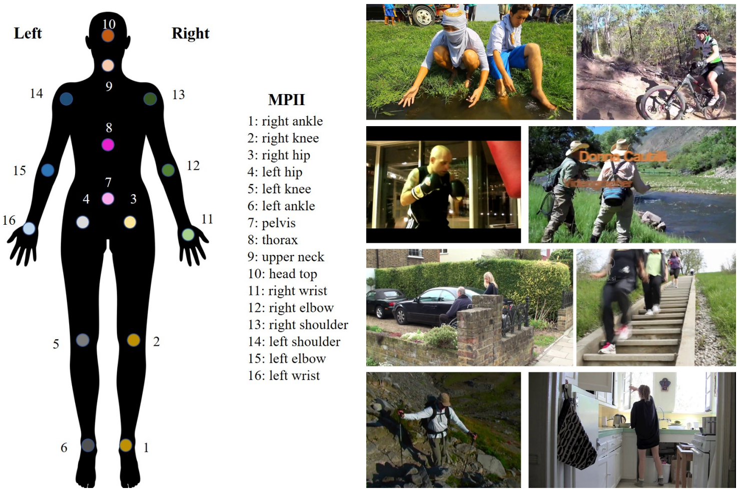
of people (Figure 2.2 (b)). The pose images are extracted from a YouTube video.

The evaluation method of MPII dataset is correct keypoints using head (PCKh) as follow equation . The PCKh measures the accuracy of the prediction same as PCK. The difference between PCK and PCKh is a threshold. PCKh counts the number of joints that are within a distance of head.

$$\frac{\sum_{n=1}^{N}(bool(Dist(J_{gt}^{n}, J_{pred}^{n}) < (0.5 * d_{head}^{n})))}{N} \tag{2.2}$$

2.4.3 COCO 2017 keypoint dataset

COCO 2017 keypoint is a large-scale keypoint dataset [46]. COCO has more than 200k images with poses of 250k people. The poses are annotated with 17 joint parts as shown in Figure 2.3 (a). The pose images include not only daily life



(a) (b)

Figure 2.2: The order of annotations (a) and example images (b) in MPII.

but also sports. This is an unrestricted and extensive range of pose data (Figure 2.3 (b)).

The evaluation metric of COCO is average precision (AP). Originally, the AP is used in the object detection field. The AP computes the average precision value for recall values over 0 to 1. When measuring the similarity between ground-truth box and predicted box, the intersection of union (IoU) are used. The AP of pose estimation is inspired by object detection. In the pose estimation, IoU is replaced by object keypoints similarity (OKS). OKS is used for similarity measures among ground-truth pose and predicted pose as follow equation 2.3.

$$\frac{\sum_i exp(-d_i^2/(2s^2k_i^2))\delta(v_i > 0)}{\sum_i \delta(v_i > 0)} \tag{2.3}$$

The d_i means the Euclidean distances between ground-truth and predicted joint and v_i means the visibility label of ground-truth. s is the object scale and

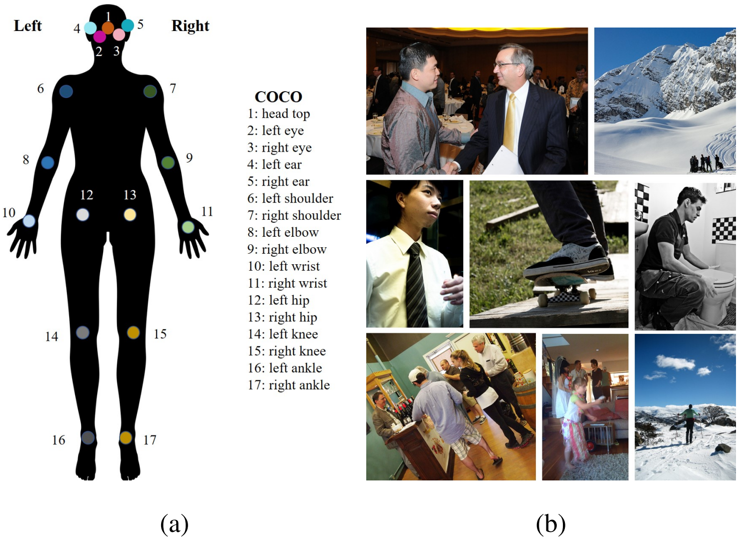


Figure 2.3: The order of annotations (a) and example images (b) in COCO keypoints 2017.

k_i is a constant of per-joint.

2.4.4 PoseTrack

PoseTrack datasets are large-scale benchmarks for human pose estimation and tracking [2]. The PoseTrack datasets have various videos of human activities, including fishing, running, tennis, etc. The datasets include a wide range of pose variations, from a monotonous pose to a complex pose. PoseTrack datasets have videos more than 500 sequences that are expected to be more than 20K frames. It comprises 250 videos for training, 50 videos for validation, and 214 videos for testing.

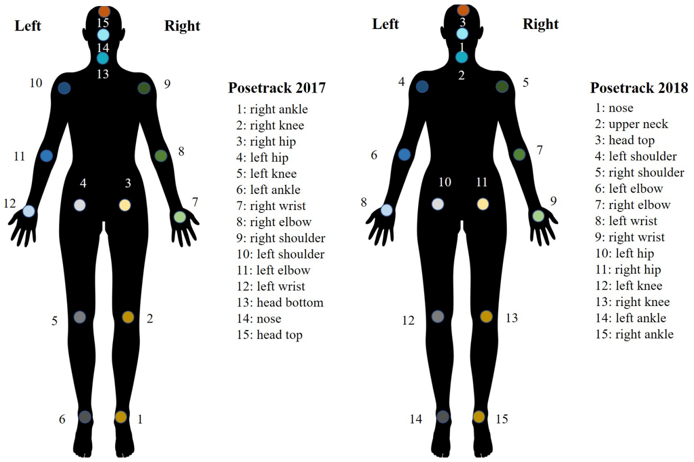
PoseTrack datasets have the two different version: PoseTrack 2017 and PoseTrack 2018. The difference between the two versions is the order of annotation and number of annotation as shown in Figure 2.4.

The evaluation metric of Posetrack is a mean average precision (mAP) for pose estimation and multiple object tracker (MOT) metric for pose tracking. We measured the mAP using the PCKh. First, we calculate PCKh between the predicted multi poses and ground truth poses. Then, only one predicted pose was assigned to the ground truth based on the highest PCKh. Unassigned other poses are counted to false positive. Finally, the mAP is measured using the AP of each body part.

The purpose of MOT is to track multi poses simultaneously and track them for a long time. Among the MOT metric, multiple object tracker accuracy (MOTA) and multiple object tracker precision (MOTP) are used. For MOTA, first, we compared the predicted poses, which have their own personal id and ground truth pose. When comparing the poses, we count three situations: missed detects ($MISS$), False Positives (FP), miss-match ($MissMatch$). $Miss$ is the case of missing tracking trajectory. $MissMatch$ means that tracking information is replaced with other objects. g_t means the number of poses at time t.

$$MOTA = 1 - \frac{\sum_t (Miss_t + FP_t + MissMatch_t)}{\sum_t g_t} \tag{2.4}$$

The MOTP is the total error in predicted poses. The MOTP exposes the ability to estimate the exact position of the object in the tracker, independent of the technique of recognizing the composition of the object and keeping the trajectory constant.



(a) Posetrack 2017 dataset

(b) Posetrack 2018 dataset

Figure 2.4: The order of annotations (a) in posetrack 2017 and (b) in posetrack 2018.

Chapter 3

Single-person pose estimation

The single-person pose estimation is to detect locations of parts in the single person image. There are many components to increase the performance: features, structure of network, optimization, etc. In this section, we focus on how to effectively use the features. The features can be separated into global and local features. The global feature represents the whole body of a person, and the local feature represents a specific region such as hand, leg.

We propose a global-local network that can be learned the global and local feature as end-to-end. The local network is behind the global network, and it works to refine the output of the global network. The first global network is a big deep network that estimates parts' locations using the global features, and the second local network is a small network that modifies the parts' locations using local feature.

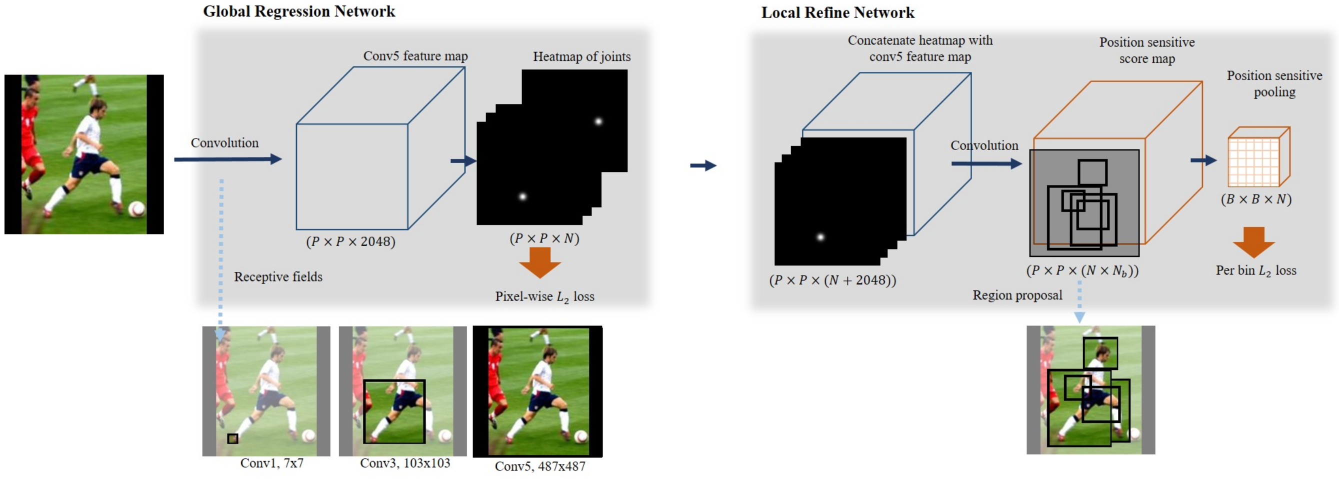


Figure 3.1: Overall architecture of the proposed method (global-local network). The output of the global network is used as an input to a local network to refine the location using a variety of region proposals. On the left, we show each receptive field of features after the corresponding convolution layer with its size (e.g., 7 × 7 for conv1 layer). On the right, several region proposals are shown.

3.1 Global-local Network

The global network regresses the whole body parts using the Heatmaps. We generate the Heatmap centered on the location of each part through a Gaussian map. Because human pose estimation is a highly non-linear problem, it is difficult to directly regress the locations of the parts. Rather than directly regressing the position of the parts, we followed a simple alternative method, which regressed a set of Heatmap centered at the visible target joints as in [7].

We used the ResNet-101 model[24] to a global network to jointly regress the position of each part. The ResNet-101 network has a huge receptive field, as shown in Figure 3.1. The global network learns the Heatmap using a wide range of global features obtained from a receptive field. To increase the resolution of the Heatmap, the output feature of ResNet-101 was enlarged by adjusting the stride of convolution filter. Specifically, the stride of the *conv3*, *conv4* and *conv5* layers are changed to 2,1 and 1 respectively. Lastly, the output feature map of *conv5*) is 14×14.

We have used a pixel-wise L_2 loss to regress the Heatmap as follows:

$$L_g = \frac{1}{N} \sum_{n=1}^{N} \sum_{x,y} \left\| H_n(x,y) - \bar{H}_n(x,y) \right\|^2 . \tag{3.1}$$

Here, H_n means the predicted heatmap and $\bar{H}_n$ is the ground truth heatmaps. N is a number of parts and (x, y) means pixel locations of a Heatmap.

The local network is executed to refine the pose of global network. The output of global network are fed into the local network. The output is a combination of the output of *conv5* and the Heatmaps. The local network learned the partial Heatmap using direct local evidences.

The low layer of networks can represent the local information by the small size of the receptive field, which regionally covers the image. As shown in the

Layer name	Output size	Layer size
conv6	$P \times P$	$1 \times 1, 2048, 1$
conv7	$P \times P$	$1 \times 1, N_b \times N, 1$
PS-ROI pooling	$B \times B \times N$	

Table 3.1: The structure of the local refine network. The values in the layer size tap means (kernel, channels, stride). P is the output size of *conv6*, which depends on the stride of the global network. N is the number of parts and N_b is the number of bins.

bottom-left of Figure 3.1, the small size of the receptive field on the conv1 layer includes only the foot. Furthermore, we need local information which has various locations or scale. Because the receptive fields of convolution layer have fixed sizes of strides, it is restricted to get the various local information.

Thus, we use a region of interest (ROI) to extract the local evidence. ROIs have various scales, sizes and positions, and they contain a variety of combinations of body parts. These characteristics help to increase the expressiveness and generalization power of the network. As shown in the bottom-right of Figure 3.1, the detected ROIs have a various size bounding box and a various union of body parts. We expect that the local feature of ROIs helps to increase the expressiveness and generalization of the network.

We adopted the position-sensitive score maps and the position-sensitive ROI pooling form R-FCN [43] to train the local information. The R-FCN is one of the most popular object detectors. In the original R-FCN network, the position-sensitive score map channel represents the specific bin of the ROI for each class.

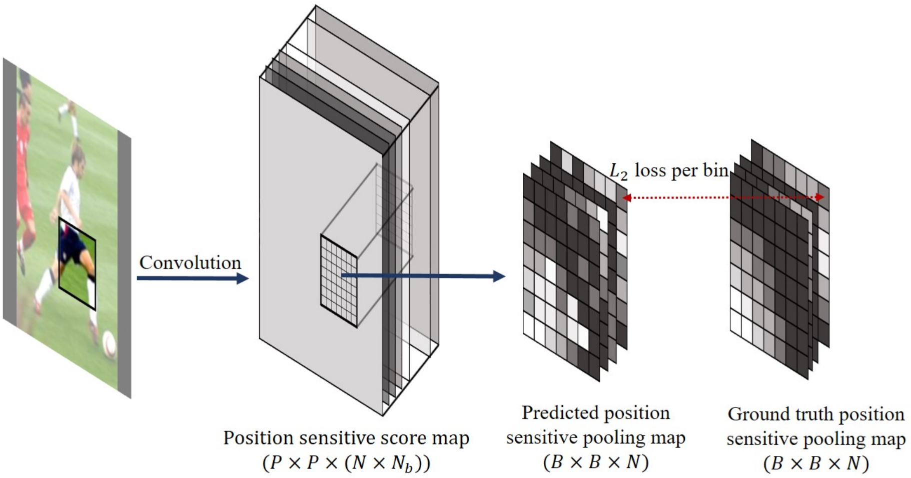

Convolution
L_2 loss per bin
Position sensitive score map
$(P \times P \times (N \times N_b))$
Predicted position sensitive pooling map
$(B \times B \times N)$
Ground truth position sensitive pooling map
$(B \times B \times N)$

Figure 3.2: The position-sensitive model applied to our local network.

In this paper, the channel means the specific bin of the ROI for each body part. The position-sensitive score map has $N_b \times N$ channel to describe spatial information for each joint. Here, N is the number of parts, and N_b is the number of bins to which ROIs are divided. Note that $N_b = B \times B$ in the figure. The position-sensitive ROI pooling is applied to the score maps to generate the feature maps used to locate the body parts as shown in Figure 3.2.

The Table 3.1 shows the structure of the local refine network. The local network has three layers: one convolutional layer, one ReLu layer, and one position-sensitive score map layer. The P means an output size changed by the stride of the last layer on the global network. The B is the number of bins on the bounding box.

We used the position-sensitive ROI pooling to extract the features on ROI region. The position-sensitive ROI pooling works average pooling for all channels on each bin from the position-sensitive score map. The output value of b -th bin after the pooling is calculated as

$$r(b) = \frac{1}{E} \sum_{(x,y)\in bin(b)} C_b(x_0 + x, y_0 + y) \tag{3.2}$$

where E is the number of elements in a feature map that are inside the b-th bin, C_b is the value of the feature map that corresponds to the b-th bin, (x_0, y_0) is the offset of the top-left corner of an ROI, and (x, y) are the offsets in the b-th bin.

We need to selectively detect the ground-truth Heatmap based on ROI region and applied the ROI pooling same as position-sensitive pooling. As shown in Figure 3.2 The average score value in each bin was trained using a L_2 loss where the loss function is as follows:

$$L_l = \frac{1}{N} \sum_{n=1}^{N} (\frac{1}{N_b} \sum_{b=1}^{N_b} \left\| r_n(b) - \bar{r}_n(b) \right\|^2). \tag{3.3}$$

Here, $r_n(b)$ is the value after the selective pooling in the b-th bin of the n-th joint and $\bar{r}_n(b)$ is the corresponding ground truth heatmap.

Figure 3.3 shows a visualization of position-sensitive ROI pooling on two region proposals. The number of bin is 7. E_1 and E_2 are the region proposals extracted from Edgebox. The pooled features that are used to locate right ankle, right shoulder, and head are visualized for both region proposals. It is verified that the proposed local network successfully locate the joints in each region proposal. For example, because E_1 included the head and the shoulder but not the right ankle, the pooled features for the right ankle have low values while the values of the other joints are high at the position of the joints.

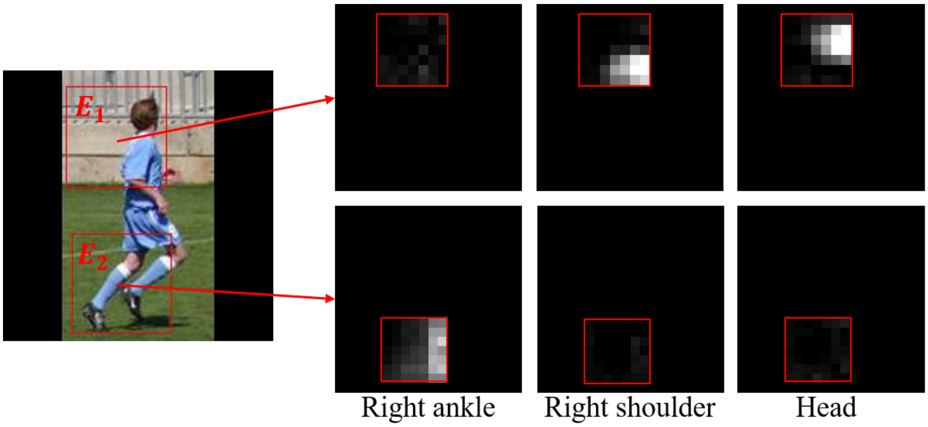


Figure 3.3: Visualization of position-sensitive score maps on two different ROIs, E_1 and E_2.

3.2 Experiments

To prove the efficiency of the proposed method, we experiment with our method on the Leeds sports dataset (LSP). The proposed method is implemented using the deep learning open-source library Caffe[32]. We use the baseline model of the global network as ResNet-101 network and use the pre-trained parameter, which is trained on ImageNet dataset[40]. For training, the learning rate is 0.0001, weight decay is 0.0005, and momentum is 0.9. We train the proposed model in two steps. First, the global network is learned except the local network. As it were, we train the ResNet-101(global network) on the LSP dataset using a pre-trained parameter of ImageNet. And then, the full network of the global-local network are trained end-to-end. When training the full network, only one image input to the network for every iteration. ROIs of the input image is generated via EdgeBox [89]. Among them, 20 ROIs are randomly selected and are fed to the local network.

	Head	Shoulder	Elbow	Wrist	Hip	Knee	Ankle	Total
Local	74.8	63.9	44.7	29.7	66.6	47.9	28.3	50.8
Global	89.3	71.5	58.0	51.0	70.5	66.5	62.5	67.0
Global(14)-local	91.8	76.0	64.7	58.6	76.9	72.9	68.8	72.8
Global(14)-local*	92.3	79.1	69.2	62.9	80.8	76.0	71.5	76.0
Global(28)-local*	96.2	85.4	76.1	71.2	85.7	81.8	76.2	**81.8**
Fan[18] et al. CVPR'15	92.4	75.2	65.3	64.0	75.7	68.3	70.4	73.0
Yang [83] et al. CVPR'16	90.6	78.1	73.8	68.8	74.8	69.9	58.9	73.6

Table 3.2: PCK-based comparison on LSP. A threshold value was measured at 0.2 (@0.2). The mark * indicates weights from the additional fine-tuning step is used.

Results on the LSP dataset are shown in Table 3.2 and Figure 3.4. We compared the performance of the proposed global-local network (Global-local) with the case that only the global network is used (Global), the case that only the local network is used (Local), and recent human pose estimation methods [83, 18]. Global network was based on the ResNet-101 and used L_2 pixel-wise loss to regress heatmap. Local network was also based on the ResNet-101, and L_2 loss is used for the position-sensitive score maps. Yang et al. [83] proposed a combined network with the expressive deformable mixture of parts. Fan et al. [18] proposed a dual-source CNN without using any explicit graphical model. They used the local information in image patches. Unlike our method, they put the cropped image on input image. We compared those methods as representative methods of exploiting global information [83] and local information [18]. In the method tab of Table 3.2, the numbers in parentheses are the size of the output heatmap.

The performance of the local network and the global network were 50.8%

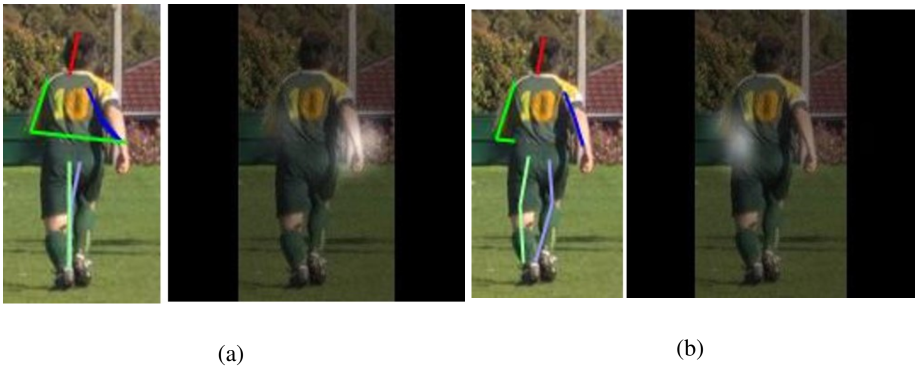


Figure 3.4: Compared result of global network and the proposed method. (a) Global network, (b) Global(14)-local network. For both (a) and (b), the left image shows the position of body parts and the right image is the original image overlapped with the heatmap of the left wrist.

and 67% in terms of PCK accuracy respectively. The accuracy of the proposed global-local network is 5.8% higher than that of the global network. From the results, we can see that using only local or only global feature is insufficient for expressing complex human poses. To boost the performance, we added intermediate fine-tuning step before training the global-local network. For the model, we added the deconvolution layer [49] after the last convolutional layer of the ResNet. The deconvolution layer upsamples the size of the feature maps to 224×224. Then, 224×224 heatmaps are generated, which are trained using the L_2 loss with the ground truth heatmaps. We found that using the weight trained from the intermediate fine-tuning step improves the performance. In Table 3.2, methods with the postfix (*) are the networks trained from the weights that comes from the intermediate step. It can be seen that the intermediate fine-tuning improves the PCK performance by 3.2%. As stated Section 3.1 that the stride of the convolutional layers inside the ResNet is adjusted to control the output heatmap size, we tested two different output heatmap sizes, 14×14 and 28×28 to show the importance of the output heatmap size. When the output heatmap size is doubled, PCK has been improved from 76.0% to 81.8% by 5.8%.

The model that shows the best performance is Global(28)-local* model of which PCK@0.2 is 81.8%. Note that the PCK of head regression shows superior results to the compared methods [83, 57]. Figure 3.5 shows the PCK curve according to the normalized distance of each part. It can be seen that the proposed Global(28)-local* model outperforms the other methods in estimating a variety of parts. Qualitative results from LSP dataset are shown in Figure 3.6.

The proposed method included the local network to effectively learn local information. The role of the local network is to find the local information which cannot be inferred in the global network. Figure 3.4 is the example that shows the effect of the local network. The images that shows the results of all parts locations and the images that show the output heatmap of the left wrist are shown. Figure 3.4(a) is the results from Global model, and Figure 3.4(b) is the results from Global(14)-local model. In the case of Global model which aggregates the global information, the heatmap of the left wrist has high values around the right wrist which is visible in the image. On the other hand, in the case of Global(14)-local model which exploits the local information as well as the global one, it is possible to refine the heatmap even for the occluded part, and the position of the left wrist is correctly inferred as shown in Figure 3.4(b). Thus, we conclude that the global-local network is able to learn both global and local information.

Next, we tested the Global(28)-local* network model that had been trained using the LSP dataset on the UCF sports dataset and provide qualitative results. Figure 3.7 shows the results of representative frames of various videos such as

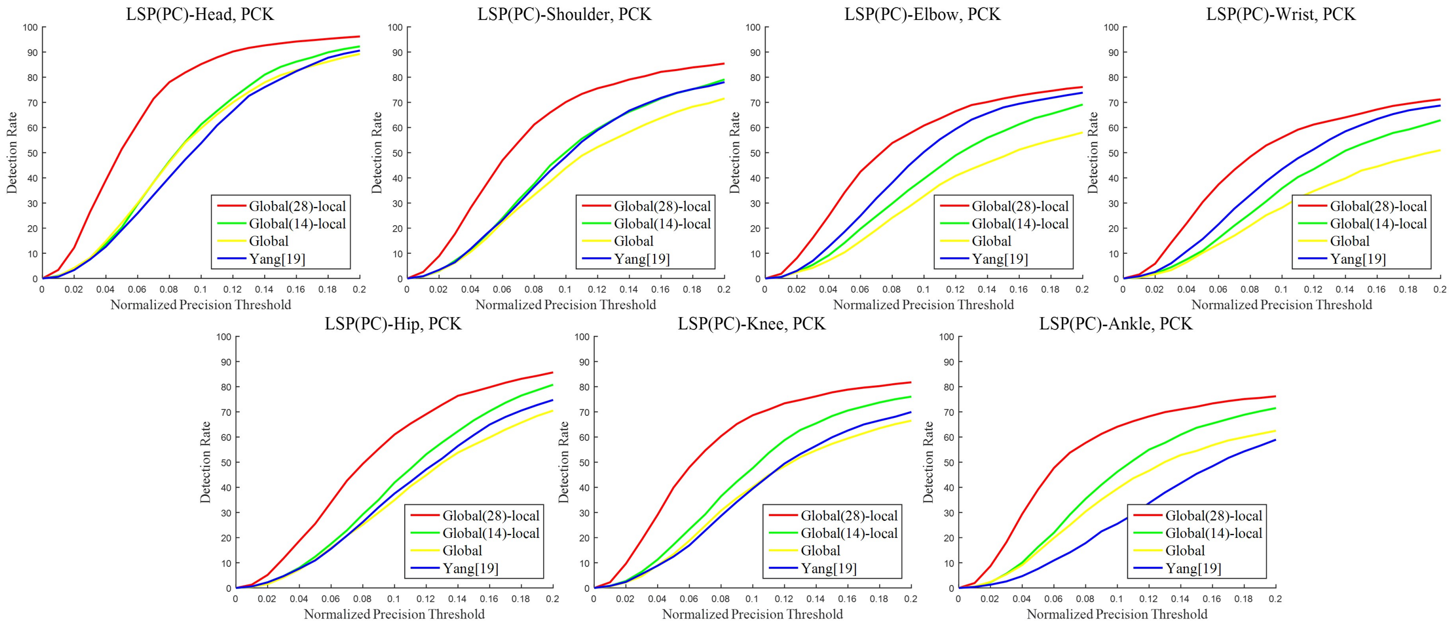
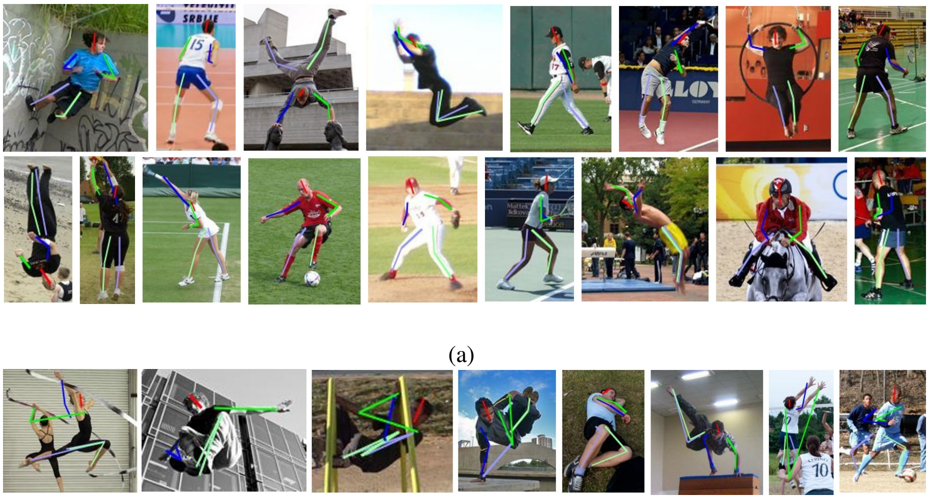


Figure 3.5: Quantitative typical results on the LSP dataset using the PCK. Our proposed method the highest performance especially on head part. The other parts tend to be similar.



(a)

(b)

Figure 3.6: Qualitative results of our method on LSP dataset.
(a) Successful results, (b) Failure results. Proposed method was successful in various poses. As like squatting pose, many joints had self-occlusion, then it made a failure result.

kicking and skate-boarding. The network successfully locates parts even though it is not trained on the UCF-sports dataset. Especially the head, knee and shoulder positions were estimated with small amount of errors. Compared to LSP dataset, UCF sports dataset is more challenging since the datset contains low resolution and blurry images as can be seen in Figure 3.7. The proposed network produces reliable results despite those challenging conditions. Lastly, we showed the failure cases of our methods in Figure. When a person is in a squatting position or a person is wearing loose clothes, it was difficult to locate the body parts. The proposed method also suffers from self-occlusions. In those case, it is difficult to regress the part's location correctly on a single frame. Using a tracking algorithm that contains temporal information can be a solution

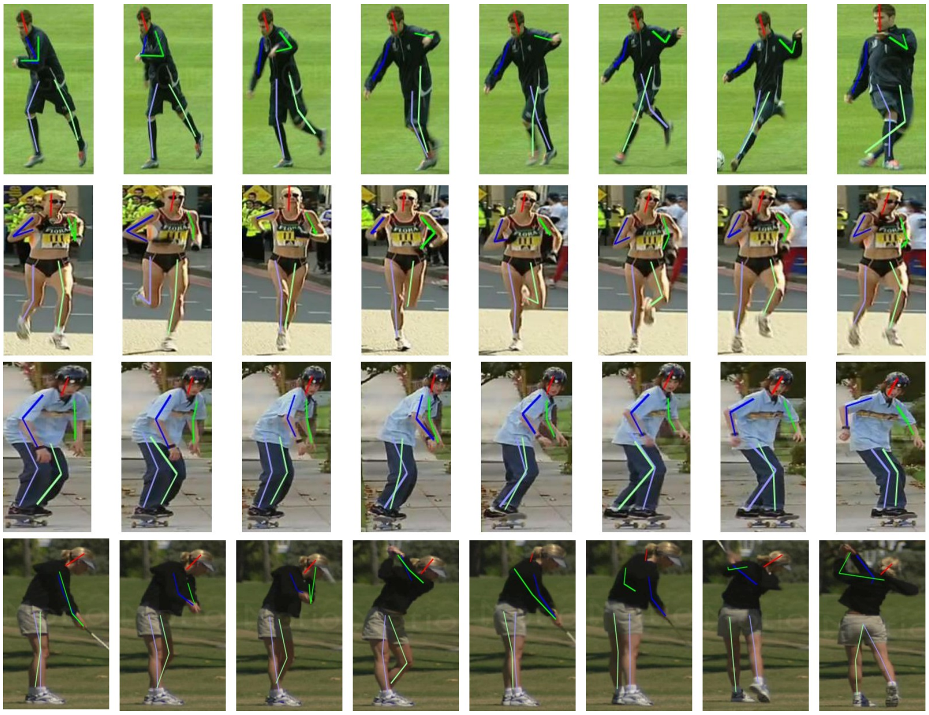


Figure 3.7: Qualitative results of our method on UCF dataset time-sequentially.

for the case. Our method performs slightly worse than the state-of-the-art. However, since the state-of-the-art methods are constructed as a repetitive structure or a very deep structure, our proposed method will work a little faster. Also, attaching our local refine network to other structures which has been previously proposed will be left for the future work.

Chapter 4

Rare pose estimation

In pose estimation, imbalance among parts refers to the difference of each part's accuracy, and most works try to diminish this accuracy gap.

A data imbalance is inherent in many applications [34]. The imbalanced data could learn bias towards the majority class and ignore the minority class. Rarely cases in which you need to handle minor classes exist. The lack of gathering rare data due to the low frequency occur. Thus, it is necessary to find a way to detect the minor class well in the current data.

Various methods have been proposed to solve the class imbalance. Representatively, over-sampling, under-sampling, re-weighting the loss and synthetic minority over-sampling techniques are used [64, 77, 74, 47, 20, 6]. Most researchers have proposed methods trained by focusing on minor classes that have a small number of data. Unfortunately, simple ways such as increasing the number of minor class data may reduce the performance of the major class. Also, they need a specific parameter to set an optimal model. Methods such as Groups Softmax[44] have been proposed to overcome this problem. [78, 31, 44, 26, 84]. [44] divides classes into several groups by the number of data and trains the model group-wise. However, since pose data has no class label, existing methods cannot be applied directly to the pose imbalance problem.

Our method takes a different approach by defining pose imbalance differently and suggests several methods to resolve this problem. Most available pose datasets consist of data samples collected in daily-based situations that are natural in motion (e.g., walking and playing sports). Huang et al.[27] has reported that 85% of COCO[46] dataset is composed of standing poses, with the rest being either sitting or lying poses.

Their work argues that the severe imbalance in the data pool makes the generalization of pose detection difficult. However, pose imbalance in their work is mainly decided by whether a person is stood upright. Considering that various factors such as self-occlusions may affect the overall pose estimation performance even among the standing poses, a more deductive method must be suggested to quantitatively measure pose abnormality for a better analysis of the imbalance in pose data.

In this Chapter, our method's rare poses are newly defined for the first time, and other approaches to improve the estimation performance against rare poses are proposed consequently.

First, we establish an appropriate definition of unique pose samples to solve the problem to enhance the robustness of a pose estimator. To this end, we firstly define a *rare pose* as "a pose that occupies as a minority within a data population". Examples of such rare poses include squatting poses, poses with self-occlusion, horizontally extended poses (e.g., swimming poses), and more. A minority of a dataset, in this context, refers to outliers from the distribution of whole data.

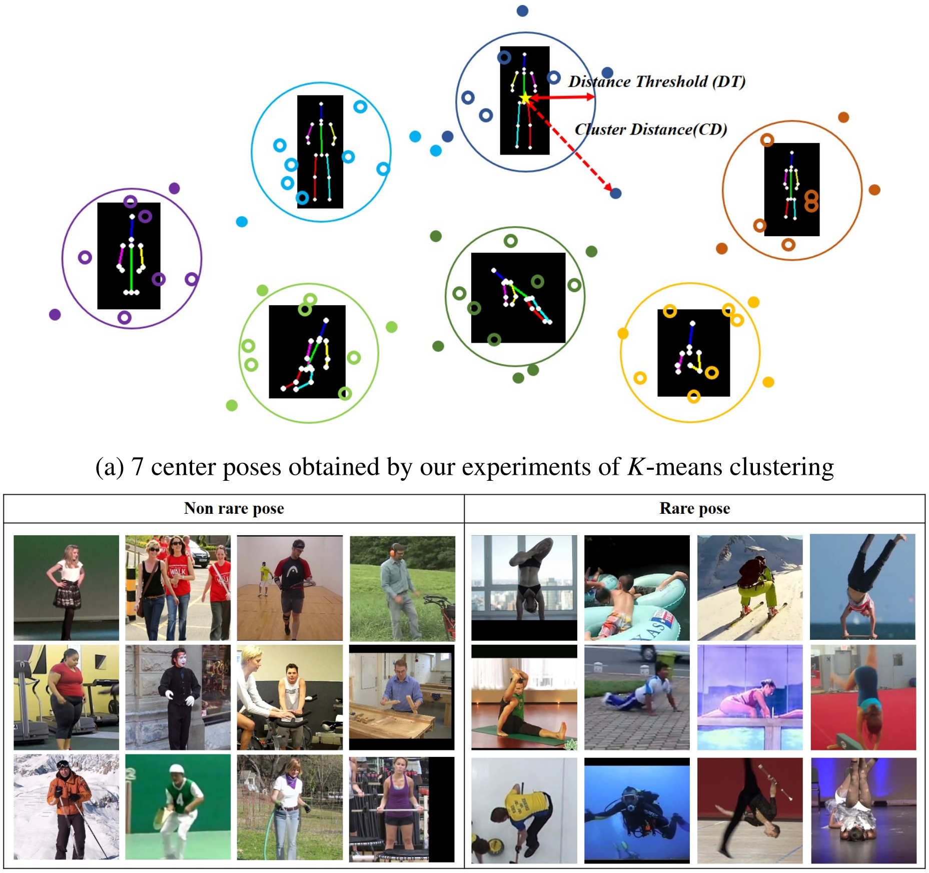
An outlier generally means that a data sample is significantly different from others, the meaning of which is also applied for rare poses. However, unlike outliers, rare poses are not to be discarded from the set. Among various methods proposed for outlier detection, we use K-means clustering to detect rare poses because of its computational advantage of being a training-free clustering method. In this work, we empirically show that it is suitable to define an outlier as a rare pose that is distant from a center of clusters, unlike other dense data samples near the centers. Once all samples are clustered, a sample's *cluster distance* (CD), the distance between a pose sample and the center point of its classified cluster, is compared with a pre-defined *distance threshold* (DT) value to determine whether it is a rare sample or not.

Fig 4.1 (a) illustrates a distribution of MPII[3] pose data samples and their clusters resulted by K-means clustering with $K = 7$. While the solid red arrow represents a DT, the dashed arrow represents the CD of a pose sample. If a sample's CD is larger than DT, the pose is classified as rare data. Fig 4.1 (b) shows images of rare and non-rare pose samples selected by our proposed method. Classified pose samples show the clear difference of complexity between rare and non-rare poses.

Since not only the rare poses are difficult to detect, but also there exists only a scarce amount of similar data samples, we propose following three techniques to enhance the pose estimation performance:

1. Duplication of rare pose data samples. In addition to the given training data samples, we repeat rare pose data samples once more within the dataset.

2. Addition of synthetic rare pose data samples. We have created and added synthetic samples with annotations of rare pose samples to the training



(a) 7 center poses obtained by our experiments of K-means clustering

(b) Examples of 'non rare' (CD $<$ DT) and 'rare' poses (CD $\geq$ DT) in MPII dataset

Figure 4.1: Illustration of rare pose identification via clustering in MPII pose dataset.

(a) Center poses often represent typical poses such as standing upright or tilted towards left or right side. The dashed red arrow represents the distance between the cluster's center pose (yellow star) and a pose sample (small circle). A pose is classified as a *rare* pose (solid circle) if this cluster distance (CD) exceeds a distance threshold (DT) (large circle). Other small circles (hollow circle) is classified as a usual pose (Non rare pose). (b) shows examples of rare and non-rare poses which are identified with a DT value of 1.2.

dataset.

3. Rarity-based loss weights. After clustering poses, the distances between poses and the center points of their corresponding clusters are used as weights for learning the amount of parameter update.

We have conducted comparison experiments among our proposed techniques to evaluate performance improvement on rare pose estimation. To further show the effectiveness of our proposed methods, we also provide quantitative results and mean average precision (mAP) scores on COCO keypoint[46] and the percentage of correct keypoints (PCKh) on MPII[3] datasets which are commonly used benchmarks of 2D multi-person pose estimation. As baselines, we have used Simple[80] and CPN[13] models which are popular networks in the multi-person pose estimation problems. From the experiments, we observed a larger increase of accuracy scores for rare pose samples.

4.1 Identification of Rare Poses

Conventionally, a pose sample that is rare represents either poses with a lot of invisible parts or an unusual pose as shown in Fig 4.1(b). Many methods have previously struggled from estimating such samples because of their rarity within a dataset and no clear definition to distinguish them from the usual ones. Although rare, the pose estimation for these rare poses is critical to human eyes in some areas that involve a lot of pose deformation such as gymnastics and extreme sports.

In order to improve the performance on rare poses, we firstly need to have a clear measure to identify a rare pose. Since a 2D image sample with a pose P is composed of (x, y) coordinate values of J joints, i.e. $p = \{(x_j, y_j)\}_{j=1}^{J}$, which can be considered as a $2J$-dimensional continuous real random vector, it is very complicated to set a clear definition of a rare pose using the coordinate. Even if we set a heuristic rule, it takes time and cost because people have to label it. In order to solve this problem, we propose a new rare pose identification method which does not require additional learning.

The rare poses occupy a small fraction within a dataset, and have a relatively large difference from the majority of data, appearing as outliers. For the computational advantage, we aim to detect the outliers using a simple clustering method without any other additional learning of anomaly detection. In this paper, we conduct the *K*-means clustering method [48], a popular unsupervised clustering method that searches for clusters with the minimum distance between the *K* cluster centers and data samples. The method allows grouping similar poses as densely as possible and labeling of rare poses which are relatively distant from the centers of clusters.

2D location information of body joints, without color and texture information, is considered to classify the poses because color and texture information tends to depend on various factors such as clothes and skin colors and thus spans an excessively wide search space. Clusters are therefore defined only by 2D coordinates $p = \{(x_j, y_j)\}_{j=1}^{J}$ of parts from the image space which mainly represent uniqueness of each pose sample. When detecting the location information of each part, the object is positioned in the center of a certain bounding box with similar scales as illustrated in Fig. 4.1(a).

First, the training data are classified based on the predetermined number of clusters K. In doing so, the distance between the pose p_i and the center of the corresponding cluster m_c is measured as follows which is denoted as the cluster distance d_i^c:

$$d_i^c = min\{\|p_i - m_c\|\}_{c\in\{1,\cdots,K\}} \tag{4.1}$$

Then, the cluster distance is used to to determine whether the pose p_i is rare pose or not. The Fig 4.6 (b) and (c) show the histograms of cluster distance. Both graphs confirm that the number of samples suddenly decreases from a certain value. We consider this point corresponding to a sudden drop of the number samples as the distance threshold (DT) τ for the rare pose. We have conducted experiments to measure accuracy for this threshold setting. The same DT (τ) is applied to all clusters. Finally, the pose p_i is classified as a rare pose R or an usual pose U as follows.

$$p_i \in \begin{cases} R & \text{if} \quad d_i^c \geq \tau \\ U & \text{otherwise.} \end{cases} \tag{4.2}$$

4.2 Enhancing the performance of rare pose estimation

We have empirically found that the reason of low performance on rare poses is not only that they are difficult to estimate, but also that only a small amount of such samples are present in a dataset compared to relatively simpler poses. Table4.1 shows accuracy of *Simple*[80] based on various distance thresholds between the centers of clusters and their corresponding poses. It can be seen from the results that the accuracy and the amount of rare pose data decreases as the threshold increases. With such understanding, we propose following methods to improve the performance against rare poses by focusing on the rare data: Addition of duplicates of rare pose samples and synthetic samples with rare pose labels to the training set and an objective function that reflects rarity based on the distance from the cluster centers.

Table 4.1: The accuracy of *Simple*[80] for samples of train data with cluster distance that exceeds each distance threshold (τ).

(a) Results of COCO

τ	All (0)	<1.4	1.4	1.5	1.6	1.7	1.8	1.9	2
#data	257252	249789	7463	3317	1395	551	217	95	45
mAP	72.4	80.7	56.5	50.4	44.3	35.4	30.7	22.8	15.7

(b) Results of MPII

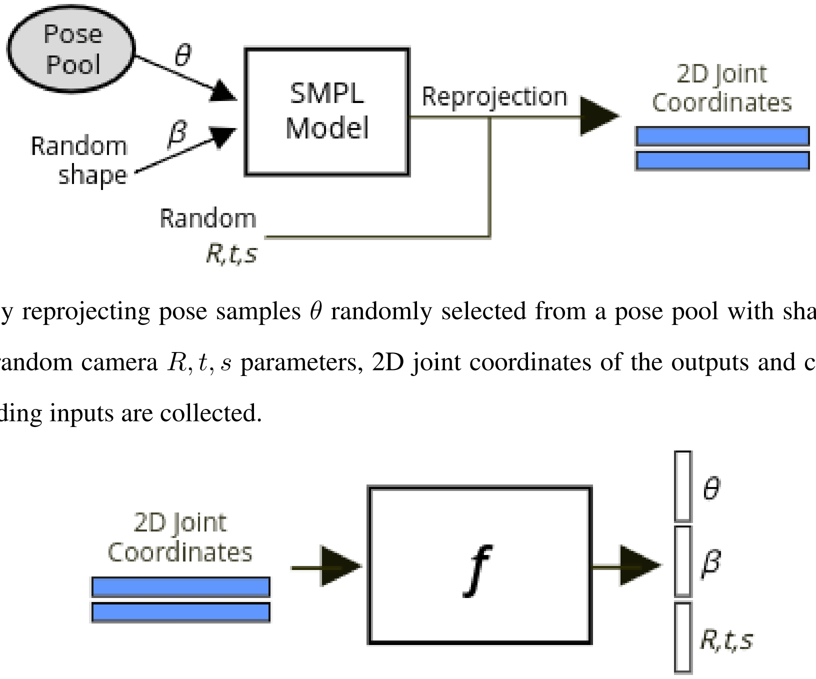
τ	All(0)	<0.5	0.5	0.6	0.7	0.8	0.9	1	1.2
#data	22246	16885	5361	3090	2316	1709	1044	644	282
PCKh	97.92	98.604	95.73	94.721	94.123	93.406	90.93	86.85	74.09

4.2.1 Duplication of Rare Pose Samples (DRP)

One of the effective ways to improve general performance is to train with a better-balanced dataset. To achieve a similar effect and to provide data samples from the same domain as the majority of training data, instead of collecting additional data, we have added duplicates of rare pose samples. The rare samples are firstly labeled from the training data in a preprocess and they are simply repeated once within the training set.

The ground truth poses, $P_{gt} = \{p_1, p_2, ..., p_N\}$, are used for learning. Once we detect the rare poses, $R = \{p_i | d_i^c > \tau\}$, from the ground truth poses $R \subset P_{gt}$, we duplicate R which are added to the original P_{gt} to constitute P_{drp}. Then, P_{drp} is fed into the network for learning.

This method is a simple but effective way to augment scarce samples from the same domain. There exists a risk on a model to over-fit on the data samples



(a) By reprojecting pose samples θ randomly selected from a pose pool with shape β and random camera R, t, s parameters, 2D joint coordinates of the outputs and corresponding inputs are collected.

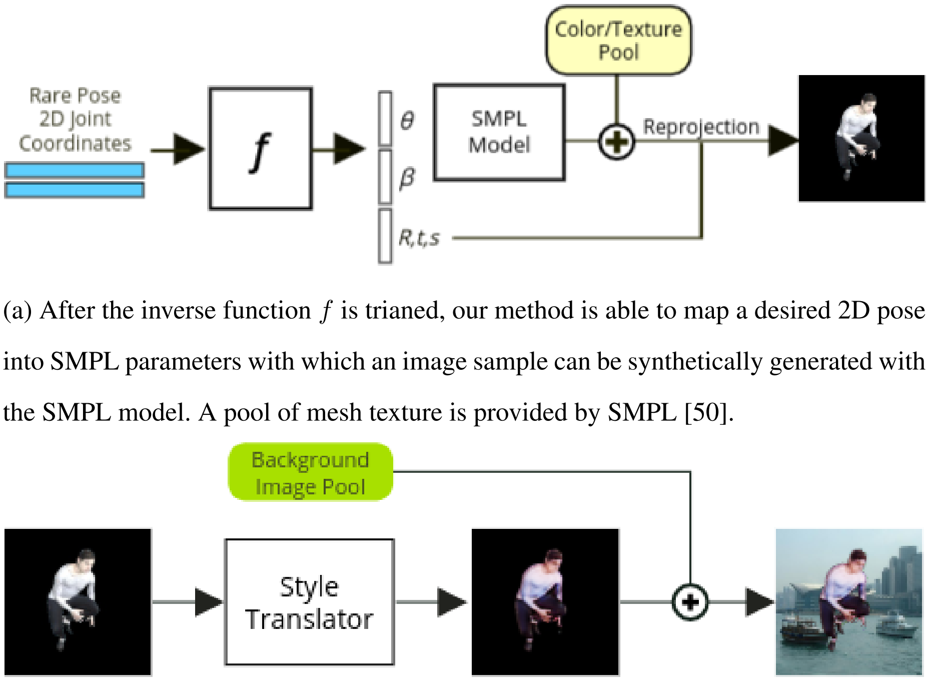
(b) Since rare pose data samples from MPII and COCO are defined as (x, y) coordinates of body parts within images, an inverse function f is newly learned in order to map the given joint coordinates to SMPL parameters needed.

Figure 4.2: An overall illustration of the synthetic rare pose data generation process.

that are duplicated, however in the case of rare pose samples, since their distribution is comparably smaller than other samples, the overall performance is not severely altered.

4.2.2 Addition of Synthetic Rare Pose data (ASRP)

Since data collection is expensive and collecting rare pose data is particularly more difficult, it is reasonable to synthetically generate rare samples with ac-



(a) After the inverse function f is trianed, our method is able to map a desired 2D pose into SMPL parameters with which an image sample can be synthetically generated with the SMPL model. A pool of mesh texture is provided by SMPL [50].

(b) Each synthetic image is transitioned with realistic texture by a pre-trained synthetic-to-real style translator. Lastly, a random image is then assigned in the background to create a rare pose data sample.

Figure 4.3: An overall illustration of the synthetic rare pose data generation process.

companying annotation ground truths if more various color/texture must be considered[56, 82].

For generations of synthetic rare pose data, we have used SMPL human body model [50]. SMPL is a mesh deformation model that is defined by pose θ and shape β parameters for controlling the model's 3D mesh outputs. The constructed 3D human mesh models from SMPL can then be projected to 2D images with camera parameters consisting of scales s, translations t and rotations R to be re-created as pose data samples with 2D joint location annotations.

However, since the annotations for rare pose data samples are given with image coordinates (x, y) for each joint, we were required to map the 2D coordinates to the corresponding pose and camera parameters that allow creating and reprojecting SMPL mesh models in order to align the annotations of resultant synthetic samples with those of given 2D rare pose samples.

Fig 4.2 and 4.3 illustrates the overall generation process of synthetic rare pose data samples. For authenticity of human poses, SMPL provides a pool of known pose parameters and color/texture information for each mesh collected from real poses, which we utilize for generating random synthetic samples. As an initial phase, since we are required to learn a function f that maps 2D joint coordinates to corresponding SMPL parameters θ, β, R, t, s, we collect inputs and outputs of SMPL models in order to train f (see Fig 4.2(a)). With the trained f with the setting in Fig 4.2(b), we are able to find the right parameters that result in a 3D human mesh model with 2D annotations when re-projected to the 2D image space, as depicted in Fig 4.3(a). A random image then fills the background in order to create a synthetic rare pose sample which can be shown in Fig 4.3(b). A pool of body texture provides color values of each mesh that expresses color and wrinkle of clothes or skin. Backgrounds are randomly cropped patches from randomly selected samples of VOC2012 dataset[17]. Examples of synthetically generated pose data are shown in Fig 4.4. After generating the resultant synthetic pose samples $S = \{p_1^s, p_2^s, ..., p_m^s\}$ are generated, the samples are added upon the given training set of ground truth poses P_{gt} so that poses that are used for training are $P_{all} = P_{gt} \cup S$.

To generate more realistic synthesized samples, we have pre-trained a generator that translates styles from synthetic to real. We have used U-GAT-IT [38], an unsupervised generative model for image-to-image translation, for its com-

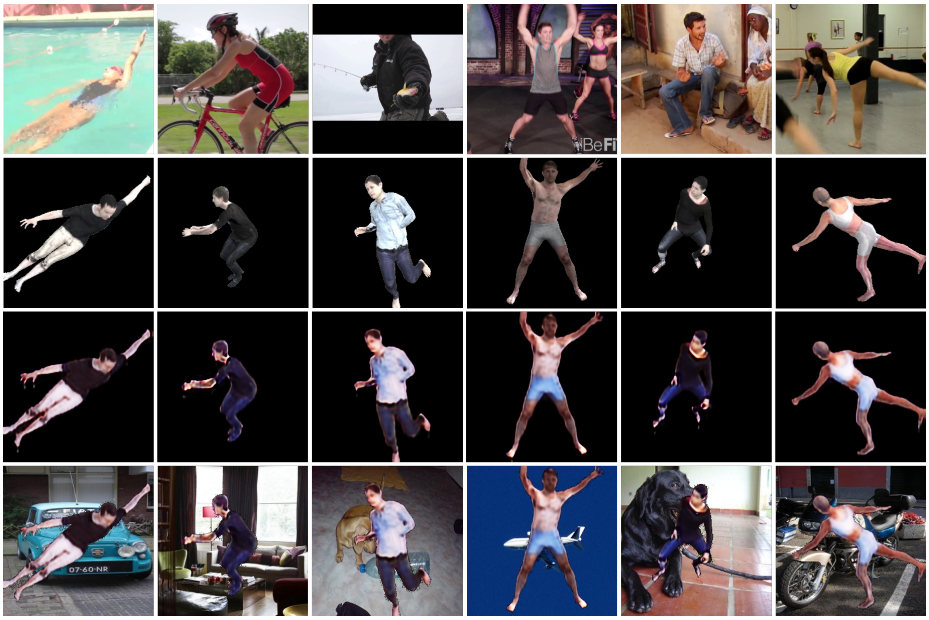


Figure 4.4: Examples of synthetic MPII rare pose data generated

Images in the first row show MPII data samples that are defined as rare poses according to our method. Images in the second row are the synthetic samples. Images in the third and forth rows show synthetically generated samples of which style is transferred from synthetic to real, respectively, with and without backgrounds.

petent performance of style transfer from cartoon to real and vice versa.

The model f is structured as PoseResnet with ResNet50 structure from Simple[80] that takes 256 x 256 sized inputs. The network is selected for its reported and empirical efficiency. We have selected 13 keypoints aligning universally with SMPL, MPII and COCO datasets, so that after the network is trained with keypoints of SMPL, rare pose annotations from MPII and COCO can be used to generate corresponding synthetic samples (See Fig 4.4). The network is fed with 13 channels of heatmaps that are created based of 2D coordinate inputs.

4.2.3 Weighted Loss based on Cluster Distance (WLCD)

In object detection problems, soft sampling methods are applied to solve data imbalance issues. [9, 41, 45, 60] The degree of contribution is assigned to a value between 0 and 1 for each data to solve data imbalance problem. Similarly, after i-th pose is assigned with a cluster class $c \in \{1, \cdots, K\}$ through K-means clustering, a cluster distance d_i^c, a distance between the pose and its corresponding cluster center, can be measured. The cluster distance values are applied as weights when calculating the loss, which yields larger gradient updates for rarer poses.

The weighted objective based on cluster distances of our proposed method is as follows:

$$L = \frac{1}{N} \sum_{i}^{N} \sum_{j}^{J} \left(h_{ij} - \hat{h}_{ij}\right)^2 * w(d_i^c) \tag{4.3}$$

where a loss function L with a weight $w(d_i^c)$ is multiplied to the mean square error (MSE) between heatmap predictions $\hat{h}_{ij}$ and ground-truths h_{ij} for j-th joint from i-th pose data. Here, N and J are the number of training samples and the number of joints respectively. The weight is determined as follows:

$$w(d) = \begin{cases} 1 & \text{if} \quad d < \tau \\ 1 + (d - \tau) & \text{if} \quad \tau \leq d < \tau + 0.5 \\ 1.5 & \text{if} \quad d \geq \tau + 0.5. \end{cases} \tag{4.4}$$

The cluster distance is a value indicating how far the pose is from the usual pose, in other words, how the pose is rare. Even within poses classified as rare, it is possible to learn with different weights for different samples.

4.2.4 Divide and Conquer Strategy for pose estimation (DACP)

We have proposed DRP, $ASRP$ and $WLCD$ to improve the performance of pose estimation models using rare poses. The three methods have been designed for an efficient learning of the rare pose. The proposed methods generally maintain the performance of the usual pose, but some experiments have also shown results sacrificing the performance of the usual pose for the boosted performance of the rare pose, which is not a significant drop comparing to the performance gain in rare pose.

Thus, we have adopted the *divide and conquer* strategy to the network structure. The divide and conquer is an algorithm which recursively breaks down a problem into two or more sub-problems. To resolve the tradeoff between the performance of rare pose and usual pose at the same time, we divide our pose estimation architecture into two networks each of which focuses more on the rare pose or the others. The proposed algorithm works as below.

Algorithm 1 uses two networks in parallel: The Net_r is learned by the proposed methods (DRP+ $ASRP$ + $WLCD$) for boosting the performance of the rare pose and Net_b is the baseline network for retaining the performance on the usual pose. We calculate the confidence scores with the output Heatmaps (h_b, h_r) of each network. The Confidence score is the mean of max values of Heatmaps extracted from all parts. Between Net_b and Net_r, the one with larger confidence score is selected as the final prediction.

4.3 Experiments

Earlier in this paper, we have newly defined rare poses and proposed three strategic methods to improve the performance on the rare poses. MPII and COCO

Algorithm 1 Divide and conquer for pose estimation

```
$Net_b$ : baseline network.
$Net_r$ : baseline network with proposed methods
$score$ calculates a confidence score of heatmaps.
$postprocessing$ : detects the $(x, y)$ locations of pose from the heatmap.
for $i = 1, \cdots, N$ do
    $Img_i$ is an input image.
    Obtain $h_b$ from $Net_b(Img_i)$.
    Obtain $h_r$ from $Net_r(Img_i)$.
    if $score(h_b) < score(h_r)$ then
        return $postprocessing(h_r)$
    else
        return $postprocessing(h_b)$
    end if
end for
```

keypoints datasets are used in this section for performance evaluation of the proposed methods.

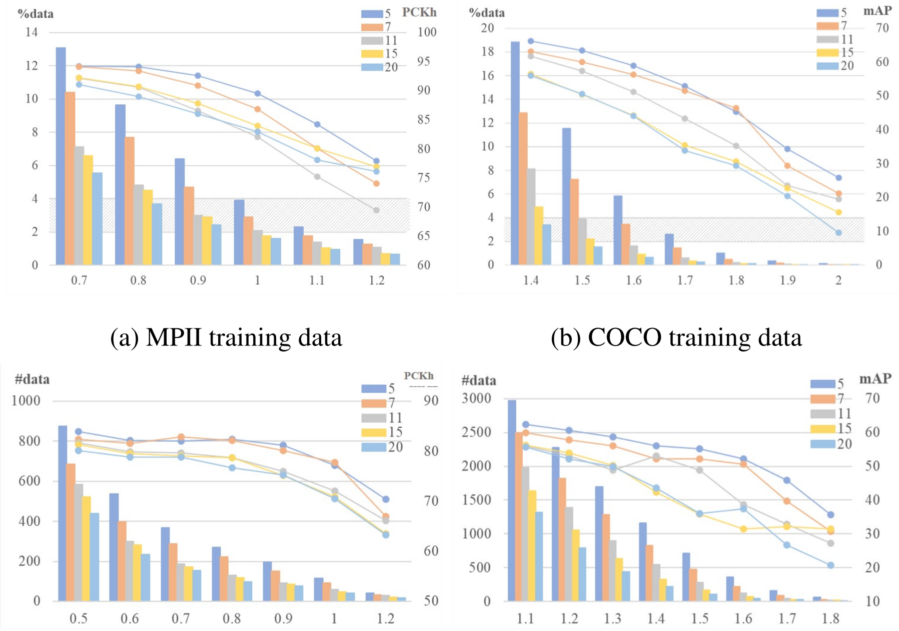
For the results of clustering, the location coordinates (x, y) of poses are normalized and used as the input feature values for clustering because the location information can classify the data regardless of the texture of the image. So, coordinates of 16 parts of MPII are used as a 32 dimensional feature vector and those of 17 parts of COCO are used as a 34 dimensional feature vector for clustering.

In order to show the effectiveness of our proposed method for performance enhancement on rare pose samples, we have set Simple[80] and CPN [13] as our baseline models. Both methods are top-down methods, and the basic structure of both methods is widely used in human pose estimation. We have conserved the network structure, hyper-parameters and the training criteria of the baseline reported except that a different batch size is used for our implementation due to our given computational resource. We use ground-truth bounding box labels of people to exclude the possibility of differences in performance caused by using an external object detector. All of the ground-truth Heatmaps are generated only using visible parts. In case of Simple[80], we adopt ResNet-50 network and input image resolution of (256,192) for COCO and (256,256) for MPII. We use data augmentations such as rescaling(±30%), rotation(±40 degrees) and flip. In case of CPN [13], we adopt the input image resolution of (256,192) for COCO and MPII. Similarly, data augmentations include rescaling(0.75∼1.35), rotation(±45 degrees) and flip. In MPII and COCO, hard samples such as self-occluded poses can be frequently observed. Training the generator in ASRP, those samples participate in the training and thus the generator is able to produce challenging samples.

4.3.1 Results of Rare Pose Identification

Since *K*-means clustering is an algorithm that collects similar data by using differences among features based on *K* centers, its cluster classification results vary greatly depending on the number of *K*.

Fig 4.5 shows the results of experiments using different *K*-means clustering with different number of clusters K for MPII and COCO. Figs 4.5 (a) and (b) show the experiment results on the training sets for each dataset, and (c) and (d) show the results of experiments with each validation data on the Simple[80] baseline model. Experiments are performed by changing the number of clusters from 5 to 20 for each dataset. The x-axis represents various distance thresholds



(a) MPII training data

(b) COCO training data

(c) MPII validation data

(d) COCO validation data

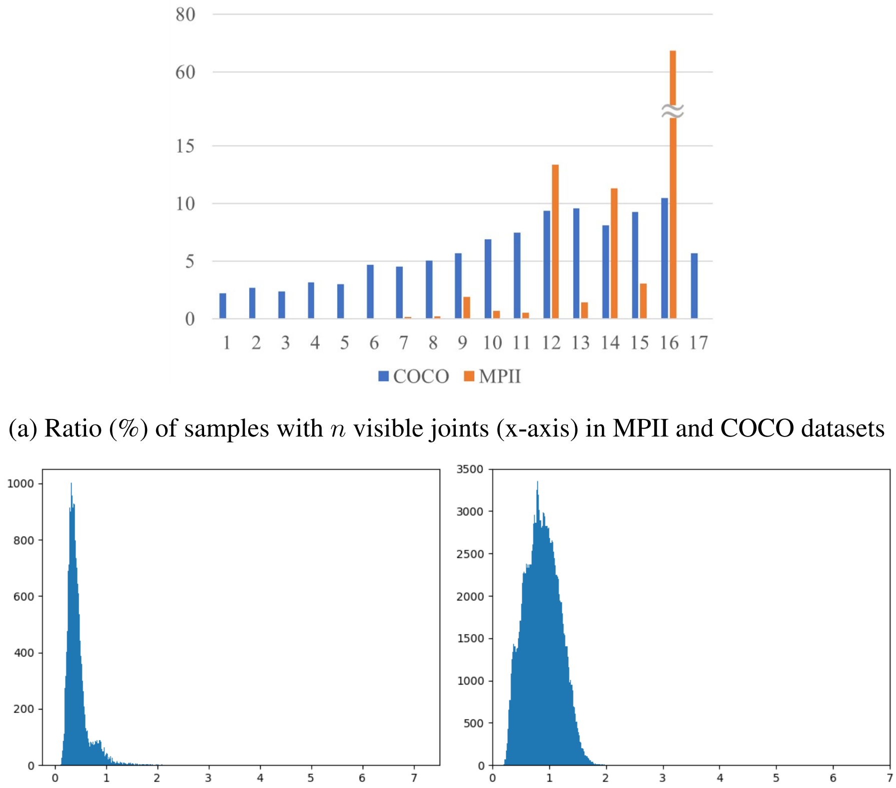
Figure 4.5: The pose estimation results for rare pose samples whose cluster distance d_i^c exceeds each threshold τ (x-axis) using Simple [80] baseline model. The numbers 5 to 20 in the upper right of the graph represent numbers of clusters K. The bar graphs in (a) and (b) indicate the percentage of corresponding rare samples from the total number of pose samples (%data) while the lines represent the accuracy scores. We chose hyper-parameters so that the rare poses would occupy as 2-3 %, the range of which is indicated by the gray area. In (c) and (d), bars of validation data indicate the numbers of corresponding rare samples (#data) while the lines are the accuracy.

τ, and the y-axis represents the resultant accuracy values for poses larger than each threshold. We also included the number of corresponding samples in the graphs. In both datasets, all clusters show a decrease in accuracy as distance threshold increases. We have chosen a relatively large number of clusters to avoid the risk of clustering to focus on a few rare poses.

We thus have selected the number of clusters to be intuitively large which also yields gradual decrements of accuracy score for a fixed threshold τ as the number of clusters increases. It is experimentally considered suitable that about 2-4% of whole dataset should be set as rare poses, which is represented as gray areas in Fig 4.5(a) and (b). In the COCO case, the bar graphs in the gray section are $\tau = 1.4$ for cluster 20 / $\tau = 1.5$ for cluster 11, 15 / $\tau = 1.6$ for cluster 7 / $\tau = 1.7$ to cluster 5. Clusters 15 and 20 had low mAP with the same number of data as cluster 5, 7, and 11. This means that rare poses are not well classified as outliers when the number of clusters is too small because of the characteristics of COCO data which have many occlusions including self-occlusion. For this reason, we chose 15 clusters with $\tau = 1.5$ for COCO dataset. In the case of MPII, cluster was selected based on the same criteria. We chose 7 clusters because there were many visible parts comparing with COCO. The corresponding threshold was set $\tau = 1.0$. Finally, the values of K for MPII and COCO are respectively determined as 7 and 15 through experiments.

Table4.1 shows the results with various numbers for clusters. In the tables, the row '#data' represents the number of samples with larger distance than a threshold τ. An exception is the second column with '$< \tau$' which tells the number of non-rare samples whose distance is smaller than the threshold τ. In this context, each pose sample means one pose within a ground truth bounding box, and we had excluded COCO samples that have zero visible annotations from this experiment.

Fig 4.6(b) and (c) are histograms of the distance values from the cluster centers to each data respectively for MPII and COCO. In the case of MPII, most



(a) Ratio (%) of samples with n visible joints (x-axis) in MPII and COCO datasets

(b) Histogram of d_i^c for MPII

(c) Histogram of d_i^c for COCO

Figure 4.6: (a) shows the ratio of number of samples with certain number of visible parts out of the whole data. Subfigures (b) and (c) represent histograms of the distance values from the cluster centers to each data respectively for MPII and COCO.

data lie within cluster distances and distributed in a narrower graph width, and values tend to be biased on certain distance. On the other hand, the histogram for COCO tends to have a larger variance in cluster distance than MPII. This is because the COCO data is comparatively much larger than that of MPII with much more diverse poses, and in MPII, there are more cases where all body parts are visible than COCO. Fig 4.6(a) shows the number of data according to the number of visible parts. Orange is the result of MPII and blue is the result of COCO. The x-axis represents the number of visible parts and the y-axis represents the percentage of the samples. For the case of MPII data of which 16 parts to be annotated if fully visible, all joints from most of its data samples (67%) are visible. Fully visible COCO data samples are defined by 17 part locations, and there are not many fully visible pose samples. In the case of MPII, from a total of 16 joint locations annotated, most data samples are annotated visible with an average of 12 or more visible parts. On the other hand, in the case of COCO, an average of 6 parts or less is visible among the 17 available parts.

We provide the results in Table 4.1 to show tendency of labeling rare poses according to certain thresholds. Each table shows the number and accuracy of train data by τ. Also, Fig 4.1(b) shows examples of rare and none-rare poses. Images that are detected as non-rare pose can be confirmed that the object has less active movements of parts with more frontal views than the ones detected as rare poses. From these results, it was confirmed that the higher the thresholds are, the lower the accuracy is with more peculiar poses are defined. Through these experiments, we have determined a reasonable thresholds (τ) 1.0 and 1.5 respectively for MPII and COCO, having a reasonable amount of data classified into rare poses with a low mAP. The 'Simple'[80] baseline model is used to select the number of clusters and the threshold of rare pose. In the other baseline model 'CPN'[13], the number of clusters and the threshold τ are set to be the same as for 'Simple'.

Table 4.2: The accuracy for each distance threshold (τ) on the MPII validation

(a) Results of Simple[80]

	All(0)	<0.5	0.5	0.7	0.9	1	1.2
#data	2958	2274	684	289	150	91	34
Basline	88.53	90.33	82.4	82.9	80.18	77.81	66.98
DRP	88.6(+0.07)	90.24(-0.08)	82.99(+0.59)	83.71(+0.81)	82.06(+1.88)	79.27(+1.46)	73.07(**+6.09**)
ASRP	88.81(+0.28)	90.62(+0.29)	82.63(+0.23)	83.45(+0.55)	80.18(+0)	77.81(+0)	67.62(**+0.64**)
ASRPT	88.68(+0.15)	90.48(+0.15)	82.52(+0.12)	83.69(+0.79)	80.89(+0.71)	78.64(+0.83)	70.83(**+3.85**)
WLCD	88.84(+0.31)	90.46(+0.13)	83.33(+0.93)	83.8(+0.9)	81(+0.82)	78.43(+0.62)	68.91(**+1.98**)
DRP + WLCD	88.60(+0.07)	90.16(-0.17)	83.29(+0.89)	84.06(+1.16)	81.71(+1.53)	79.58(+1.77)	70.83(**+3.85**)
ASRPT + WLCD	88.54(+0.01)	90.29(-0.03)	82.56(+0.16)	82.79(-0.1)	80.71(+0.53)	78.22(+0.41)	69.872(**+2.89**)
DRP+ASRP+WLCD	88.431(-0.09)	90.267(-0.06)	82.17(-0.23)	83.66(+0.76)	82.23(+2.05)	80.10(+2.29)	74.67(**+7.69**)
DACP	88.69(+0.16)	90.47(+0.14)	82.60(+0.20)	83.08(+0.18)	80.71(+0.53)	78.33(+0.52)	68.91(**+1.93**)

(b) Results of CPN[13]

	All(0)	<0.5	0.5	0.7	0.9	1	1.2
#data	2958	2274	684	289	150	91	34
Basline	85.34	88.52	74.49	68.36	59.84	61.04	48.71
DRP	85.59(+0.25)	88.74(+0.22)	74.86(+0.37)	68.65(+0.29)	60.25(+0.41)	62.7(+1.66)	50.96(**+2.25**)
ASRP	85.67(+0.33)	88.76(+0.24)	75.14(+0.65)	68.73(+0.37)	60.31(+0.47)	61.14(+0.1)	48.07(-0.64)
ASRPT	86.13(+0.79)	89.12(+0.6)	75.92(+1.43)	69.63(+1.27)	61.01(+1.17)	61.35(+0.31)	48.39(-0.32)
WLCD	85.79(+0.45)	89(+0.48)	74.87(+0.38)	69.69(+1.33)	62.48(+2.64)	62.29(+1.25)	51.92(**+3.21**)
DRP + WLCD	86.21(+0.87)	89.3(+0.78)	75.68(+1.19)	70.73(+2.37)	63.3(+3.46)	64.16(+3.12)	51.92(**+3.21**)
ASRPT + WLCD	85.1(-0.24)	88.92(+0.4)	75.35(+0.86)	69.89(+1.53)	61.43(+1.59)	62.48(+1.44)	46.47(-2.24)
DRP+ASRP+WLCD	85.15(-0.19)	88.17(-0.35)	74.89(+0.4)	68.13(-0.23)	59.2(-0.64)	61.25(+0.21)	49.35(**+0.64**)
DACP	85.7(+0.36)	88.70(+0.18)	75.48(+0.99)	68.76(+0.40)	59.61(-0.23)	61.14(+0.10)	49.00(**+0.29**)

Table 4.3: The accuracy for each distance threshold (τ) on the COCO 2017 validation

(a) Results with Simple[80]

	All(0)	<1.1	1.1	1.3	1.5	1.6	1.7	1.8
data	10777	9142	1635	636	170	69	28	14
Baseline	72.4	76.6	56.3	50.2	35.9	31.4	32.2	31.4
DRP	72.6(+0.2)	76.5(-0.1)	56.8(+0.5)	50(-0.2)	36.3(+0.4)	34.3(+2.9)	33.7(+1.5)	35.1(**+3.7**)
ASRP	71.9(-0.5)	76.2(-0.4)	55.7(-0.6)	49.2(-1.0)	35.7(-0.2)	32.7(+1.3)	35.7(+3.5)	39.8(**+8.4**)
WLCD	72.7(+0.3)	76.5(-0.1)	57(+0.7)	50.9(+0.7)	35.3(-0.6)	31.9(+0.5)	32.2(0)	34.7(**+3.3**)
DRP + WLCD	72.8(+0.4)	76.8(+0.2)	57.1(+0.8)	50.7(+0.5)	37.5(+1.6)	33.5(+2.1)	33.9(+1.7)	37.7(**+6.3**)
ASRP + WLCD	72.6(+0.2)	76.5(-0.1)	56.8(+0.5)	50.2(+0)	36.2(+0.3)	32.4(+1)	34.8(+2.6)	40.1(**+8.7**)
DRP+ASRP+WLCD	72.6(+0.2)	76.8(+0.2)	56.6(+0.3)	50(-0.2)	37.1(+1.2)	34.1(+2.7)	37.1(+4.9)	39.7(**+8.3**)
DACP	72.8(+0.4)	77.1(+0.5)	56.6(+0.3)	50.3(+0.1)	36.8(+0.9)	34.0(+2.6)	37.1(+4.9)	38.1(**+6.7**)

(b) Results with CPN[13]

	All(0)	<1.1	1.1	1.3	1.5	1.6	1.7	1.8
#data	10777	9142	1635	636	170	69	28	14
Baseline	71.2	75.6	54.8	48.2	32.8	28.4	29	26.2
DRP	71.2(+0)	75.7(+0.1)	54.7(-0.1)	48.9(+0.7)	34.2(+1.4)	31.2(+2.8)	29.3(+0.3)	32.6(**+6.4**)
ASRP	71.1(-0.1)	75.5(-0.1)	54.9(+0.1)	48.1(-0.1)	34.9(+2.1)	31.7(+3.3)	33.3(+4.3)	36.1(**+9.9**)
WLCD	71.3(+0.1)	75.6(+0)	54.9(+0.1)	48.5(+0.3)	35.2(+2.4)	31.4(+3.0)	30.3(+1.3)	32.9(**+6.7**)
DRP + WLCD	71(-0.2)	75.3(-0.3)	54.8(+0)	48.1(-0.1)	35.9(+3.1)	32.5(+4.1)	33(+4)	35.1(**+8.9**)
ASRP + WLCD	71.2(+0)	75.8(+0.2)	54.4(-0.4)	47.6(-0.6)	34.7(+1.9)	31.9(+3.5)	33.7(+4.7)	35.4(**+9.2**)
DRP+ASRP+WLCD	71.1(-0.1)	75.5(-0.1)	55(+0.2)	48.9(+0.7)	35.4(+2.6)	31.5(+3.1)	35.5(+6.5)	39.7(**+13.5**)
DACP	71.4(+0.2)	75.8(+0.2)	55.3(+0.5)	48.9(+0.7)	34.4(+1.6)	30.2(+1.8)	33.9(+4.9)	34.5(**+8.3**)

4.3.2 Results of Proposed Methods

The proposed methods are divided into methods with and without additional data. The methods of adding data (duplication of rare poses and addition of synthetic rare poses) are labeled as DRP and $ASRP$, respectively. For the DRP case, MPII has 644 poses and COCO has 3317 poses repeated within the training set. While $ASRP$ is a way to add a newly generated synthetic image, for a fair comparison against other proposing methods, $ASRP$ method creates and adds the same number of rare poses as DRP's added samples. During the process of $ASRP$, we can obtain re-calibrated pose annotations from the SMPL model. Based on the given annotations, we can calculate the bounding box coordinates and so on. $ASRPT$ represents the method of $ASRP$ with samples that are transferred from synthetic to real. Lastly, the method that does not alter the training set (weighted loss based on cluster distance) is referred as $WLCD$.

Tables 4.2, and 4.3 show the comparison results of the baseline models and our proposed methods on MPII and COCO datasets. The values in the table represents accuracy, and the value in brackets means the difference from the performance of the baseline model. The proposed methods are only used at a training time. At inference, pose is detected on a Heatmap which is generated from trained network.

At the MPII results in Table 4.2, the overall results mostly increases as the highest as 0.79. τ=1.0 assigned to rare pose in gray background, all the proposed methods show increases in performance. Especially, at τ=1.2 where cluster distance is relatively very high, the largetst increment is 6.09. We also show an increasing tendency with $\tau < 0.5$, which only covers usual poses, indicating that the proposed method does not get hindered from learning usual poses. The performance of $ASRPT$ is higher than that of $ASRP$ in estimating rare poses, which indicates that matching the style (real) with the training data helps improve performance. It is possible to further improve the rare pose performance when experimenting with improving the transfer performance in future research. Unfortunately, the methods of adding synthetic rare pose data showed poor performance as shown in Table 4.2 (b) at τ=1.2. However, in COCO data, when the synthetic was added, the performance was improved in Table 4.3 (b) τ=1.5 and 1.8, and even when the baseline network was the Simple[80] model, the performance was increased. It can be expected that adding the synthetic data is not a problem and the proposed methods must be adapted to the network model and data.

Table 4.3 shows the results of the experiments with COCO 2017 validation set. Compared to total mAP, the proposed methods increased by about 0.1-0.3 over the baseline method except for $ASRP$, where some decrease of performance is observed. The τ=1.5 assigned to rare pose in gray background. At the rare pose, all of the values were increased except for two methods. Furthermore, the all cases of upper τ=1.6 tend to generally increase the performances of the proposed method. Especially, the highest accuracy improvement is 13.5. Unfortunately, several methods where τ¡1.1 tends to have performance diminution, but the difference is 0.1 which is not large.

DRP and $ASRP$ are methods of data augmentation, and $WLCD$ is a method to give weight to loss. It is more effect to use the method of increasing data and weight loss at the same time to improve the rare pose. Experiments were performed on the combination of $DRP + WLCD$ and $ASRP + WLCD$ from COCO and MPII data. $ASRP + WLCD$ showed lower results than $DRP + WLCD$ combination, but $DRP + WLCD$ combination outperformed the method used alone. Especially, the $DRP + WLCD$ in Table 4.2 (b) showed the highest performance for all τ when compared with others. Combining all the proposed method ($DRP + ASRP + WLCD$), we have improved the performance on rare pose for both MPII and COCO datasets compared to both baselines (CPN and Simple).

$DACP$ means the result of an experiment applying the divide and conquer method. $DACP$ shows meaningful improvement in both rare pose and usual pose under all experimental settings. In rare pose, the accuracy of $DACP$ is less than $DRP + ASRP + WLCD$, but still higher than baseline. $DACP$ generally shows improvement in any pose.

We have proposed methods of defining a rare pose and improving performance for rare pose. In some methods, there has been a slight performance drop in usual pose due to the trade-off between usual pose and rare pose. Though it is a tolerable amount of degradation, we can still resolve this issue with $DACP$, sacrificing the inference time.

Fig 4.7 shows results of our methods and a baseline model[80] on one of the samples of MPII validation and COCO 2017 validation. While the baseline model struggles estimating 2D pose as Fig 4.7 shows, our proposing techniques better performs against the rare pose sample.

In this paper, we have evaluated our method on test images only, numbering 1000 images, to check the effect of the proposed methods in a different domain. Table 4.4 is the results of experiments on the Leeds sports pose dataset (LSP) test. The evaluation was performed using the trained Simple model [80] on COCO keypoint data without learning with LSP. In $ndata$, All (1000 poses) means all of the validation data, and selected (44 poses) means the dataset that we chose rare pose in the validation data. The results were measured by Percent-

Table 4.4: The comparison accuracy on Leeds Sports Pose validation dataset (LSP) using Simple[80] model trained on COCO.

ndata	Parts	Ankle	Knee	Hip	Wrist	Elbow	Shoulder	Head	Mean
All	Baseline	91	92	90.6	83.5	86.9	92.3	23.2	79.9
	DRP	91.1(+0.1)	92.5(+0.5)	91(+0.4)	83.2(-0.3)	87.5(+0.6)	93.3(+1.0)	24(+0.8)	80.4(+0.5)
	ASRP	91.5(+0.5)	92.6(+0.6)	90.6(+0)	82.9(-0.6)	86.7(-0.2)	92.2(-0.1)	25(+1.8)	80.2(+0.3)
	WLCD	91.5(+0.5)	92(+0)	91.3(+0.7)	83.4(-0.1)	87.3(+0.4)	92.6(+0.3)	23.4(+0.2)	80.2(+0.3)
Selected	Baseline	36.4	34.1	44.3	40.9	53.4	56.8	27.3	41.9
	DRP	46.6(+10.2)	43.2(+9.1)	46.6(+2.3)	51.1(+10.2)	53.4(+0)	61.4(+4.6)	25(-2.3)	46.8(+4.9)
	ASRP	44.3(+7.9)	39.8(+5.7)	38.6(-5.7)	37.5(-3.4)	47.7(-5.7)	54.5(+2.3)	28.4(+1.1)	41.6(-0.3)
	WLCD	44.3(+7.9)	39.8(+5.7)	47.7(+3.4)	44.3(+3.4)	55.7(+2.3)	59.1(+2.3)	28.4(+1.1)	45.6(+3.7)

age of Correct Keypoint (PCK). In the case of Head, because COCO annotation were different with LSP, they were excluded from the comparison. All PCK increased with the average PCK ($Mean$) except $ASRP$ of selected data. Among the proposed methods, $WLCD$ showed an increasing trend from all parts. This is because the other two methods were data augmentation with the existing data domain, so there is a domain specific point. So, $WLCD$ is more robust to domain transfer.

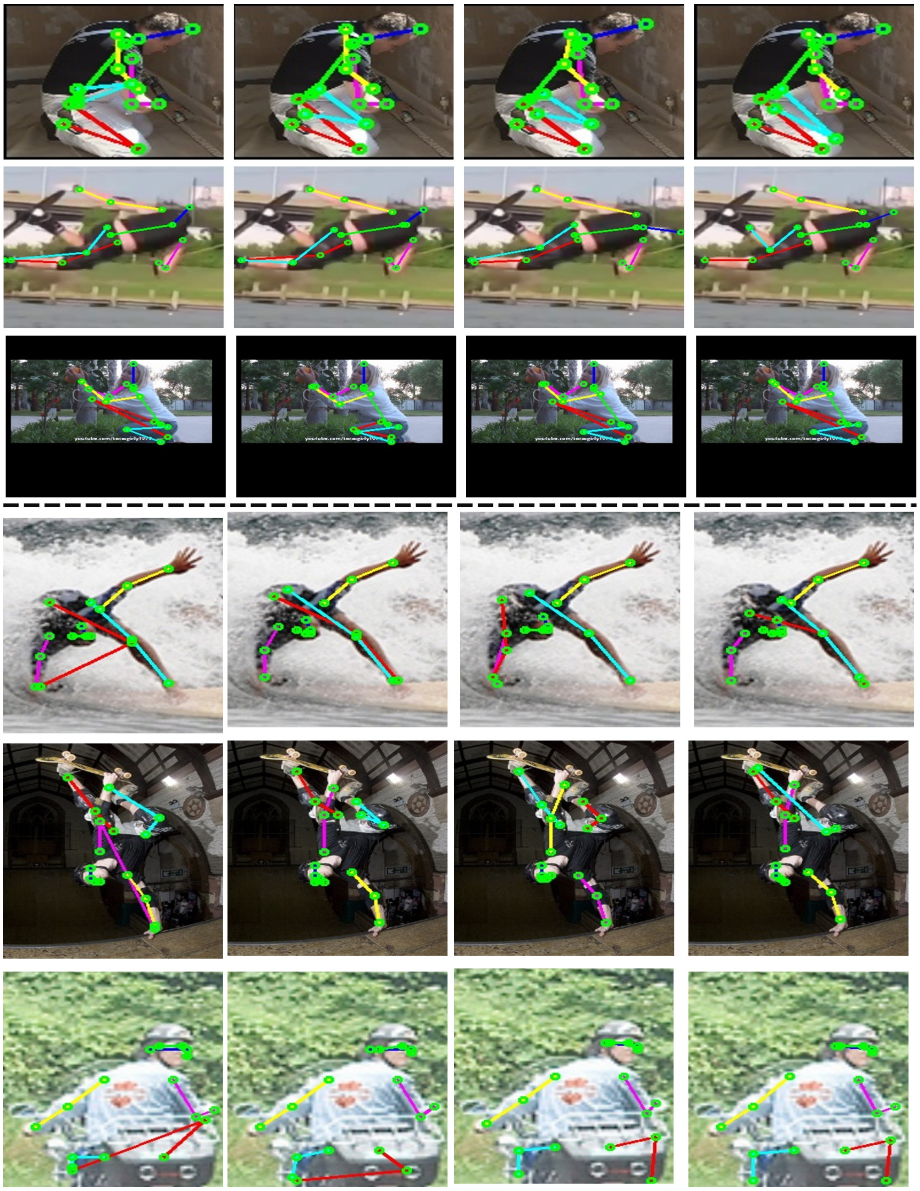


Figure 4.7: The qualitative results of the proposed methods and baseline in MPII dataset and COCO dataset.

Based on the dotted line, MPII results are above and COCO results are below.

From left to right, baseline[80], DRP, ASRP and WLCD.

Chapter 5

Multi-person pose estimation and tracking

In this chapter, we propose a method to track the pose using temporal information. It is essential to exploit the temporal information for tracking. There are many different ways to use temporal information, such as maps and vectors. For example, the optical flow is a usual method to calculate the velocities of the object. Existing research has applied the optical flow to track the pose by calculating the amount of change.

We need the vector map to denote the motion of the part to track the part. Thus, we proposed the unit vector map about the movement of limbs. The vector map is called a temporal flow map for limbs (TML). The TML is generated using the variation of limbs between two frames as shown in Figure 5.3.

We have designed a single-network to estimate and track human poses using spatial and temporal features. The single-network has two sub-parts: Spatial part and Temporal part, as shown in Figure 5.1. The spatial part has the same structure with [10] which is one of the most popular networks in the bottom-up approach of multi-person human pose estimation. The spatial part has iterative stages. The stages have two branches to learn the part Heatmaps and Part Affin-

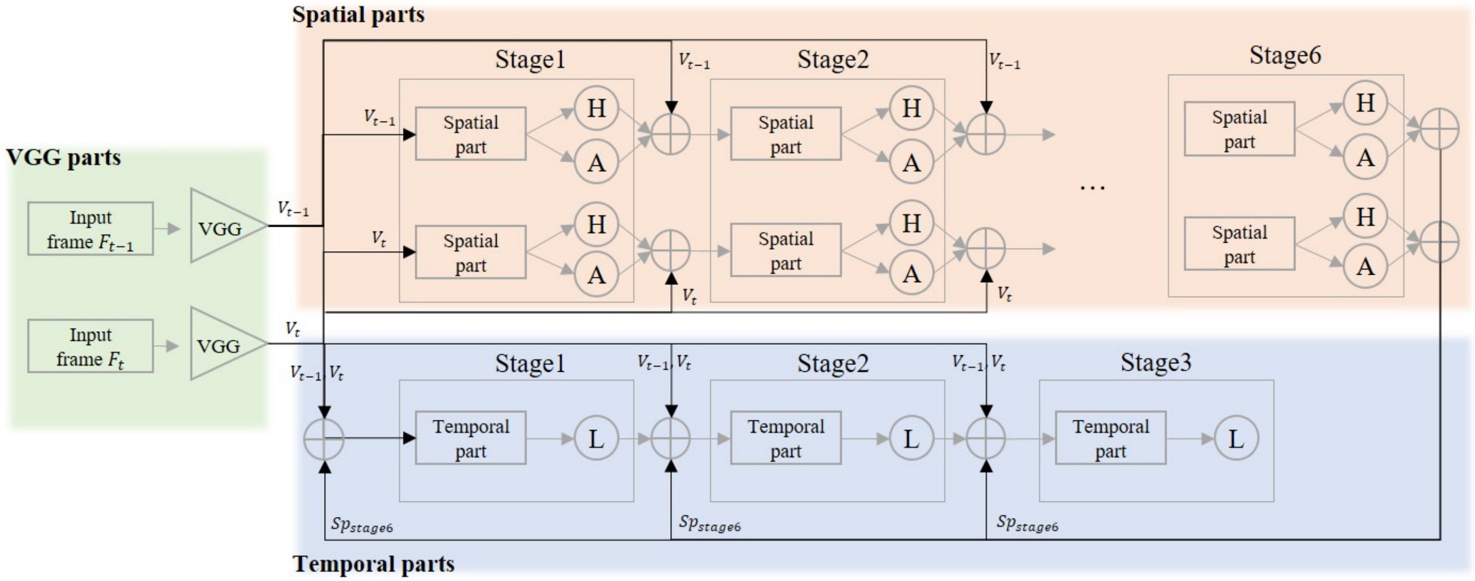


Figure 5.1: The structure of the spatial-temporal network.

The spatial and the temporal parts are combined together in a single network. On the spatial part, joint heatmaps (H circle) and part affinity fields (A circle) are regressed. Outputs from the spatial part and features from the final layers of the VGG parts are fed into the temporal part. The temporal part regresses the TML (L circle). Pixel-wise L2 losses are used to optimize all the outputs. V_{t-1} and V_t mean the extracted features from VGG parts at $t-1$ and t frame respectively. Sp_{stage6} is the concatenated spatial features of $t-1$ frame and t frame at the stage6.

ity fields. Likely the iterative stage of the spatial part, the temporal part has the iterative stage with a single branch to learn the TML.

For training, two frames are taken for inputs, and the spatial and the temporal information affect each other through the end-to-end learning. At inference time, three frames are fed into the spatial part to detect poses, and pairs of frames are fed into the temporal part to detect the association of poses between two frames.

Below is a more detailed description on each part.

- VGG part: VGG part means the VGG network [69]. VGG part performs to extract the features from the input image. Extracted features are fed into each parts such as spatial parts and temporal parts. At the training time, VGG part also is trained as end-to-end learning.

- Spatial parts: The spatial parts has consisted of the iterative structure in which the same stages are stacked. Each stage has two branches to learn Heatmaps (H circle in Figure 5.1) and part affinity maps (A circle in Figure 5.1). Each stage has three 3×3 convolutions and two 1×1 convolutions. The number of stage is six and the loss function is pixel-wise L2 loss function as in [10].

- Temporal parts: the temporal parts have a resemble structure with spatial parts. The structure is consisted three stages which have the same convolution layer with spatial network. Each stage has one branches to learn TML(L circle in Figure 5.1). The first stage takes the features which concatenate with VGG and the last layer of spatial parts. Other stages takes the features of previous stage. Loss function is a pixel-wise L2 loss function at each stage.

As the spatial parts are the same network presented in [10], we will focus on the temporal part in the following subsections.

We have a unique way of inference as shown in Figure 5.2. Our tracking way is the off-line approach. Three frames are taken as a frame set at a time. The poses are extracted on the each of three frames using the joint Heatmaps and part affinity fields at the spatial parts. The pair of two frames is fed into the temporal parts and get the TML. The first pair is a first frame and second frame. We calculate an associated score from a TML score and a joint distance. Based on the associated score, the poses are tracked. The second frame and third frame are the second pair. The pair also is applied as same procedure.

After tracking the poses in three frames, we refine the missing pose in the middle frame because of the blurring or occlusion. The missing poses in second frame are filled by Analyzing the association scores of between first and third frames. This makes the model stable since the information from the frames back and forth adjust the result of the intermediate frame.

5.1 Temporal flow Maps for Limb movement (TML)

The TML is a vector map representing the movement of a person's limbs. The limb means a part linking two joints such as a knee and an ankle. Figure 5.3(b) shows the visualization of TML of the bottom of the left arm, which links the elbow and wrist. For generating the vector map, each limb is divided at regular intervals as shown in Figure 5.3(a). In the Figure 5.3(a), we can see that the left hand of the person moves to the left-down side and that the left hand of the person moves to the left-down side. We divide the limbs on each F_{t_1} and F_{t_2} frames. The red circles in Figure 5.3(a) are joints, and the red lines linked be-

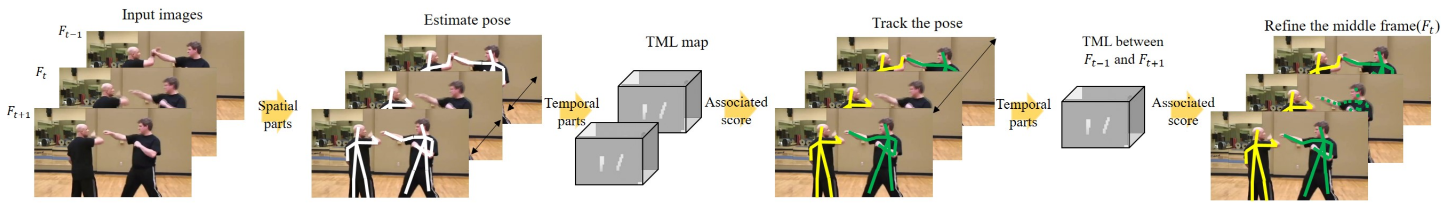


Figure 5.2: An inference flow of spatial-temporal network method. A set of frames (F_{t-1}, F_t, F_{t+1}) is defined for temporal inference. Two frames that are (F_{t-1}, F_t) or (F_t, F_{t+1}) are input into the network as a pair. First, poses are estimated using the spatial part on the each frame and TML are extracted by the temporal part at the same time. To track poses, we calculate the association score of each person using the TML and the joint distance score. The optimal connection is found by using a bipartite method. In order to refine the middle of frame F_t, we need to get the associated information between (F_{t-1}, F_{t+1}) through the TML and the joint distance. If the pose is connected between F_{t-1} and F_{t+1}, we added the average pose of between (F_{t-1}, F_{t+1}). Note that, the time interval of TML is 1 at inference stage but it can be a greater number at training stage which will be described in the multi-stride method.

tween two joints are limbs. The same divided parts on the limb between frames have calculated the unit vector to make the TML. More specifically, a separated part ($S_{t,p,l,n}$), which means an n-th separated part on l-th limb on the p-th person at the frame t, is used to calculate the movement direction between two frames. Based on the pair $(S_{t_1,p,l,n}, S_{t_2,p,l,n})$, we calculate a unit vector v as follows:

$$v = \frac{(S_{t_1,p,l,n} - S_{t_2,p,l,n})}{\|S_{t_1,p,l,n} - S_{t_2,p,l,n}\|_2}. \tag{5.1}$$

Here, n, l and p represent the index of a separated part, a limb and a person respectively, and t_1 and t_2 are the frame indices. The part S is represented by a two dimensional vector corresponding to the position of the part and thus v is also a two-dimensional vector.

Then, the L for the l-th limb is encoded through the unit vector v for each pixel $s = (x, y)$ which is the limb passes through at the time interval t_1 and t_2. To draw the TML, we applied the similar process of part affinity field in [10].

$$L_{l,p}(s) = \begin{cases} v & \text{if } s \subseteq C \\ 0 & \text{otherwise.} \end{cases} \tag{5.2}$$

According to the condition (C), each pixel is determined to whether it is on the path of limb movement at the time interval t_1 and t_2. More concretely, in our case, the pixels belonging to the line segment ($S_{t_1,p,l,n}$, $S_{t_2,p,l,n}$) with a constant width is filled with the value of v and the other pixels remain as zero.

When the TML of multi person are overlapped at the same position, it is averaged to preserve the scales of the output. Thus, the final TML for the l-th joint averages the TML of the joints of all people appeared in the image as follows:

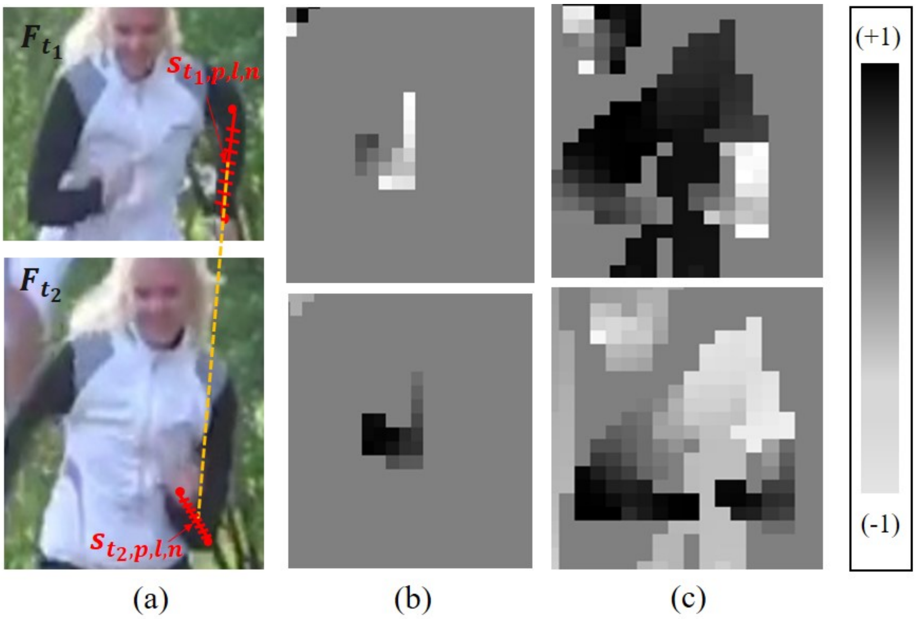


Figure 5.3: An example of TML.

(a) Illustration explaining how to obtain the TML using the frames F_{t_1} and F_{t_2}. We subdivide each limb into several parts and calculate the unit vector of each pair (connected by the yellow lines, $S_{t_1,p,l,n}$ and $S_{t_2,p,l,n}$). (b) Visualization of the left arm TML on x(top) and y(bottom) coordinates. (c) Accumulated TML for all limbs on x(top) and y(bottom) coordinates. The values of TML are between the range of -1 and 1.

$$L_l(s) = \begin{cases} \frac{1}{P(s)} \sum_{p=1}^{P(s)} L_{l,p}(s), & \text{if } P(s) \geq 1 \\ 0 & \text{if } P(s) = 0, \end{cases} \tag{5.3}$$

where $P(s)$ means the number of non-zero vectors at pixel s. All of n divided parts follow the above process to make the TML.

Unlike optical flow [25] representing directions and magnitudes at each location, the TML only represents the directions using unit vectors. Because the

TML does not contain magnitude information, it is more prone to change of time interval between frames. The multi-stride method for data augmentation, which will be describe in the next subsection, helps to alleviate this issue and successfully trains the network using video frames with different sampling rates.

Furthermore, the TML channel can be set as an individual channel for each limb (Figure 5.3(b)) or as an accumulated channel which accumulates the TML of all limbs (Figure 5.3(c)). The number of individual channels becomes *the number of limbs* $\times$ *2* (x, y coordinate channel) while the accumulated channel has only 2 channels (x, y coordinate channel). We will show the efficiency of different types of channel in the evaluation section.

5.2 Multi-stride method

The TML has explained the flow of temporal movement. To generate the TML, we use two frames and basically the time interval of two frames is one. Unfortunately, using only the time interval of one in video sequential has limited types of TML because of the number of restricted training data. We need a various type of TML and a huge dataset for TML. Thus, we use the various time interval that means a multi-stride method. The multi-stride method take two frames within a given time range which in this paper set as five.

Figure 5.4 shows the examples of the TML on five time range. Figure 5.4(a) has the one time interval. Because the frames are generally gathered at 30 frame per second, the motion of the one time interval is too small and not diversity. It is better to train the various motion to track the pose. Thus, we use the various time intervals as shown in Figure 5.4(b) and (c). The bigger time interval has

Furthermore, our multi-stride method can be used to refine poses at infer-

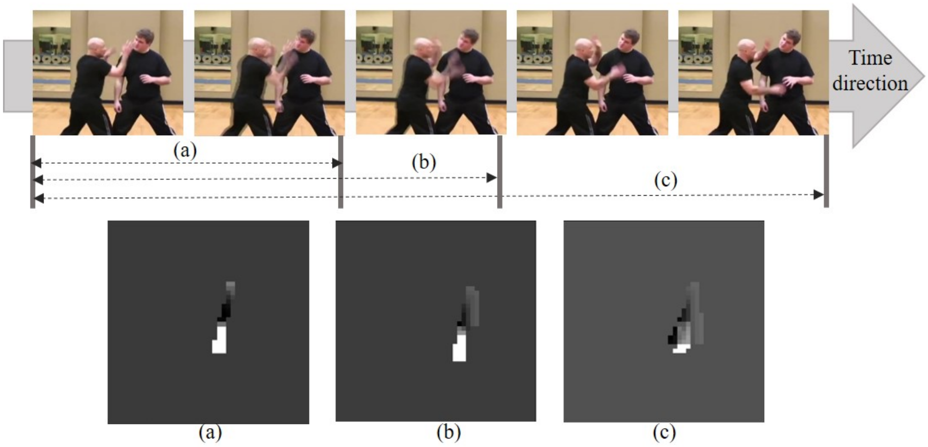


Figure 5.4: Examples of the TML of x coordinate with various time intervals. Consecutive image sequences are shown from left to right. (a), (b) and (c) are the right arm TML of the left person with the different time intervals, 1, 2 and 4 respectively. Using various strides, it is possible to get the TML of both small and large movements.

ence time. The proposed refining method can be useful when a frame misses a person but the preceding and the next frames successfully target the person. In this case, because our multi-stride method randomly selected two frames at the training time and the network learned this situation, we can extract the TML and track the pose between frame F_t and F_{t+2}.

5.3 Inference

In this section, the off-line inference approach is proposed to track poses. The proposed network is structured to take the two frames for input image and to generate the spatial maps and temporal map at the same time. Thorough the outputs of network, the poses are tracked. By extension, we pursuit to refine the

Table 5.1: The estimation and tracking results of the proposed methods on the PoseTrack2017 and 2018 validation data.

#stage means the number of stacked stages in the temporal part. Joint-Flow has a different type of temporal map that is created by joint movement. Basically, the proposed TML has two channels (x and y) for each limb. ($*$) means the method in which the TML of all limbs are accumulated in a single map for x and y directions. $+$ adopted the non-maximum suppression (NMS) for joints. $++$ indicates that the proposed refining method for the middle frame pose is applied. Distance means that The TML is not used in the calculation of the association score in (5.4) by setting α to 0, which means that it only uses the torso distance of a person for the associated score.

data	Method		MOTA								mAP
		#stage	Head	Shou	Elb	Wri	Hip	Knee	Ankl	Total	
2017	Joint-Flow($*$)	1	70.6	70.1	50.6	37.5	53.9	41.8	30.3	52	73.1
	Joint-Flow	1	48.5	48.3	30.3	19.3	33.9	23	13.5	32.1	73.2
	TML($*$)	1	72	70.6	52.1	37.7	53.8	41.3	30.9	52.6	71.3
	TML	1	70.1	69.5	51.9	40.5	53.8	43.5	32.7	52.9	72.9
	TML	3	74.7	74.1	61.7	49.4	59	52.6	43.7	60.3	70.9
	TML+	3	75.1	74.6	62.5	50.1	59.5	53	44.2	60.9	71.3
	Distance+	3	49.9	50.1	40.5	31.5	37.7	32.4	26.7	39.2	71.3
	TML++	3	75.5	75.1	62.9	50.7	60	53.4	44.5	61.3	71.5
2018	TML++	3	76	76.9	66.1	56.4	65.1	61.6	52.4	65.7	74.6

pose using a relation of three frames (F_{t-1}, F_t, F_{t+1}) which are defined to a set of frames as shown in Figure 5.2. More specific inference approach is presented below.

On each frame, we estimate the candidates of parts using the extracted

Heatmaps. The candidates of parts are connected by calculating an score of part affinity fields as in [10]. At this time, the poses don't have an unique tracking id and they just are denoted as the poses (I). The Heatmaps and part affinity fields are created by the spatial part as shown in Figure 5.1.

For the tracking poses, we have proposed the method to calculate the associated score of poses between the front and back frames. The associated score is calculated by each person in different frames. For example, a pose $I_{a,F_{t-1}}$ in frame F_{t-1} and a pose I_{b,F_t} in frame F_t calculate the associated score using the TML and a distance. The associated score is calculated by a linear combination of a score of the TML (S_T) and a score of joint distance (S_d):

$$S = \alpha S_T + (1-\alpha) S_d, \tag{5.4}$$

where α is a hyper-parameter which is set to 0.5 in our experiments.

We measure the score of a candidate movement on each TML by calculating the line integral. More specifically, we extract two joint candidates $I_j^{t_1}$ and $I_j^{t_2}$ in different frames at time t_1 and t_2 corresponding to the joint j and make a normalized directional vector between the two joint candidates. Then the value of the TML corresponding to the line segment ($I_j^{t_1}$, $I_j^{t_2}$) is obtained to take inner product with the directional vector. This is done for all the points in the line segment and integrated as follows:

$$S_T = \frac{1}{n_J} \sum_{j=1}^{n_J} \int_{u=0}^{u=1} L_l(K(u)) \cdot \frac{I_j^{t_1} - I_j^{t_2}}{\left\| I_j^{t_1} - I_j^{t_2} \right\|_2} du. \tag{5.5}$$

Here, I is a joint candidate and n_J is the number of joints for a person which is determined in the spatial part. $K(u)$ indicates interpolated points in the line segment ($I_j^{t_1}$, $I_j^{t_2}$) where $u \in \{0, 1\}$, i.e., $K(u) = (1-u) \cdot I_j^{t_1} + u \cdot I_j^{t_2}$. This score measures the plausibility of joint association between frames using the TML.

We measured the joint distance (S_d) between the frames using the Euclidean distance.

$$S_d = \frac{1}{n_J} \sum_{j=1}^{n_J} \left\| I_j^{t_1} - I_j^{t_2} \right\| \tag{5.6}$$

Both scores are given a different weight by using the variable α which is determined through experiments. Finally, we find the optimal connection by applying a bipartite graph [16].

After tracking the poses between front and back frames in the set frames, we refine the poses of the middle frame. We want to improve the situations of the blur image, the occlusion and other reasons that could make the poses have disappeared in the intermediate frame. On the other hand, the disappeared pose comes out again in the first and third frames as shown in Figure 5.2. Specifically, the poses are not extracted on the frame F_t and extracted and tracked on the frame $(F_{t-1}$ and $F_{t+1})$.

To refine the poses in the intermediate frame, we use the association of three frames. The frame F_{t-1} and F_{t+1} are fed into the proposed network followed by the above tracking approach. After that, we could take the poses which have the unique track id. The track id of poses is compared with the pose's track id on the second frame, whether it has existed or not. If the pose hasn't existed, the poses or the joints missed in the middle of the frame F_t are filled with average locations of those in F_{t-1} and F_{t+1}.

5.4 Experiments

The task of tracking the pose has to experiment on the video or real-time camera. We have used the PoseTrack datasets with information of multi-person pose estimation and tracking based on the video data. The PoseTrack datasets have consisted of the PoseTrack2017 and Posetrack2018. Two datasets have different annotation types. The order of part is different and more parts such as ear are added to joint. For the test, mean average precision (mAP), multiple object tracker accuracy (MOTA) and multiple object tracking precision (MOTP) are evaluated in the annotation order of PoseTrack 2017.

Our proposed model based on the COCO keypoints dataset pre-trained model [10]. The network has various parameters: a weight decay is 0.005, a momentum is 0.9 and the learning rate is 0.00005. Efficiently using the pre-trained parameter at the training network, we have changed the part order and added non-existent parts. We used the Caffe open-source library [32].

For the data augmentation method, we have used random crop, random rotate and random scaling. Because our proposed network is trained as end-to-end approach, the two input frames have applied the same parameter of data augmentation. More specifically, on the first frame, the parameters of data augmentation methods are randomly decided and the next frame are equally applied.

MOTA, MOTP and mAP are used to evaluate the performance [54]. Table 5.1 shows the results of the proposed methods by different settings - using different numbers (1 or 3) of iterative stages in the temporal part (#stage), using channel accumulation of TML instead of using individual channels for each joint ($*$), and a tracking method only using distance score by setting α in (5.4) as 0 (distance). Through the experiment, we empirically decide the number of subdivide each limb to 20 pieces to make the TML.

To make the temporal network part having as few parameters as possible while maintaining high performance, we experimented with different number of repetition stages, 1, 3 and 6. The spatial part used a fixed six stages. Table 5.1 only compares the performances with one and three iterative stages in the temporal part, because the experimental result of the iterative 6 stages is lower than that of 3 stages and has a huge number of parameters.

Similar to optical flow [25], we accumulate all limb movements in one map called accumulated channel map as shown in Figure 5.3(c). On the Table 5.1, $(*)$ means that the network used the accumulated TML. Basically, we use a map with a channel for each limb called individual channel map. The number of channel on individual channel map is (the number of (x, y) channels = 2)×(the number of limbs), but the accumulated channel map has only two (x, y) channels. In all the tested networks, the accumulated channel map obtained lower accuracy than the individual channel map. Huge amount of the directional information of each limb is lost in the accumulated map, because the map includes some problems, e.g., different limbs overlap in the same location and have an averaging effect on that point.

We implemented and compared the performance of the Joint-Flow map to show that the map created using limbs is more efficient than the map created using joints. The Joint-Flow map is constructed as a direction in which the joint moves between two frames. The Joint-Flow map follows the equation of (5.2) but uses the joint location instead of separated part s.

The mAPs of Joint-Flow are higher than the TML, but MOTAs are lower. This results shows the difficulty of tracking using the Joint-Flow, because the

Table 5.2: Pose estimation and tracking performance on PoseTrack 2017 test dataset.

Method		mAP	MOTA	MOTP	Prec.	Rec
Top-down	Poseflow[81]	63	51	16.9	71.2	78.9
	MVIG	63.2	50.8	-	-	-
	Xiao et al.[80]	74.6	57.8	62.6	79.4	80.3
Bottom-up	JointFlow[16]	63.6	53	23.2	82.1	70.6
	Jin et al.[33]	59.16	50.59	-	-	-
	TML++	68.78	54.46	85.2	80	76.1

Table 5.3: Pose estimation and tracking performance on PoseTrack 2018 test dataset.

Method	Additional training data	MOTA	mAP	Wrists AP	Ankles AP
Xiao et al. [80]	+COCO+Other	61.37	74.03	73	69.05
ALG	+COCO+Other	60.79	74.85	72.62	71.11
Miracle	+COCO+Other	57.36	70.9	68.19	66.06
CMP	+COCO	54.47	64.67	61.78	60.86
PR	+COCO	44.54	59.05	50.16	49.4
TML++	+COCO	54.86	67.81	60.2	56.85

Joint-Flow map has less information than the TML. Moreover, we compared with JointFlow [16] that proposed a temporal map about joint movement as

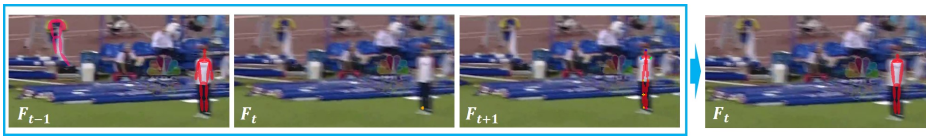


Figure 5.5: An example of pose refinement using multi-stride inputs during the inference.

The person at the right side of input images (red line) is tracked from F_{t-1} to F_{t+1}, but the pose of the person is not detected at F_t. By associating the poses at F_{t-1} and F_{t+1}, we can retrieve the missed pose at F_t. On the other hand, we cannot refine the person on the left side (pink line), because it is only estimated at the F_{t-1}.

shown in Table 5.2. On the PoseTrack 2017 test set, our results are better than those of the JointFlow [16].

Because the proposed method is the bottom-up approach, it is possible to detect many joint candidates on the same part. Thus, a non-maximum suppression (NMS) is applied for joints to reduce confusion after estimating joint location. (+) in Table 5.1 means that first we detect joints using the joint heatmaps and refine the joint using NMS. Reducing the confusing candidates increases tracking performance by around 0.4% in mAP and 0.6% in MOTA.

The sum of the TML score and the joint distance score is used for the association score to track poses. We experimented to see how the joint distance affects to association score. On Table 5.1, (Distance) means that only joint distances of a person is used in the calculation of association score. To enable this, at inference time, we use the same structure as TML+ and set α to 0. Only using the distance score incurs more confusion with nearby people and the resultant MOTA is by far lower than others on average. However, we need to use the distance score to handle the case of no motion. Thus, we apply α to 0.5 in all the other cases.

One of our contributions is the refining method for the middle frame pose. We refine the pose on the middle of frame by analyzing between three frames. (++) in Table 5.1, Table 5.2, and Table 5.3 means that the refining method is applied. Figure 5.5 shows an example result of the refined pose. The pose on F_t is refined through the association between frames F_{t-1} and F_{t+1}. In case of the person on the right side (red line), the person is tracked at the F_{t-1} and the F_{t+1}, but not tracked at the F_t. Through the refining method, an average pose between F_{t-1} and F_{t+1} is added on the frame F_t. Unfortunately, the person on the left side (pink line) can not be tracked through the refining method, because the pose is not estimated at the F_{t+1}.

Figure 5.6 shows qualitative results of pose estimation and tracking. Poses are estimated and tracked well in a variety of environments even when several people move close together or quickly. Because our association score considers the distance score, poses that have a little movement can also be tracked as shown in the fourth row on Figure 5.6. Unfortunately, if the poses nearly occludes each other as in the last row of Figure 5.6, the pose is likely to be missed. For future work, we may propagate the pose through the TML and refine the estimated pose by comparing it with the propagated pose to address this.

We compare our method with the state-of-the-art methods on the PoseTrack 2017 and 2018 test datasets as shown in Table 5.2. Though the proposed method shows a lower performance than the highest record [80], the result of the proposed network is the best among the bottom-up approaches. The bottom-up method tends to be relatively less accurate than the top-down method, which detects people first and estimates poses. So mAP of proposed method tends to be relatively lower than the top-down method. However, our method is available to be advantageous in a blurry situation because TML is used to refine poses in the post-processing stage.

Because the PoseTrack challenge was held on the September 2018, papers using the PoseTrack 2018 data have not been published yet. We could not compare the proposed method with other methods on the PoseTrack 2018 validate data. However, we can compare results of state-of-the-art on the PoseTrack 2018 test data through the results on the PoseTrack leader-board site as shown in the Table 5.3. We cannot compare the structures of the networks, but ours shows the best performance among the ones trained only using COCO data.

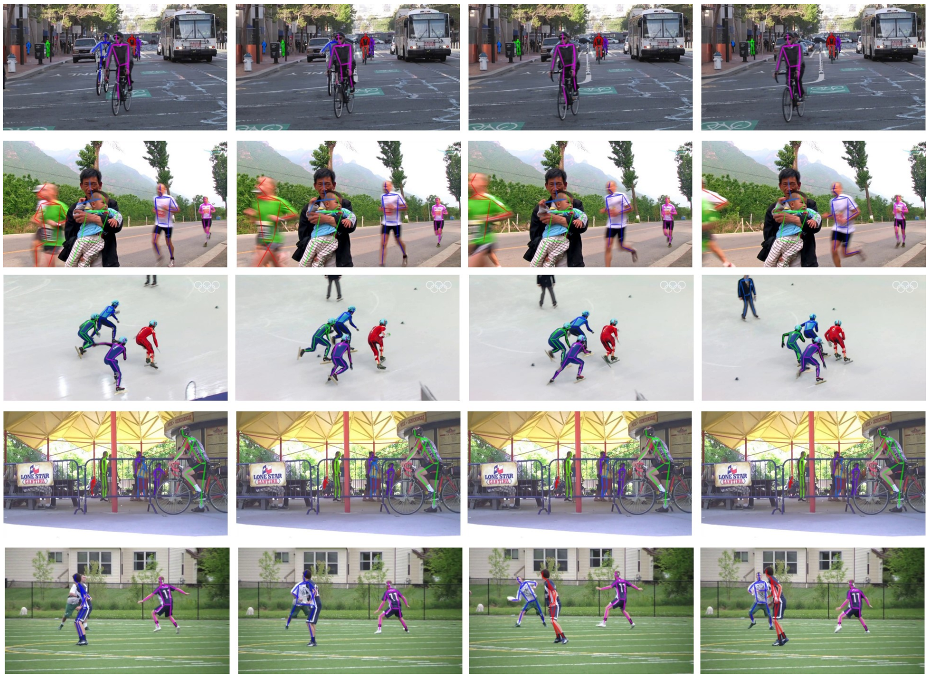


Figure 5.6: The qualitative results of the proposed multi-stride pose estimator and tracker.

The images are in chronological order from left to right. Tracked poses are displayed in the same color.

Chapter 6

Future work

For a better performance of human pose estimation models, we have analyzed the spatial and temporal features and proposed three different methods. First, two types of spatial features, global and local features, have been efficiently exploited by using an end-to-end global and local network. Second, we have proposed TML which can understand the continuous movements of limbs. By training the TML with a spatial map, the spatial and temporal features can exchange information each other. Finally, in order to balance the amount of the usual pose and rare pose, we have defined the rare pose and several methods to improve the performance. Methods in this paper may seem to be irrelevant each other but can be integrated and utilized altogether. For example, we can propose an higher accuracy pose estimator and tracker by combining the all methods mentioned above.

As shown in Figure 6.1, a new structure combined with proposed methods can be suggested to estimate and track a pose in a top-down manner. The new structure consists of two parts which are respectively a feature extractor and a module of estimating maps. The feature extractor network ($Network$ in Figure

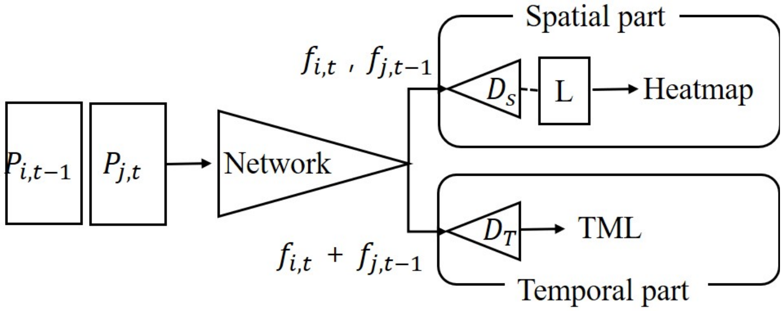


Figure 6.1: The new structure is combined with the proposed methods. P is a input image, i, j is an unique person identity and $t, t-1$ are frames. Input images ($P_{i,t}, P_{i,t-1}$) are fed input the backbone network ($Network$). Feature (f) is extracted on the backbone network and fed into each deconvolution layer (D_S, D_T). Each deconvolution layer generates spatial and temporal maps ($Heatmap, TML$), respectively. Additionally, the Local network (L) is applied to learn the Heatmap.

6.1) is based on frequently used backbone networks such as ResNet and MobileNet. The feature extractor extracts the feature of person image to provide input feature of the spatial and temporal parts.

Introduced in Chapter 5, our multi-person estimation model using spatial and temporal networks proceeds several stages of learning the maps. In our method, the repeated stage structure is adopted to harvest the feature because bottom-up methods usually uses the whole part of an image and an intensive extraction is needed. However, the newly proposed method is a top-down method and does not require repetitions of modules. [80] has proposed a structure that connects three deconvolution layers to the backbone network. The performance is adequate and one of the top-ranking methods in multi-person pose estimation. Accordingly, only a few deconvolution layers are used in the new structure to learn maps.

In the proposing method, Heatmap and TML can be adopted for a better performance. The Heatmap is a Gaussian map representing the location of a part and the TML a the unit vector map representing the movement of limbs. For the training, each stage of decovolution as shown in Figure 6.1 is used to process each map. D_S and D_T denote the decovolution networks. The D_S takes the output feature of the backbone network as the input $(f_{i,t}, f_{j,t-1})$ frame-by-frame. On the other hands, The D_T takes the concatenated feature of two frames $(f_{i,t} + f_{j,t-1})$. Additionally, at the spatial part module, the local network we proposed is applied.

Because the unique bounding box of a person is given in every frame at training time, bounding boxes of i'th person from different time steps $P_{i,t}, P_{i,t-1}$ are available. At inference time, bounding boxes of humans are detected using a human detector such as Mask R-CNN [1]. Unlike in the training time, the identity of a person is unknown making bounding boxes of the same person from different frames hard to match. The simplest way of choosing an input bounding box pair is to check all the combinations. To reduce the running time, only pairs of boxes having low enough pose distance may be under consideration. To track the pose, we calculate the associated score using the TML and the distance of pose. Among candidates of box pair above, one with the highest associated score is selected to be the same person. Finally, our rare pose augmentation can be used to balance the general pose and rare pose at the training time.

By Observing the [80] which has a similar structure with our proposed method, the highest performance in COCO validation is 70.4 AP and our proposed method is available to reach high performance. Also, the proposed method parameter is almost 43 million with $256 * 192$ input image size and ResNet-50 backbone. By the small number of parameter, the proposed method is available in real-time.

Bibliography

[1] W. Abdulla. Mask r-cnn for object detection and instance segmentation on keras and tensorflow. https://github.com/matterport/Mask$_RCNN$, 2017.

[2] M. Andriluka, U. Iqbal, E. Ensafutdinov, L. Pishchulin, A. Milan, J. Gall, and S. B. PoseTrack: A benchmark for human pose estimation and tracking. In *CVPR*, 2018.

[3] M. Andriluka, L. Pishchulin, P. Gehler, and B. Schiele. 2d human pose estimation: New benchmark and state of the art analysis. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2014.

[4] B. Artacho and A. Savakis. Unipose: Unified human pose estimation in single images and videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7035–7044, 2020.

[5] Q. Bao, W. Liu, Y. Cheng, B. Zhou, and T. Mei. Pose-guided tracking-by-detection: Robust multi-person pose tracking. *IEEE Transactions on Multimedia*, 2020.

[6] M. Buda, A. Maki, and M. A. Mazurowski. A systematic study of the class imbalance problem in convolutional neural networks. *Neural Networks*, 106:249–259, 2018.

[7] A. Bulat and G. Tzimiropoulos. Human pose estimation via convolutional part heatmap regression. In *ECCV*, 2016.

[8] Y. Cai, Z. Wang, Z. Luo, B. Yin, A. Du, H. Wang, X. Zhou, E. Zhou, X. Zhang, and J. Sun. Learning delicate local representations for multi-person pose estimation, 2020.

[9] Y. Cao, K. Chen, C. C. Loy, and D. Lin. Prime sample attention in object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11583–11591, 2020.

[10] Z. Cao, T. Simon, S.-E. Wei, and Y. Sheikh. Realtime multi-person 2d pose estimation using part affinity fields. *CVPR*, 2017.

[11] X. Chen and A. L. Yuille. Articulated pose estimation by a graphical model with image dependent pairwise relations. In *Advances in Neural Information Processing Systems*, pages 1736–1744, 2014.

[12] Y. Chen, C. Shen, X.-S. Wei, L. Liu, and J. Yang. Adversarial posenet: A structure-aware convolutional network for human pose estimation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1212–1221, 2017.

[13] Y. Chen, Z. Wang, Y. Peng, Z. Zhang, G. Yu, and J. Sun. Cascaded pyramid network for multi-person pose estimation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7103–7112, 2018.

[14] C.-J. Chou, J.-T. Chien, and H.-T. Chen. Self adversarial training for human pose estimation. In *2018 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, pages 17–30. IEEE, 2018.

[15] X. Chu, W. Yang, W. Ouyang, C. Ma, A. L. Yuille, and X. Wang. Multi-context attention for human pose estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1831–1840, 2017.

[16] A. Doering, U. Iqbal, and J. Gall. Joint flow: Temporal flow fields for multi person tracking. *BMVC*, 2018.

[17] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results. http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html.

[18] X. Fan, K. Zheng, Y. Lin, and S. Wang. Combining local appearance and holistic view: Dual-source deep neural networks for human pose estimation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1347–1355, 2015.

[19] H.-S. Fang, S. Xie, Y.-W. Tai, and C. Lu. RMPE: Regional multi-person pose estimation. In *ICCV*, 2017.

[20] A. Fernández, S. Garcia, F. Herrera, and N. V. Chawla. Smote for learning from imbalanced data: progress and challenges, marking the 15-year anniversary. *Journal of artificial intelligence research*, 61:863–905, 2018.

[21] R. Girdhar, G. Gkioxari, L. Torresani, M. Paluri, and D. Tran. Detect-and-Track: Efficient Pose Estimation in Videos. In *CVPR*, 2018.

[22] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

[23] D. Groos, H. Ramampiaro, and E. Ihlen. Efficientpose: Scalable single-person pose estimation. *arXiv preprint arXiv:2004.12186*, 2020.

[24] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[25] B. K. Horn and B. G. Schunck. Determining optical flow. *Artificial intelligence*, 17(1-3):185–203, 1981.

[26] Y.-C. Hsu, Y. Shen, H. Jin, and Z. Kira. Generalized odin: Detecting out-of-distribution image without learning from out-of-distribution data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10951–10960, 2020.

[27] Y. Huang, B. Sun, H. Kan, J. Zhuang, and Z. Qin. Followmeup sports: New benchmark for 2d human keypoint recognition. In *Chinese Conference on Pattern Recognition and Computer Vision (PRCV)*, pages 110–121. Springer, 2019.

[28] E. Insafutdinov, M. Andriluka, L. Pishchulin, S. Tang, E. Levinkov, B. Andres, and B. Schiele. ArtTrack: Articulated Multi-person Tracking in the Wild. In *CVPR*, 2017.

[29] E. Insafutdinov, L. Pishchulin, B. Andres, M. Andriluka, and B. Schieke. Deepercut: A deeper, stronger, and faster multi-person pose estimation model. In *European Conference on Computer Vision (ECCV)*, 2016.

[30] U. Iqbal, A. Milan, and J. Gall. Posetrack: Joint multi-person pose estimation and tracking. In *CVPR*, 2017.

[31] M. A. Jamal, M. Brown, M.-H. Yang, L. Wang, and B. Gong. Rethinking class-balanced methods for long-tailed visual recognition from a domain

adaptation perspective. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7610–7619, 2020.

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

[33] S. Jin, X. Ma, Z. Han, Y. Wu, W. Yang, W. Liu, C. Qian, and W. Ouyang. Towards multi-person pose tracking: Bottom-up and top-down methods. In *ICCV PoseTrack Workshop*, 2017.

[34] J. M. Johnson and T. M. Khoshgoftaar. Survey on deep learning with class imbalance. *Journal of Big Data*, 6(1):27, 2019.

[35] S. Johnson and M. Everingham. Clustered pose and nonlinear appearance models for human pose estimation. In *Proceedings of the British Machine Vision Conference*, 2010. doi:10.5244/C.24.12.

[36] S. Johnson and M. Everingham. Learning effective human pose estimation from inaccurate annotation. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2011.

[37] L. Ke, M.-C. Chang, H. Qi, and S. Lyu. Multi-scale structure-aware network for human pose estimation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 713–728, 2018.

[38] J. Kim, M. Kim, H. Kang, and K. Lee. U-gat-it: unsupervised generative attentional networks with adaptive layer-instance normalization for image-to-image translation. *arXiv preprint arXiv:1907.10830*, 2019.

[39] M. Kocabas, S. Karagoz, and E. Akbas. Multiposenet: Fast multi-person pose estimation using pose residual network. In *Proceedings of the European conference on computer vision (ECCV)*, pages 417–433, 2018.

[40] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[41] B. Li, Y. Liu, and X. Wang. Gradient harmonized single-stage detector. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 8577–8584, 2019.

[42] W. Li, Z. Wang, B. Yin, Q. Peng, Y. Du, T. Xiao, G. Yu, H. Lu, Y. Wei, and J. Sun. Rethinking on multi-stage networks for human pose estimation. *arXiv preprint arXiv:1901.00148*, 2019.

[43] Y. Li, K. He, J. Sun, et al. R-fcn: Object detection via region-based fully convolutional networks. In *Advances in Neural Information Processing Systems*, pages 379–387, 2016.

[44] Y. Li, T. Wang, B. Kang, S. Tang, C. Wang, J. Li, and J. Feng. Overcoming classifier imbalance for long-tail object detection with balanced group softmax. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10991–11000, 2020.

[45] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017.

[46] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.

[47] C. X. Ling and C. Li. Data mining for direct marketing: Problems and solutions. In *Kdd*, volume 98, pages 73–79, 1998.

[48] S. Lloyd. Least squares quantization in pcm. *IEEE transactions on information theory*, 28(2):129–137, 1982.

[49] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3431–3440, 2015.

[50] M. Loper, N. Mahmood, J. Romero, G. Pons-Moll, and M. J. Black. SMPL: A skinned multi-person linear model. *ACM Trans. Graphics (Proc. SIGGRAPH Asia)*, 34(6):248:1–248:16, Oct. 2015.

[51] D. C. Luvizon, D. Picard, and H. Tabia. 2d/3d pose estimation and action recognition using multitask deep learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5137–5146, 2018.

[52] D. C. Luvizon, D. Picard, and H. Tabia. 2d/3d pose estimation and action recognition using multitask deep learning. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[53] W. McNally, K. Vats, A. Wong, and J. McPhee. Evopose2d: Pushing the boundaries of 2d human pose estimation using neuroevolution. *arXiv preprint arXiv:2011.08446*, 2020.

[54] A. Milan, L. Leal-Taixé, I. Reid, S. Roth, and K. Schindler. Mot16: A benchmark for multi-object tracking. *arXiv preprint arXiv:1603.00831*, 2016.

[55] G. Moon, J. Y. Chang, and K. M. Lee. Posefix: Model-agnostic general human pose refinement network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7773–7781, 2019.

[56] F. Mueller, F. Bernard, O. Sotnychenko, D. Mehta, S. Sridhar, D. Casas, and C. Theobalt. Ganerated hands for real-time 3d hand tracking from monocular rgb. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 49–59, 2018.

[57] A. Newell, K. Yang, and J. Deng. Stacked hourglass networks for human pose estimation. In *European conference on computer vision*, pages 483–499. Springer, 2016.

[58] G. Ning, J. Pei, and H. Huang. Lighttrack: A generic framework for online top-down human pose tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 1034–1035, 2020.

[59] G. Ning, Z. Zhang, and Z. He. Knowledge-guided deep fractal neural networks for human pose estimation. *IEEE Transactions on Multimedia*, 20(5):1246–1259, 2017.

[60] K. Oksuz, B. C. Cam, S. Kalkan, and E. Akbas. Imbalance problems in object detection: A review. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020.

[61] G. Papandreou, T. Zhu, N. Kanazawa, A. Toshev, J. Tompson, C. Bregler, and K. Murphy. Towards accurate multi-person pose estimation in the wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4903–4911, 2017.

[62] X. Peng, Z. Tang, F. Yang, R. S. Feris, and D. Metaxas. Jointly optimize data augmentation and network training: Adversarial data augmentation in human pose estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2226–2234, 2018.

[63] L. Pishchulin, E. Insafutdinov, S. Tang, B. Andres, M. Andriluka, P. Gehler, and B. Schiele. Deepcut: Joint subset partition and labeling for multi person pose estimation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[64] S. Pouyanfar, Y. Tao, A. Mohan, H. Tian, A. S. Kaseb, K. Gauen, R. Dailey, S. Aghajanzadeh, Y.-H. Lu, S.-C. Chen, et al. Dynamic sampling in convolutional neural networks for imbalanced data classification. In *2018 IEEE conference on multimedia information processing and retrieval (MIPR)*, pages 112–117. IEEE, 2018.

[65] Y. Raaj, H. Idrees, G. Hidalgo, and Y. Sheikh. 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*, pages 4620–4628, 2019.

[66] I. Radwan, A. Asthana, and R. Geocke. Global pose refinement using bidirectional long-short term memory. https://posetrack.net/workshops/iccv2017/pdfs/MPR.pdf.

[67] W. Ruan, W. Liu, Q. Bao, J. Chen, Y. Cheng, and T. Mei. Poinet: pose-guided ovonic insight network for multi-person pose tracking. In *Proceedings of the 27th ACM International Conference on Multimedia*, pages 284–292, 2019.

[68] A. Shrivastava, A. Gupta, and R. Girshick. Training region-based object detectors with online hard example mining. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 761–769, 2016.

[69] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[70] M. Snower, A. Kadav, F. Lai, and H. P. Graf. 15 keypoints is all you need. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6738–6748, 2020.

[71] K. Sun, B. Xiao, D. Liu, and J. Wang. Deep high-resolution representation learning for human pose estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5693–5703, 2019.

[72] W. Tang, P. Yu, and Y. Wu. Deeply learned compositional models for human pose estimation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 190–206, 2018.

[73] A. Toshev and C. Szegedy. Deeppose: Human pose estimation via deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1653–1660, 2014.

[74] J. Van Hulse, T. M. Khoshgoftaar, and A. Napolitano. Experimental perspectives on learning from imbalanced data. In *Proceedings of the 24th international conference on Machine learning*, pages 935–942, 2007.

[75] C. Wang, Y. Wang, and A. L. Yuille. An approach to pose-based action recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 915–922, 2013.

[76] M. Wang, J. Tighe, and D. Modolo. Combining detection and tracking for human pose estimation in videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11088–11096, 2020.

[77] S. Wang, W. Liu, J. Wu, L. Cao, Q. Meng, and P. J. Kennedy. Training deep neural networks on imbalanced data sets. In *2016 international joint conference on neural networks (IJCNN)*, pages 4368–4374. IEEE, 2016.

[78] X. Wang, Y. Lyu, and L. Jing. Deep generative model for robust imbalance classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14124–14133, 2020.

[79] S.-E. Wei, V. Ramakrishna, T. Kanade, and Y. Sheikh. Convolutional pose machines. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 4724–4732, 2016.

[80] B. Xiao, H. Wu, and Y. Wei. Simple baselines for human pose estimation and tracking. In *Proceedings of the European conference on computer vision (ECCV)*, pages 466–481, 2018.

[81] Y. Xiu, J. Li, H. Wang, Y. Fang, and C. Lu. Pose flow: Efficient online pose tracking. *BMVC*, 2018.

[82] J. Yang, H. J. Chang, S. Lee, and N. Kwak. Seqhand: Rgb-sequence-based 3d hand pose and shape estimation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020.

[83] W. Yang, W. Ouyang, H. Li, and X. Wang. 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*, pages 3073–3082, 2016.

[84] M. Z. Zaheer, J.-h. Lee, M. Astrid, and S.-I. Lee. Old is gold: Redefining the adversarially learned one-class classifier training paradigm. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14183–14193, 2020.

[85] F. Zhang, X. Zhu, H. Dai, M. Ye, and C. Zhu. Distribution-aware coordinate representation for human pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7093–7102, 2020.

[86] F. Zhang, X. Zhu, and M. Ye. Fast human pose estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3517–3526, 2019.

[87] H. Zhang, H. Ouyang, S. Liu, X. Qi, X. Shen, R. Yang, and J. Jia. Human pose estimation with spatial contextual information. *arXiv preprint arXiv:1901.01760*, 2019.

[88] A. Zhu, S. Zhang, Y. Huang, F. Hu, R. Cui, and G. Hua. Exploring hard joints mining via hourglass-based generative adversarial network for human pose estimation. *AIP Advances*, 9(3):035321, 2019.

[89] C. L. Zitnick and P. Dollár. Edge boxes: Locating object proposals from edges. In *European Conference on Computer Vision*, pages 391–405. Springer, 2014.

초 록

2D 이미지에서 사람의 포즈를 검출하는 연구는 사람의 파트들의 위치를 검출하는 것을 목표로한다. 포즈는 사람의 파트들로 구성되어 있고 사람의 파트는 팔, 다리, 머리 등으로 사람을 구성하는 신체의 요소들을 의미한다. 사람의 포즈 정보는 다양한 분야에서 활용 될 수 있다. 또한, 사람의 동작 감지 연구 분야에서는 사람의 포즈 정보가 매우 훌륭한 입력 특징 값으로 사용된다.

사람의 포즈 검출 연구를 실제 시스템에 적용하기 위해서는 높은 정확도, 실시간성, 다양한 기기에 사용 가능하도록 가벼운 모델이 필요하다. 본 논문에서는 정확도를 개선하는 연구에 초점을 맞췄다. 높은 정확도를 달성하기 위해서 특징값을 어떻게 활용할지에 대해 고민을 했으며, 지역적 특징값과 시간 특징값을 사용해서 문제를 개선했다.

지역적 특징값은 사람의 텍스쳐, 형태와 같은 특징을 표현하는 것을 의미한다. 우리는 지역적 특징 값을 다수의 파트를 담고 있는 Global feature 와 소수의 파트를 담고 있는 Local feature로 분류해서 문제를 접근했다. 첫번째로는 global-local feature 을 동시에 사용해서 성능을 개선하는 연구에 집중했다. Global feature을 집중적으로 학습하는 네트워크와 다양한 형태의 local 정보를 학습 할 수 있는 local network을 설계했다. Local network에서는 global network에서 검출한 포즈를 다시 한번 개선하는 역할을 수행한다. 제안된 방법의 효율성을 증명하기 위해서 single-person pose estimation 데이터 중 하나

인 Leeds sports dataset (LSP) 데이터에서 실험을 수행했다.

두번째로는 global한 정보를 통해 희귀한 포즈를 검출해서 포즈의 불균형을 해소해 성능을 개선하는 연구를 수행했다. 우선적으로 포즈 데이터 내에서 전체 포즈의 위치 정보를 사용해서 포즈들을 분류했다. 실험 결과, 일정 포즈를 (서있는 포즈, 상반신만 있는 포즈 등) 중심으로 포즈들이 분포 된며 포즈 간의 불균형이 있음을 밝혀냈다. 우리는 포즈 간의 불균형을 해소하기 위해 weight loss, generate rare pose data 등의 방법을 제안했다. 제안된 방법의 효율성을 증명하기 위해서 multi-person pose estimation 데이터에서 많이 사용되는 MPII와 COCO 데이터에서 실험을 수행했다.

시간 특징값은 시간 흐름에 따른 움직임 변화값을 의미한다. 동영상에서 객체를 분석하기 위해서는 시간 정보를 활용하는 것이 좋다. 그래서 세번째로 우리는 사람의 움직임 변화를 맵으로 표현해서 포즈를 추적했다. 이때 포즈의 지역적 특징값과 같이 학습해서 서로간의 시너지 효과를 낼 수 있도록 네트워크를 제안했다. 제안된 방법의 효율성을 증명하기 위해서 multi-person pose tracking 데이터인 posetrack 2017 과 2018에서 실험을 수행했다.

본 논문에서는 지역적 특징과 시간적 특징을 활용해서 포즈의 성능을 개선하는 방법들을 제안했다. 서로 다른 문제들을 해결했지만 나아가 하나로 묶여 문제를 해결 할 수 있다. 예를들어, top-down 형태의 네트워크 구조에서 Heatmap과 TML을 각각 학습 할 수 있는 평행적 구조의 decovolution network을 제안 할 수 있다. 여기에 Heatmap의 성능 개선을 위해 local network와 rare pose data augmentation 방식 또한 추가할 수 있다. 이렇게 제안된 방법을 결합해서 더 나은 포즈의 성능을 개선 할 수 있는 방법들이 제안 될 수 있다.

주요어: 2차원 사람 자세 인식, 2차원 사람 자세 추적, 공간적 특징값, 시간적 특징값, 희귀포즈 분석, 시간정보 맵, 딥러닝, 클러스터링

학번: 2015-31349

감사의 글

이 논문을 마치며 감사한 분들이 참 많았습니다. 논문의 심사를 맡아주신 이교구 교수님, 곽노준 교수님, 박재흥 교수님, 최상일 교수님, 이민식 교수님 정말 감사드립니다. 바쁜 시간 내주셔서 논문이 좋은 방향으로 나아갈 수 있도록 심사해 주셔서 감사합니다. 특히 저의 지도 교수님이신 곽노준 교수님께 감사드립니다. 박사과정 동안 교수님께 학문적으로나 인간적으로나 배운 것이 참 많습니다. 앞으로 제가 이 연구 분야에서 교수님을 본 받아 발전하는 모습 보여드리겠습니다.

항상 제 곁에서 힘이 되어주는 우리 가족에게 너무 감사합니다. 사랑하는 우리 남편 김지수씨, 자기 주장 강한 고집불통의 아내이지만 매번 연구 얘기하면 자기일 처럼 잘들어주고 고민해줘서 너무 고맙습니다. 외롭고 끝이 보이지 않을 것만 같던 연구과정 속에서 든든하게 지켜줘서 고맙습니다. 세상에서 가장 존경하는 우리 부모님, 누구보다 이 긴 시간을 걱정하셨을텐데 괜찮다며 잘하고 있다고 뒤에서 항상 응원해주시고 믿어주셔서 감사합니다. 든든한 우리 오빠, 묵묵히 힘내라고 응원해줘서 고마워. 인자하신 우리 시부모님, 아직 부족하고 철없는 며느리를 항상 먼저 생각해서 배려해주셔서 감사드립니다.

모든 연구실 선, 후배 분들에게도 너무 감사합니다. 하루 중에 가장 많은 시간을 함께해서 추억도 많고 고마운 마음도 많습니다. 더욱 주변을 잘 챙겼어야 했는데라는 아쉬움이 남습니다. 항상 어딜가도 우리 연구실 같은 곳은 없을거라고 얘기할 정도로 우리 연구실이 참 좋았습니다. 서로에게 너무

잘해주고 배려해주면서 같이 연구하려는 모습들이 제가 앞으로 공동체에서 어떻게 생활을 해야하는지를 다시 한번 배울 수 있었습니다. 연구의 정답은 없기 때문에 스트레스와의 싸움이 이어진다고 생각합니다. 항상 몸과 마음을 건강해서 건승하길 진심으로 바랍니다.