Graph-Based Human Pose Estimation Using Neural Networks

A thesis submitted in fulfillment of the requirements for the degree of Doctor of Philosophy

HUYNH THE VU

M.E., Electrical and Computer Engineering, RMIT University, Viet Nam, 2014

School of Engineering
College of Science, Engineering and Health
RMIT University
June 2019

Declaration

I certify that except where due acknowledgement has been made, the work is that of the author alone; the work has not been submitted previously, in whole or in part, to qualify for any other academic award; the content of the thesis is the result of work which has been carried out since the official commencement date of the approved research program; any editorial work, paid or unpaid, carried out by a third party is acknowledged; and, ethics procedures and guidelines have been followed.

Huynh The Vu
June 2019

Acknowledgements

I would like to thank the School of Engineering, RMIT Melbourne for the PhD scholarship, who gave me the opportunity to conduct research in my favorite area, also a trending research field, of using neural networks for everyday applications. During my research, the School provided modern computational resources, e.g., an NVIDIA M60 graphics card, and related hardware to conduct research experiments.

I would like to specially thank my supervisors: Professor Margaret Lech, Dr. Richardt Wilkinson, and Dr. Eva Cheng for the continuous support during my PhD candidature. I missed the tough time in the first few months of my PhD, as well as before important milestones, when you kept me on track and motivated me to complete my tasks. I appreciate your friendly supervision style, which reduced the stress and motivated me to do the work. My PhD journey would not have been as challenging without your great support.

I am grateful to my parents, who allowed me the freedom to pursue whichever path I chose and who provided me with moral and emotional support.

To my fellow PhD candidates at the School of Science and Technology, RMIT Vietnam, Ilya Kavalchuk, Hieu Bui Minh, Phuong Tang Tang, Linh Tran Duc, thank you. Our daily chats and discussions were immeasurably helpful and motivating. Without you, this PhD would have been much less enjoyable.

List of publications

[1] H. Vu, E. Cheng, R. Wilkinson, and M. Lech, “On the use of convolutional neural networks for graphical model-based human pose estimation,” in Proceedings of the IEEE Conference on Recent Advances in Signal Processing (SigTelCom), Da Nang, pp. 88–93, 2017.

[2] H. T. Vu, R. H. Wilkinson, M. Lech, and E. Cheng, “A comparison between anatomy-based and data-driven tree models for human pose estimation,” in Proceedings of the International Conference on Digital Image Computing: Techniques and Applications (DICTA), Sydney, pp. 1-7, 2017.

List of Abbreviations

HPE	Human Pose Estimation
2D	Two-Dimensional
CNNs	Convolutional Neural Networks
GPUs	Graphics Processing Units
LSP	Leeds Sports Pose
FLIC	Frames Labeled In Cinema
PCK	Percentage of Correct Keypoints
PDJ	Percentage of Detected Joints
PCP	Percentage of Correct Parts
PS	Pictorial Structure
HOG	Histogram of Oriented Gradient
SIFT	Scale Invariant Feature Transform
FC	Fully Connected
ReLU	Rectified Linear Unit
LRN	Local Response Normalization
PC	Person Centric
OC	Observer Centric
MPN	Message Passing Network

Abstract

This thesis investigates the problem of human pose estimation (HPE) from unconstrained single two-dimensional (2D) images using Convolutional Neural Networks (CNNs). Recent approaches propose to solve the HPE problem using various forms of CNN models. Some of these methods focus on training deeper and more computationally expensive CNN structures to classify images of people without any prior knowledge of their poses. Other approaches incorporate an existing prior knowledge of human anatomy and train the CNNs to construct graph-representations of the human pose. These approaches are generally characterised as having lower computational and data requirements.

This thesis investigates HPE methods based on the latter approach. In the search for the most accurate and computationally efficient HPE, it explores and compares three types of graph-based pose representations: tree-based, non-tree based, and a hybrid approach combining both representations. The thesis contributions are three-fold. Firstly, the effect of different CNN structures on the HPE was analysed. New, more efficient network configurations were proposed and tested against the benchmark methods. The proposed configurations achieved offered computational simplicity while maintaining relatively high-performance. Secondly, new data-driven tree-based models were proposed as a modified form of the Chow-Liu Recursive Grouping (CLRG) algorithm. These models were applied within the CNN-based HPE framework showing higher performance compared to the traditional anatomy-based tree-based models. Experiments with different numbers and configurations of tree nodes allowed the determination of a very efficient tree-based configuration consisting of 50 nodes. This configuration achieved higher HPE accuracy compared to the previously proposed 26-node tree. Apart from tree-based models of human pose, efficient non-tree-based models with iterative (looping) connections between nodes were also investigated. The third contribution of this thesis is a novel hybrid HPE framework that combines both tree-based and non-tree-based human pose representations. Experimental results have shown that the hybrid approach leads to higher accuracy compared to either tree-based, or non-tree-based structures individually.

Table of contents

List of figures

List of tables

Chapter 1

Introduction

1.1 Preview

This chapter provides the problem statement and outlines the aim, scope, contributions and the thesis structure.

1.2 Human Pose Estimation (HPE)

1.2.1 What is HPE?

Full-body pose estimation is an important building block of marker-less human motion analysis. In traditional marker-based human motion analysis, markers are required and attached to the body when taking a picture [6, 7]. On the other hand, marker-less human motion analysis is a non-intrusive and less expensive option. Human motion includes movements of body parts such as facial movements and hand movements, as well as full-body displacement. Typical marker-less human motion tasks include tracking (segmenting and tracking individual people), pose estimation (estimating poses of individuals) and recognition (determining identities or actions of individuals or groups) [7].

This thesis is concerned with the two-dimensional (2D) HPE from static images. Figure 1.1 shows a typical example of an HPE input image and the corresponding output locations

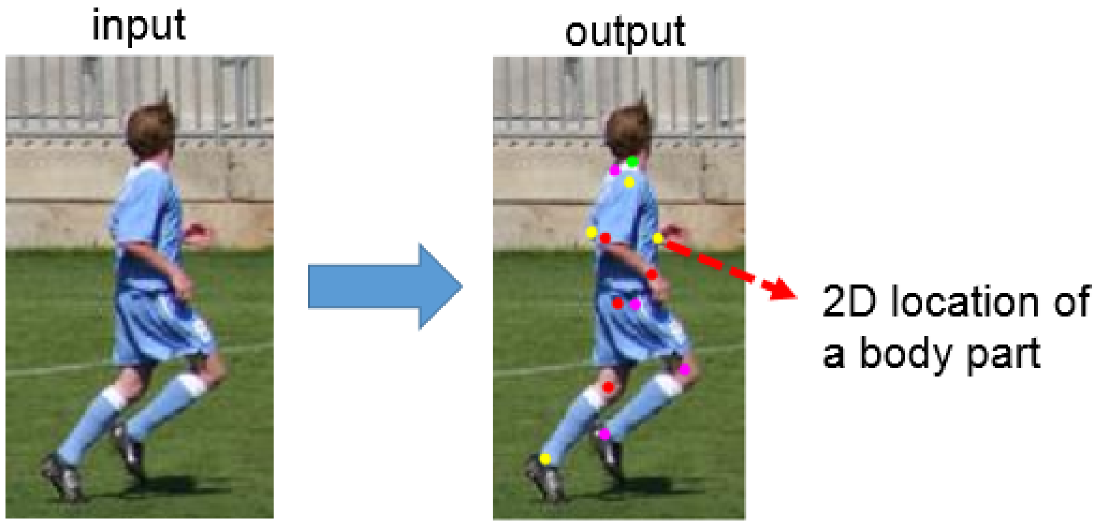


Fig. 1.1 Input and output of an HPE system (the image is taken from the LSP dataset [8]).

of the detected body joints imposed on the input image. Given an input image depicting a person, the person's pose was estimated by determining 2D locations of joints indicating positions of the head, shoulders, wrists, elbows, knees and ankles. Static 2D images can be used for the estimation of either 2D or three-dimensional (3D) poses. 3D pose estimation can be obtained using a single depth image [9] or a sequence of monocular images [10]. When the time evolution of the HPE is considered, the term of human motion analysis (HMA) is used [11].

1.2.2 Applications of HPE

HPE is an important building block of Human Activity Recognition (HAR) which aims at analyzing human activities and intepreting ongoing events given video data. HAR systems are based on the environment, spatial, temporal information and especially human poses to understand human behaviours. Applications of HAR range from systems for healthcare monitoring, security, and gaming animation. Traditionally, recognizing human activities was carried out by human operators. However, increasing of the number of cameras and the requests for continuous monitoring have caused this task to become costly and challenging [12].

In healthcare monitoring applications, systems were designed to handle urgent medical situations (e.g. fall detection) and to assist patients who suffer from diseases such as dementia and Alzheimer or assist the elderly or people with disabilities living independently. For example, the system proposed by Chen et al. [13] could automatically detect events associated with dementia and alert the caregivers so that immediate support could be provided to patients.

Applications for security have been applied in various places. For example, Bremond et al. [14] proposed a system to detect certain human behaviours, e.g. physical assault in metro areas. On the other hand, Chang et al. [15] applied HAR to identify aggressive behaviours of prisoners. For airports, a system was designed by Fusier et al. [16] to detect human activities such as unloading of baggages or refueling of aircraft.

In character animation, an animation sequence is generated based on all motions of a character together with its associated avatar. This procedure can be simplified using a human motion model which can output plausible human poses and motions. The application of human motion and pose analysis largely reduces the development cost and improves the performance of the character animation [17].

1.3 Background and problem statement

Estimation of the human pose from static 2D images can be formulated as a structured prediction problem in which the outputs (locations of joints) maintain a specific spatial relationship. In contrast to object detection, where the focus is on learning an accurate object location, HPE requires both accurate localization of the body parts and determining the correct relationship between the detected body parts. Assuming that this relationship can be described as a set of relative distances between body parts, it is important to note that it is not fixed and can vary depending on the given pose. Therefore, the process of determining the relationship between articulated body parts is a highly challenging task. Another important challenge to HPE is the presence of occlusions between body parts. This means that some body parts can be masked by other parts, or by surrounding objects, which can make the HPE even more challenging. In addition, low contrast, cluttered backgrounds, variations in the scene lighting, and the color scheme can also have a significant effect on the HPE accuracy.

Recent approaches have successfully applied deep convolutional neural networks (CNNs) to HPE. Due to their complex multi-layered structures, CNNs require a relatively large number of labeled (i.e. with given correct positions of body joints) training images to generate well performing models [1, 2]. Since the available training datasets often provide only a relatively small number of labeled images, the number of training data samples can be increased using data augmentation techniques such as image rotations, flipping or translation. This approach can significantly increase the number of training images and reduce the problem of over-fitting the model. Deeper and more complex CNN structures are more likely to reach higher levels of data generalization and discrimination capacity. Examples of such high-performing and complex neural network designs (with several types of neural networks stacked together) are given in [18–20]. These designs were shown to increase the accuracy of HPE. However, the data and computational costs were extremely high, making the use of graphics processing units (GPUs) paramount. To move away from increasing CNN depth and complexity, a number of studies have proposed to integrate "prior knowledge" (e.g. the "deformable mixture of parts" model) into CNNs to model structural information [2, 21]. These approaches offered low computational and training data requirements, while maintaining relatively high HPE accuracy.

1.4 Thesis aim

The thesis aim is to design a CNN-based approach that uses relatively low computational power while maintaining high HPE accuracy. This approach is constrained to apply the graph theory to model structural dependencies between body parts at the feature and output levels of the CNN. In the current CNN-based graph theory methods, the features characterizing body parts are represented as either tree or non-tree based structures of nodes representing body joints. These structures can be either anatomy-based or data-driven. When the structures are entered into the CNN, dependencies between body parts represented by the structures are maintained throughout the entire training process by systematic application of message passing procedures within the CNN [2, 21, 22]. In the search for the most efficient HPE approach, the thesis explores and compares all three types of graph-based pose representations: tree, non-tree and a combination of both. Within the tree-based group, anatomy-based and data-driven models are considered. All methods are consistently tested on the same benchmark Leeds Sports Pose (LSP) dataset [8]. Although the LSP dataset is a small dataset, it is a typical and challenging benchmark dataset which has been referred to by various papers [2, 21, 22]. Using only one dataset is the limitation of the thesis. The current pre-processing scheme of the LSP dataset for all experiments in this thesis is offline. Therefore, larger datasets would require a large amount of disk usage; otherwise, an on-the-fly pre-processing scheme should have been implemented. However, the implementation of this scheme is difficult and was deemed outside the scope of this thesis.

1.5 Thesis scope

HPE is a broad research field encompassing tasks such as pose estimation from 2D or 3D static images or videos, as well as pose estimation of a single person or multiple people. The pose can be labelled either descriptively (e.g. standing, sitting, running) or by a graph representing position coordinates of selected body parts.

The scope of this study is limited to HPE from static 2D images representing a single person with the pose labels given as graphs. All methods investigated in this study are tested and compared using the same LSP benchmark dataset [8].

The methodology investigated in this study is limited to CNN models which have been recently shown to provide outstanding performance in numerous image classification tasks.

In order to increase HPE accuracy, some CNN-based techniques can obtain high performance simply by making the networks deeper, which in turn makes the training data and computational requirements very high [18–20, 23–25]. Other methods look at ways of efficiently modelling the body structures and dependencies between body parts. These approaches offer significantly lower computational and data costs [2, 21, 22]. This research investigates the latter.

1.6 Thesis contributions

This thesis offers the following original contributions to the field of automatic HPE:

1. The effect of different CNN structures and transfer learning on the recognition of body parts from 2D images using the ChenNet proposed in [1] was analyzed. A new modified ChenNet (MChenNet) was proposed. Experimental results showed that the proposed MChenNet configurations achieved higher body-part recognition accuracy and used fewer network parameters than the original ChenNet network.

2. A new data-driven tree-based model for HPE was proposed and compared with an anatomy-based tree-based models. The two models are compared by comparing the HPE accuracy based on these models. Experimental results showed that the proposed data-driven tree-based model obtained higher HPE accuracy than the conventional anatomy-based tree-based models when applied within the same CNN-based framework introduced in [2].

3. The effect of node numbers in the tree-based pose representation on the accuracy of the HPE was investigated. The optimal number of tree nodes yielding significantly higher estimation accuracy compared to the conventionally used structures was determined.

4. The effect of different connections between body parts within the non-tree-based models on the HPE accuracy was investigated. As a result, new non-tree-based configurations obtaining higher HPE accuracy compared to the conventional non-tree-based models were proposed.

5. A novel hybrid HPE approach combining non-tree-based and tree-based pose representations was introduced. The hybrid model was shown to obtain higher HPE accuracy compared to the accuracy of either tree or non-tree representations alone.

1.7 Thesis narrative

To start with, in Chapter 2, the thesis describes conventional HPE techniques, recently used methods based on CNNs and research questions. Given the significant advantages of the CNN-based techniques over conventional approaches, as well as the existing potential for improvement, the remaining parts of the thesis are devoted to a detailed investigation of these techniques guided by the research questions introduced in Section 2.4.

Before the network can build an estimate of the pose representation, an image classification technique is applied to recognize body parts from 2D image patches taken from the original input image. The recognized body parts are then used to generate a tree-based or non-tree-based model by introducing an anatomy-based or data-driven set of dependencies between the recognized parts. This constitutes a pose model which is then iteratively refined through the CNN model training process.

The remaining parts of the thesis describe research investigation in an order that follows this procedure. Thus, in Chapter 3, the thesis investigates the optimization of the body part recognition procedure and shows that a high classification accuracy can be achieved at a low computational and data cost. Chapters 4 and 5 investigate various pose representations and their effect on the HPE accuracy. In particular, Chapter 4 compares data-driven tree-based models against the anatomy-based configurations and analyzes the effect of the number of tree nodes on the HPE accuracy. Given the limitations of the tree-based approaches, Chapter 5 investigates different non-tree-based representations. In Chapter 6, the feasibility of a hybrid approach that combines both tree-based and non-tree-based approaches is analyzed. Chapter 7 discusses the extent to which this study is able to answer the initial research questions and provides final conclusions.

1.8 Thesis structure

The thesis consists of the following seven chapters.

Chapter 1 contains the problem statement and outlines the thesis aims, scope, contributions and structure.

Chapter 2 describes what the HPE is and explains steps involved in the common CNN-based computational framework for the HPE that is applied throughout the thesis. The common testing database and the HPE performance measures used by the experimental validation procedures described in the thesis are discussed. A literature review of the conventional HPE techniques, as well as the recently emerging CNN-based approaches, is presented. Advantages and limitations of both types of methods as well as research questions will be discussed.

Chapter 3 investigates the optimization of the CNN network that is used at the beginning of the HPE procedure to recognize body parts depicted by image patches taken from the analyzed input image. The aim is to determine a network configuration that maximizes the body-part classification accuracy at the minimum computational cost. The factors considered in the optimization process are the network size, type of pooling scheme and the application of transfer learning.

Chapter 4 investigates the CNN-based HPE approach using tree-based representations (or models) of the human pose. New data-driven tree representations are proposed and compared with the conventional anatomy-based tree-based models. The effect of the number of tree nodes used to represent human pose on the HPE accuracy is investigated. The optimal number of nodes that gives the highest HPE accuracy within computational constraints is determined.

Chapter 5 investigates the CNN-based HPE approach using different non-tree-based structures and compares them with the tree-based models. The effect of different connections between body parts within the non-tree-based models on the HPE accuracy is analyzed.

Chapter 6 introduces two new hybrid CNN-based approaches to the HPE that combine both tree-based and non-tree-based pose representations. The proposed hybrid methods are compared with tree-based and non-tree based approaches.

Chapter 7 discusses to what extent the study is able to answer the initial research questions. It summarizes the thesis, gives final conclusions, and outlines possible future research directions.

Chapter 2

Computational benchmark and literature review

2.1 Preview

The chapter presents an overview of traditional and popular techniques used for HPE. HPE techniques are categorized into methods that do not apply CNN (Section 2.3.1) and CNN-based approaches (Section 2.3.2). Given their high performance and potential for improvements, this thesis is focused on the CNN-based methods for the HPE. This choice is justified by showing literature-based evidence of the high performance of CNNs, and by identifying existing potentials for improvement.

The chapter starts with a description of commonly used datasets and evaluation criteria for the HPE. The strict Percentage of Correct Part (PCP) criterion used to evaluate HPE in the thesis experiments is described. The non-CNN-based approaches described in this chapter include holistic and part-based models. The CNN-based approaches are described in a chronological order starting from methods, where CNNs play only the roles of part-detectors, and then moving to approaches where, CNNs are used both for detecting body parts and learning relationships between them.

Recent CNN-based approaches contain structures with successive predictors and stacked networks. In Section 2.3.3, CNN-based techniques are classified based on their main features such as network structures, image scales, successive predictors, CNN-based message passing units, and Gaussian heatmap labels. Advantages and limitations of each approach are discussed showing that further improvements can be made. The literature review leads to six research questions to be investigated in the thesis. These questions are presented in Section 2.4.

2.2 HPE Benchmark

2.2.1 Human pose estimation (HPE)

The process of HPE aims to provide a set of coordinates defining positions of body joints (or nodes) such as knee, elbow, neck, head, arm, etc., given that the input is a 2D image depicting a person. The nodes can be connected to form a graphical representation of the human pose. Figure 1.1 gives an example of a 2D image depicting a person (on the left), and the detected joints (on the right), representing a graph with a set of interconnected nodes that denote the "walking" pose.

HPE is an important building block for a variety of applications. With predicted body joints as the output, a person's pose can be described either as "walking", "standing", "sitting" or in the form of a graph depicting a set of interconnected nodes representing the person's joints. These pose descriptions can be used to recognize group activity or detect abnormal poses for a security system.

2.2.2 HPE framework

Estimation of the human pose from static 2D images can be formulated as a structured prediction problem in which the outputs (locations of joints) maintain a specific spatial relationship. Hence, an HPE framework often consists of two main processes: one process is to detect body parts, and the other, is to encode the relationship between the detected body parts. The following paragraphs describe typical HPE frameworks.

Before the introduction of CNNs (Section 2.3.1), one of the most popular HPE frameworks was based on the Pictorial Structure (PS) method introduced by Fischler and Martin [26]. This framework modeled an object as a group of parts connected in a deformable configuration. Each part represented local visual information of the object and the deformable configuration was featured as a set of spring-like connections between pairs of parts. In the context of HPE, a person was considered as a collection of object parts (or body joints). A framework for modelling HPE as structured object parts is described in Figure 2.1, which includes two separate processes: Process 1 and Process 2. Process 1 used part detectors to generate part heatmaps representing the probability distribution of body part locations. These detectors were learned by training the Histogram of Oriented Gradient (HoG) model [27] based on features extracted from body part patches as training inputs. Process 2 learned structural relationships between body-part features to determine and refine the best pose estimation from the body-part heatmaps generated in Process 1.

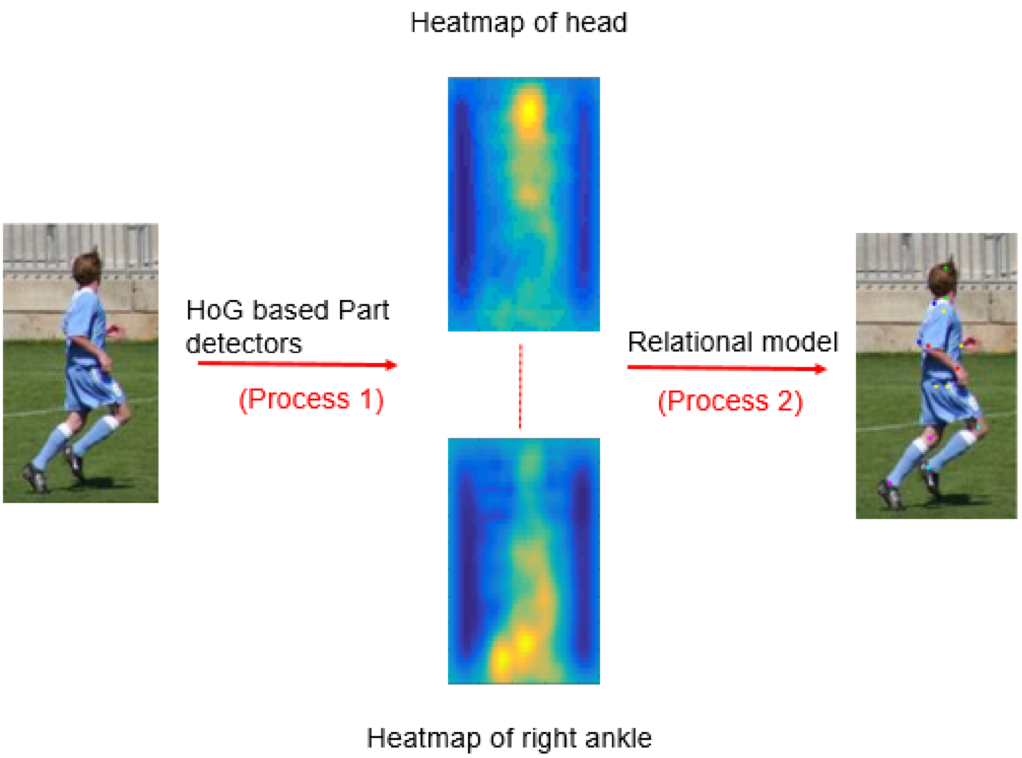


Fig. 2.1 A typical HPE framework for approaches before the use of CNNs.

The initial CNN-based frameworks (Section 2.3.2 - CNN as part detectors) acquired the same structure with the previous PS-based framework described above, including two basic processes to infer human poses (Figure 2.2). The difference lies in Process 1, where CNN was applied to detect body parts instead of the HoG-based detectors. Chapter 3 investigates Process 1 of the CNN-based framework with the aim of determining a network configuration that maximizes the body-part classification accuracy at the minimal computational cost. The factors considered in the optimization process are the network size, type of pooling scheme, and the application of transfer learning.

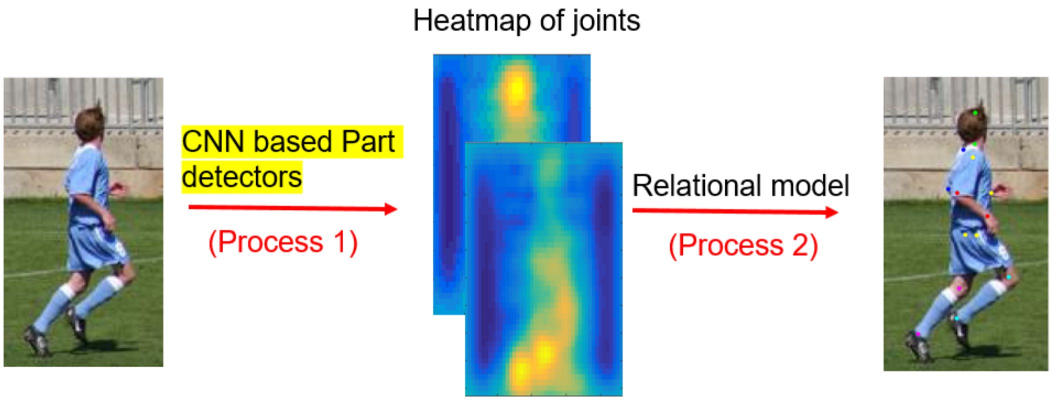


Fig. 2.2 A typical HPE framework that uses CNNs as body part detectors.

Later CNN-based designs embedded the two processes into CNNs [2, 21]. In other words, these CNN frameworks can both detect and encode body-parts relationships in a unified structure. Some of these frameworks focus on obtaining higher expressive power by stacking several CNNs and making CNNs deeper (Figure 2.3), which increase HPE accuracy significantly at the cost of high computational resources. The other frameworks (see Figure 2.4) increase the expressive power of CNNs by incorporating prior knowledge as graphs into CNNs. There is a number of different ways in which the human pose graph can be derived. Generally the human pose graphs can be divided into tree-based and non-tree-based graphs. This thesis explores and compares all three types of graph-based pose representations: tree, non-tree and a combination of both in Chapters 4, 5 and 6 respectively.

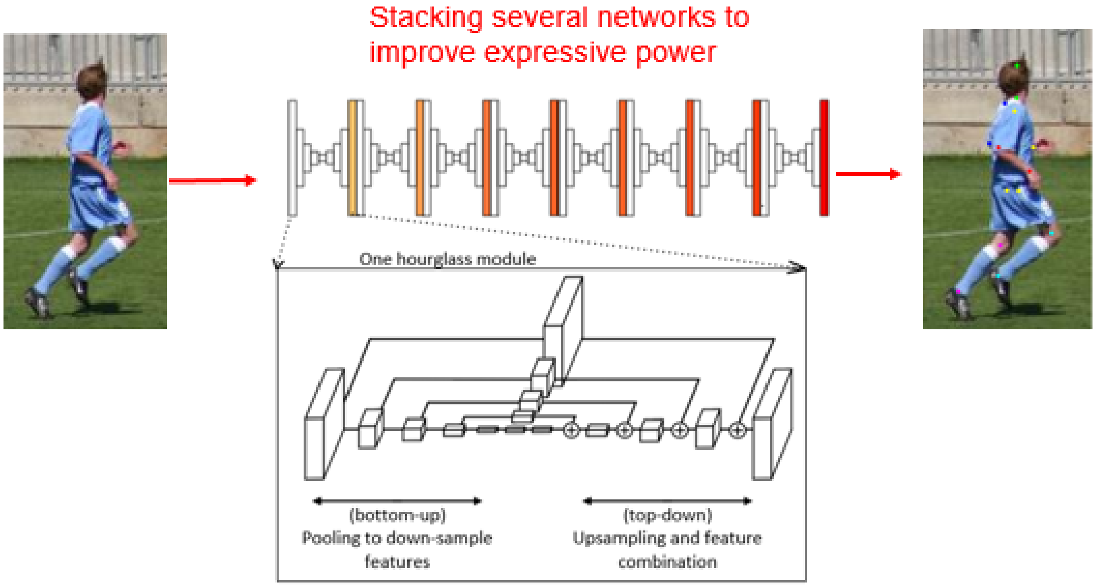

Stacking several networks to improve expressive power
One hourglass module
(bottom-up)
Pooling to down-sample features
(top-down)
Upsampling and feature combination

Fig. 2.3 An HPE framework with a stacked CNN structure.

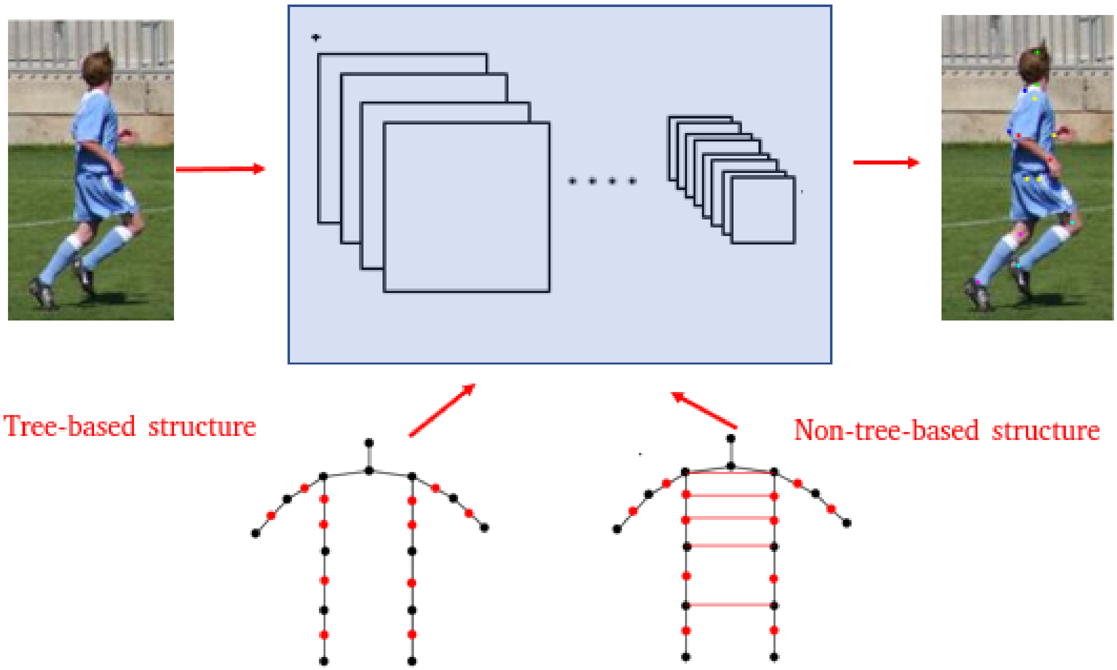


Fig. 2.4 An HPE framework with prior graph knowledge incorporated into a CNN.

2.2.3 Datasets

There are various benchmark datasets used to validate HPE techniques. The most popular datasets are as follows: the Leeds Sports Pose (LSP) [8], the LSP extended [28], MPII [29], the Frames Labeled In Cinema (FLIC) [30], the FLIC-full [30] and the Armlets [31]. More detailed descriptions of these datasets are provided in Table 2.1. These datasets are designed for a single-person pose estimation; only one pose annotation is provided in an image regardless of whether the image depicts a single person (in most LSP images) or a number of people (in most MPII images). Depending on the application, the above datasets can be used either on their own, or can be combined to create a larger dataset. For example, in the recent approaches [25, 32], HPE models were first trained on the MPII and the LSP Extended dataset and then fine-tuned on the LSP dataset. On the other hand, the HPE systems proposed in [21, 22, 33] were trained and tested on the LSP dataset only.

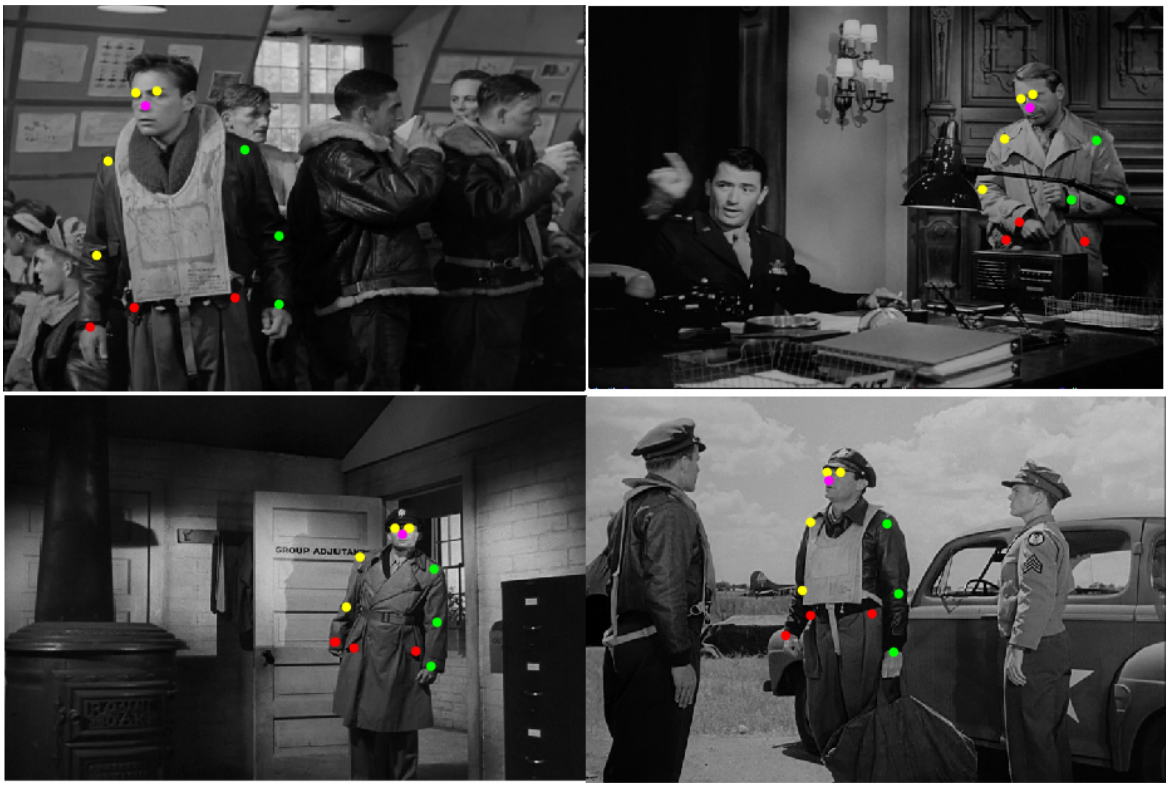


Fig. 2.5 FLIC dataset [30] and its uppper-body or 10-joint annotation.

Table 2.1 Datasets for 2D human pose estimation.

Datasets	Number of images	Description	The number of annotated joints
LSP [8]	2000	Images of sports people gathered from Flickr. People in these images are adjusted to 150 pixels in length. Full-body annotation	14
LSP extended [28]	10000	Images of ’parkour’, ’gymnastic’, ’athletic’ people gathered from Flickr. People in these images are adjusted to 150 pixels in length. Full-body annotation	14
MPII [29]	40522	Images of every day people activities extracted from YouTube video, covering 410 human activities. Full-body annotation.	15
FLIC [30]	5003	Images gathered from Hollywood movies. Upper-body annotation	10
FLIC-full [30]	20928	Images gathered from Hollywood movies. Upper-body annotation.	10
Armlets [31]	12589	Images gathered from Flickr. Upper-body annotation.	-

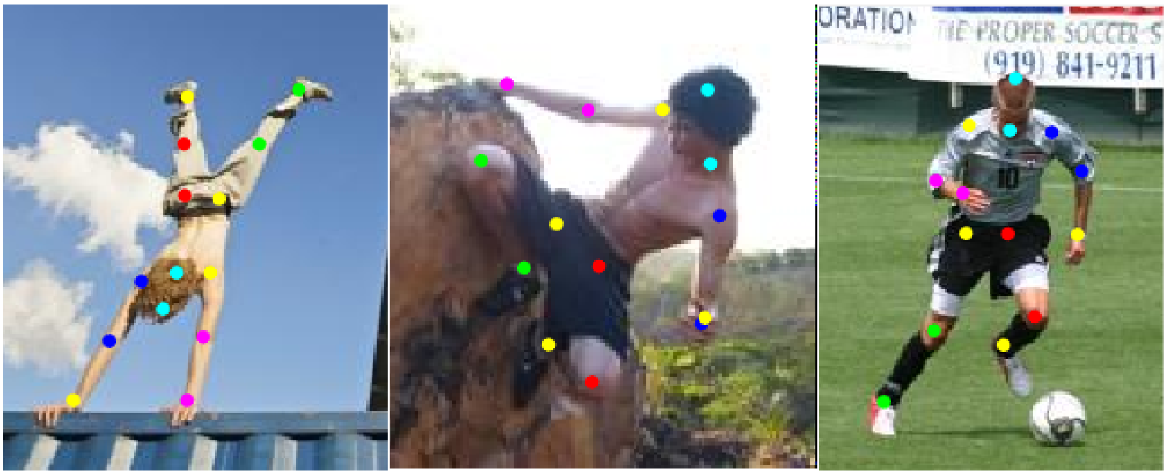


Fig. 2.6 LSP dataset [8] and its full-body or 14-joint annotation.

Among the datasets listed in Table 2.1, the LSP [8], LSP Extended [28] and MPII [29] are the most popular due to the high number of training images, as well as the full-body annotations. Datasets such as the FLIC contain only upper-body annotations. The amount of training images plays an important role in a CNN-based HPE model, since it improves the generalization of a CNN network. The FLIC, FLIC-full and Armlets are large datasets. However, they contain only upper-body annotations (Figure 2.5). Experiments in [29] demonstrated that full-body based approaches performed better than those using upper-body annotated images only. Therefore, upper-body datasets (Figure 2.5) are overall less preferred than full-body datasets (Figure 2.6). With regard to the dataset content, the MPII dataset [29] was created from YouTube videos showing everyday activities. The images are depicting multiple people with a high-scale of variation and a large number of occlusions. On the other hand, the LSP [8] and LSP Extended datasets [28] featured sports people in complex poses within a standardized length of approximately 150 pixels.

The ways of annotating occluded joints vary from dataset to dataset. For instance, the LSP Extended and MPII dataset specify whether a joint is visible or not. For the LSP Extended, if a joint is specified as invisible, no location for that joint is provided. On the other hand, the annotation of the LSP dataset is provided for all joints but does not contain joint invisibility information. As a result, different training strategies are required to manage the occluded joints in these datasets.

2.2.4 Evaluation

Given a set of test images, to determine whether the predicted locations of joints are correct or not, ground-truth information about the actual positions of joints is needed for the comparison. The comparison results calculated for all test images are then averaged to obtain the HPE accuracy. Different metrics and evaluation protocols are used to determine the HPE accuracy of a system. Widely accepted metrics include: Percentage of Correct Keypoints (PCK) [33], Percentage of Detected Joints (PDJ) [30], Percentage of Correct Parts (PCP) [34] and strict PCP [1]. While the PCK metrics uses the overlapping of keypoint bounding boxes as a measure to determine the matching, the PDJ metrics considers a body part as detected if the distance between the detected endpoints and ground-truth endpoints is smaller than a fraction of the torso's diameter. On the other hand, the PCP metrics considers a body part as detected if the distance between the detected endpoints and ground-truth endpoints falls within half of the body part's length.

Strict PCP evaluation

Strict PCP [1] is a commonly used metric to assess HPE accuracy and it has been applied in all experiments described in this thesis. The strict PCP evaluation accounts only for the highest scoring estimation. Namely, a body part is considered to be correctly identified if the relative distance between its estimated endpoints and the ground-truth endpoints is less than 50% of the head length, which is the distance between the head and the neck's keypoints.

For example, in the LSP dataset [8] each of the test images is accompanied with a 14-joint annotation. The annotation specifies the actual "true" locations of the person's body parts depicted in the image. These images can be hypothetically tested by an HPE system, which generates estimated or predicted locations of 14 joints. An example of the HPE accuracy results that could be given by the HPE system applying the strict PCP metric is shown in Table 2.2.

Table 2.2 An example of HPE accuracy in percentage (%) (using strict PCP evaluation protocol).

Configurations	**Head**	**Torso**	**Upper arm**	**Lower arm**	**Upper leg**	**Lower leg**	**Mean HPE**
TA_14	88	87.7	71.5	59.5	78.2	72.2	73.9
TA_26 ([2])	89.2	93.9	76.4	63.9	85.7	80.3	**79.6**

Given the ground-truth and estimated joint locations, the body-part representation is formed by grouping joints that belong to specific body parts.

Figure 2.7 illustrates an example showing how the ground-truth joints can be grouped into body parts. There are 10 body parts in this example, and each body part is represented by two joints. The left low leg (L.L. leg) can be formed by grouping joints 7 and 8, and the left upper leg (L.U. leg) by grouping joints 6 and 7.

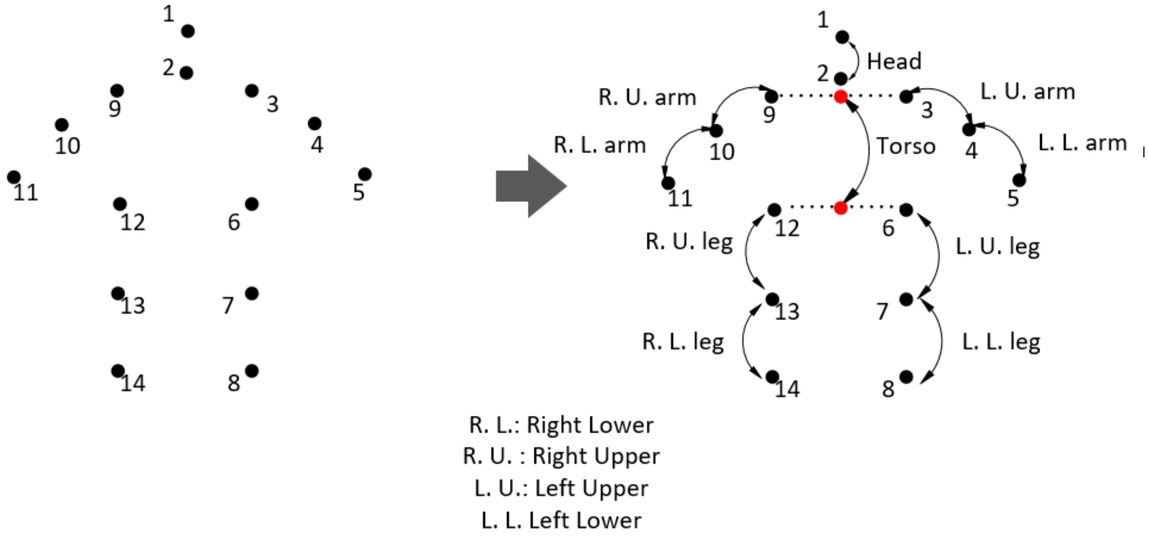


Fig. 2.7 Body parts in strict PCP.

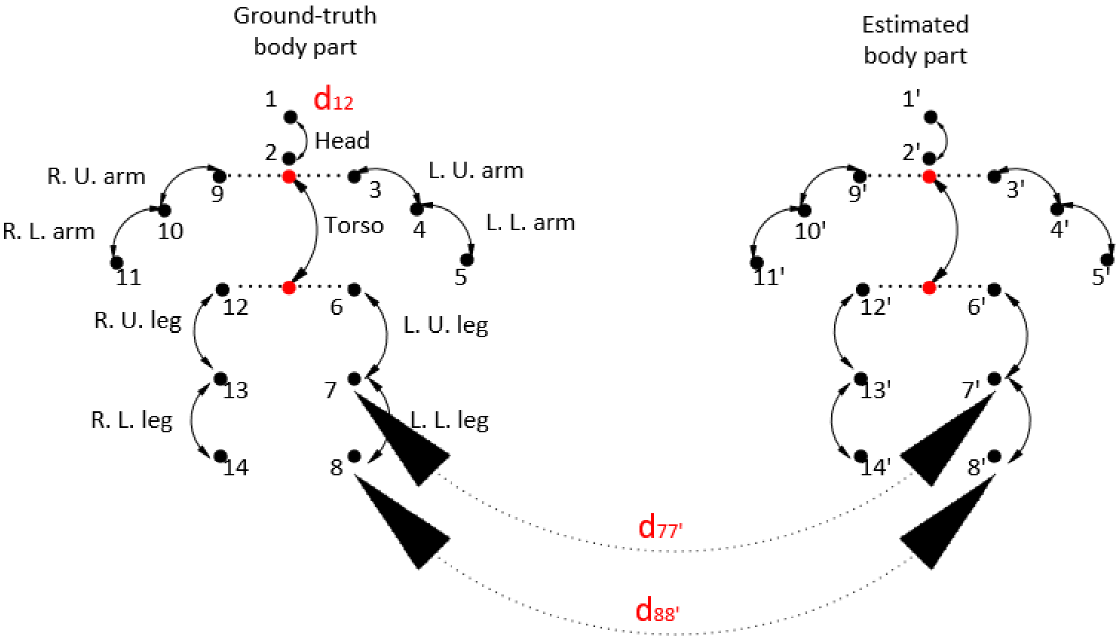


Fig. 2.8 Strict PCP protocol [1].

Ground-truth and estimation data for the strict PCP protocol is shown in Figure 2.8. The following equation determines whether a ground-truth body part matches an estimated body part:

$$\begin{cases} \frac{d_{ii'}}{d_{12}} < 0.5 \\ \frac{d_{jj'}}{d_{12}} < 0.5 \end{cases} \tag{2.1}$$

In Equation 2.1, i and j denote joints i and j of a ground-truth body part (ij), while i' and j' specify corresponding joints i' and j' of an estimated body part (i'j'). d_{12} represents the distance between joint 1 (head joint) and joint 2 (neck joint) of the ground-truth data. Body part (ij) matches body part (i'j') if Equation 2.1 is satisfied. When applying Equation 2.1 into the example illustrated in Figure 2.8, the left lower leg (L.L. leg) of the ground-truth can be considered matched with the L.L. leg of the estimated body part if $\frac{d_{77'}}{d_{12}} < 0.5$ and $\frac{d_{88'}}{d_{12}} < 0.5$. Where, $d_{77'}$ is the distance difference between joint 7 (of the ground-truth data) and joint 7' (of the estimation data) and $d_{88'}$ is the distance difference between joint 8 (of the ground-truth data) and joint 8' (of the estimation data).

2.3 Literature review

The use of CNNs led to a significant advancement in HPE technology. In comparison with other conventional techniques, the CNN-based methods were shown to obtain significantly higher HPE accuracy when tested on standard benchmark datasets [35]. Given their high performance and existing potential for further improvement, this thesis is focused on the CNN-based methods for HPE. The following sections aim to justify this choice. This is done by showing literature-based evidence of high CNN performance, and by identifying existing potential for improvement of the CNN-based approaches to the HPE problem.

The literature review discusses traditional HPE approaches (**Section 2.3.1**) and CNN-based approaches (**Section 2.3.2** and **2.3.3**). The approaches in **Section 2.3.2** are in chronological order while the ones in **Section 2.3.3** are grouped based on common structures. A majority of start-of-the-art HPE results, mostly from 2017, apply successive predictions or network stacking (**Section 2.3.3. Successive predictions**), which demand intensive training with GPUs [20, 25]. The mean HPE accuracy from this research direction is quite high, of more than 90%, probably reaching to a saturation point of HPE accuracy. Due to the high result, research activities in the field of HPE in 2D images have been less active since 2018 and 2019. On the other hand, multi-person HPE [36–38], HPE in videos [39–41] and 3D HPE [42–45] are still popular. This research investigates the techniques in **Section 2.3.3 - CNN-based message passing**, which demand lower computational cost.

2.3.1 HPE before the introduction of CNN

Before the introduction of CNN, human poses were estimated using holistic or part-based approaches. Holistic approaches were less popular, and were applied mostly in cases when there were either a small amount of training images or a need to deal with rendered images. On the other hand, part-based approaches were more common and had become the leading pre-CNN technique for HPE.

Holistic approaches for HPE

Holistic approaches consider full-body pose estimation as a whole. Mori et al. [46] applied a holistic approach by matching a test body shape with a database of exemplars using shape context descriptors. The matching process is illustrated in Figure 2.9.

Firstly, internal and external contours of a test body shape were extracted using an edge detector (Figure 2.9, stage (1)). Then, these contours were encoded by shape context descriptors. Similar procedures were performed for each exemplar in the training data. For each exemplar with provided keypoint locations, points on detected contours were transferred to kinematic chain segments, including the torso, upper and lower arms, and upper and lower legs, as seen in Figure 2.9(a)). Each exemplar with kinematic chain segments would deform by translation of the torso and 2D rotations of limbs around the shoulders, elbows, hips and knees to match with the shape context descriptors of the test data (Figure 2.9, stage (3)). Keypoint locations also moved in synchronization with this deformation. When a match was found, the keypoint locations from the corresponding exemplar was transferred to the test shape. Classical methods for exemplar-based matching are the K-nearest neighbour rule [47] and local weighted regression [48].

Although these techniques are relatively simple, their effectiveness reduces when dimensions and the number of input data increases. Therefore, Shakhnarovich et al. [49] applied the local sensitive hashing algorithm to estimate the pose of an input image quickly, tackling the issue of exhaustive search in a large database. Observing that discriminative classifiers, namely the support vector machine, performed better than the nearest neighbor methods and there were other features that were more descriptive than the edge features, Gkioxari et al. [31] combined the holistic approach with these modern classifiers and feature technologies to estimate arm configurations. This system used highly discriminative classifiers and rich feature representations, including HOG, contours and skin color.

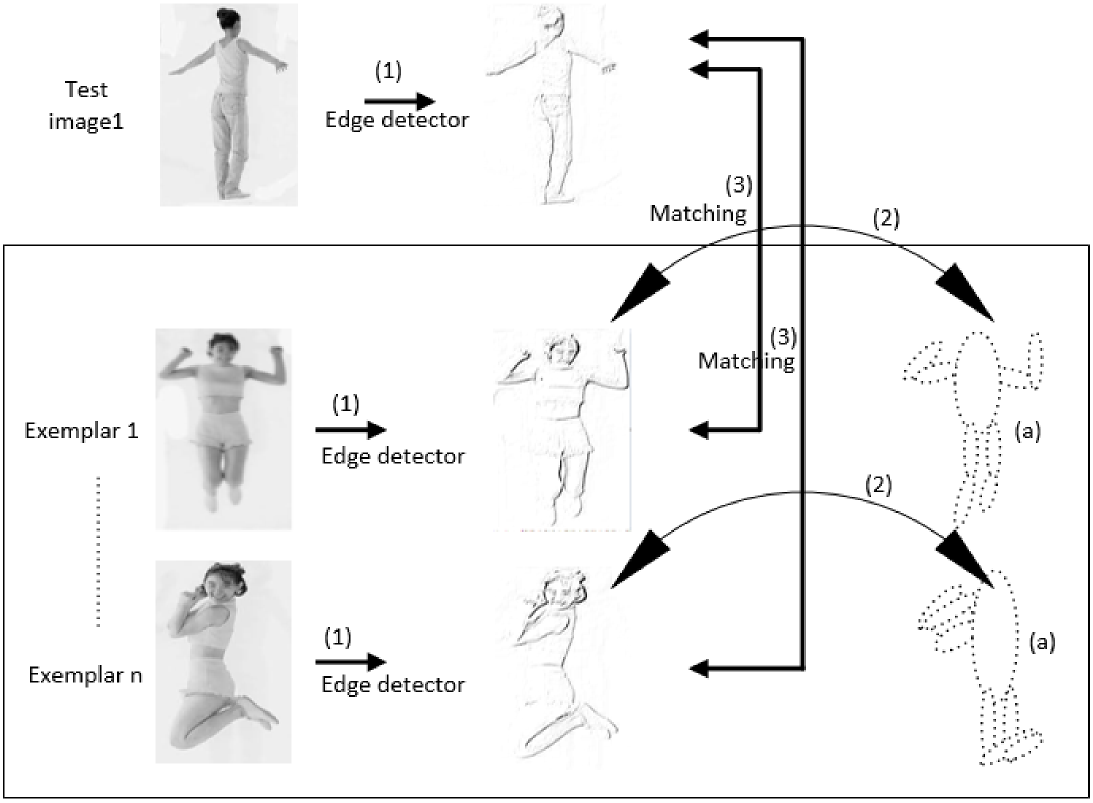


Fig. 2.9 Human Pose Estimation using shape context matching (adapted from [46]).

Part-based approaches for HPE and the Pictorial Structure framework

Part-based approaches have been the leading techniques for 2D human pose estimation prior to the introduction of CNNs [30, 50–57]. This approach learned two models separately, one for part detectors and the other for part-part relationships.

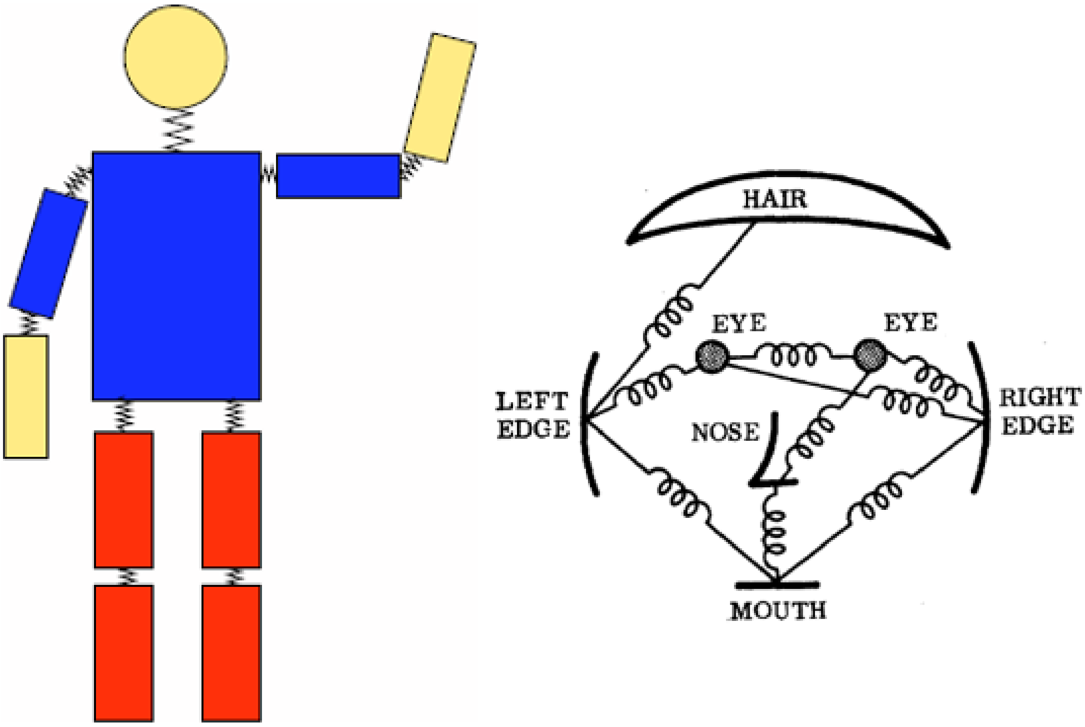


Fig. 2.10 Pictorial Structure (PS) for human body and human face (adapted from [26]).

A typical framework to model human poses in a part-based setting is the Pictorial Structure (PS), first introduced by Fischler and Martin [26]. This framework modeled an object as a group of parts connected in a deformable configuration. Each part represented local visual information of the object and the deformable configuration was featured by spring-like connections between pairs of parts as illustrated in Figure 2.10.

The PS framework can be represented as an undirected graph G = (V,E), where $V = \{v_1, v_2, ...v_n\}$ specifies n parts and E denotes connected pairs of parts $(v_i, v_j \in E)$. Each object instance is referred to as L = $l_1, l_2, ...l_n$ where l_i denotes the position of part v_i. Finding part locations of an object L is equivalent to minimizing an energy function $f(L)$ given as follows:

$$f(L) = \sum_{i=1}^{N} m_i(l_i) + \sum_{v_i, v_j \in E} d_{ij}(l_i, l_j) \tag{2.2}$$

Where $m_i(l_i)$ measures the incompatibility level of placing part v_i at the location l_i and $d_{ij}(l_i, l_j)$ measures the deformable degree of placing part v_i at l_i and part v_j at l_j. There are several problems with this original PS structure [26], including the high number of model parameters and only one single best result is obtained. Felzenszwalb et al. [58] addressed the problem by introducing a statistical approach into the PS structure. Additional improvements included methods for obtaining several good hypotheses and learning the PS model from training examples.

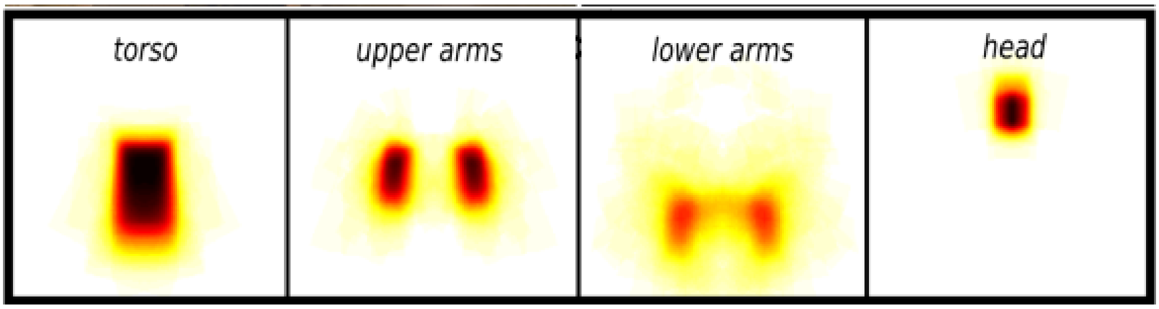


Fig. 2.11 Location priors for better appearance model (adapted from [59]).

Noticing that the PS framework contained two key elements, which were a part detector, and a part-part relationship, later approaches focused on either learning a good part detector (or part appearance model), or obtaining a good part-part relationship.

Learning good part detectors Part detectors are obtained based on visual information derived from pictures of body parts. To better encode the information, different image features and feature encoding techniques are applied as seen in Figure 2.12. Image features vary from image silhouette [61] for person segmentation from background, to color [62] for modeling skin and clothing, to edge [63] for extracting body contours, and to gradients [64] for obtaining body texture. However, these features are subject to noise and were in high dimensions. Therefore, they are often encoded by image descriptors such as Histogram of

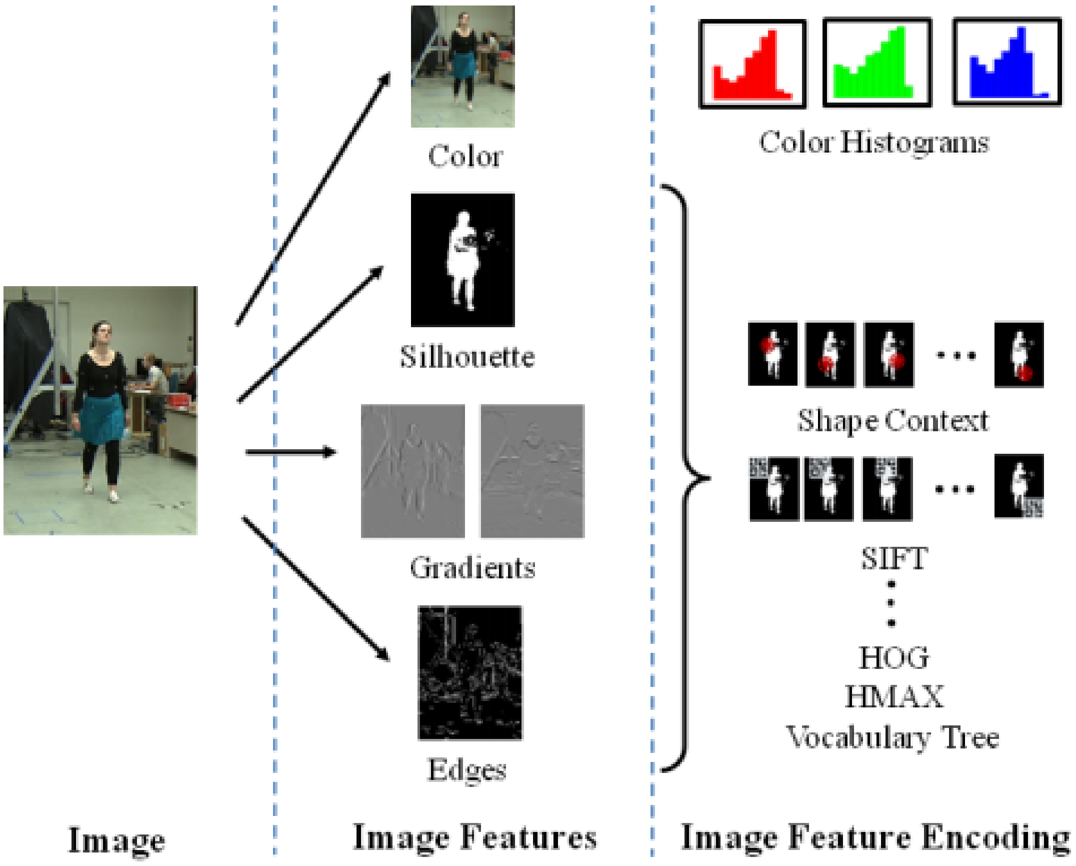


Fig. 2.12 Common image feature and encoding methods (adapted from [60]).

Oriented Gradients (HOG) [27], Scale Invariant Feature Transform (SIFT), or shape context [61, 65] to decrease dimensionality and increase robustness to noise.

Ramanan [62] obtained a good appearance model using an iterative parsing method based on edge features. From the initial parse estimated by an edge-based detector, the system routinely built better features from previous parsing data. Eichner, et al. [59] created an appearance model by exploiting latent relationships between the appearance of various body parts. By observing that relative locations of body parts to a detection window had patterns, for example, the torso was often positioned in the center of an upper-body detection, and the appearance of some body parts were related, the location distribution of body parts with regard to detection windows (or location priors) was learned and could be incorporated into existing pictorial structure engines. Figure 2.11 illustrates the probability distribution of the torso, upper arms, lower arms, and head obtained from training data. These locations would then be used in combination with the part appearance model. Andriluka, et al. [65] built strong part detectors without the use of an iterative parsing method or search space reduction. The detectors (or the part appearance models), were learned by using shape context descriptors and AdaBoost [66] classifiers. Dense evaluation and bootstrapping were performed on these detectors to improve performance.

In fact, most of the above-mentioned methods tried to improve part appearance model based on a single type of image features such as silhouettes, edges or gradients. To further improve the choice of image features, Sapp, et al. [67] introduced a cascaded model combining different image features including contours, regions, textures and colors for the appearance model. A single type of image feature was not enough to provide strong appearance cues, especially in the case that image quality is degraded resulting in poor localization and confusion of parts on a clustered background. This cascaded model was able to evaluate complex appearance models densely. Each level of the cascade used inference to find states needed to prune away, which helped to reduce the number of state spaces dramatically.

Learning good part-part relationships Part-part relationships (or spatial relationships among parts) function as part constraints to refine and remove false positives from part detectors. This spatial relationship can be modeled as a tree or non-tree-based configuration. Tree-based models of human poses were first proposed by Felzenszwalb and Huttenlocher [68] and had been used in part-based approaches to model pairwise relationships between adjacent human body parts. To capture a larger range of pose variations, a global mixture of trees [69] or a mixture of local parts for each tree node [33] was introduced. One disadvantage of the tree representation is the inability to model complex poses, as only the pairwise interactions between nearby parts are captured.

Several non-tree-based representations were proposed to model spatial body-part relationships beyond pairwise links. Wang, et al. [70] proposed a non-tree-based structure (or a loopy graph) to model high-order relationships between body parts. However, the loopy graphs used approximate inference, which lost the exact inference benefits provided by the tree-based structures. This limitation was overcome by the hierarchical tree structure with latent nodes introduced by Tian, et al. [71]. Jiang, et al. [72] combined tree and non-tree structures in a graph representation where strong (tree) edges enforced arbitrary constraints and weak (non-tree-based) edges expressed the mutual exclusivity of inter-part occlusions and symmetric conditions. To further encapsulate the complexity of relations between body parts, Tran, et al. [73] proposed a universal relation model of body parts by creating a comprehensive set of the dependencies of body parts. A hierarchical structure of body parts was proposed in [70, 71], modeling both single rigid parts, e.g. torso, head, wrist, as well as parts that contained more than one rigid element.

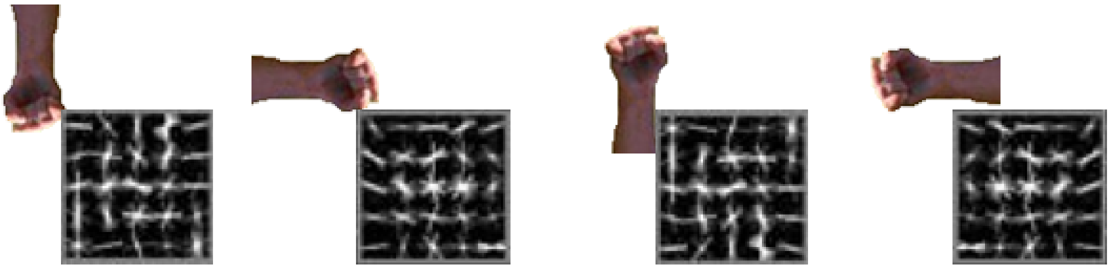


Fig. 2.13 A mixture of hand types (adapted from [33]). Each type representes an orientation of the hand.

Mixture of models Another important finding in part-based approaches for HPE was the introduction of mixture models in which parts were clustered based on their appearances or relative orientations to nearby parts. Yang and Ramanan [33] proposed a novel approach based on the pictorial structure model, where body parts were represented by a mixture of templates, one template for each orientation. Each orientation was considered to be a mixture of parts, and was obtained by clustering relative positions of the part with respect to its neighboring parts. An example of the mixture of hand types is shown in Figure 2.13, where various hand orientations are represented by a different mixture type.

Eichner and Ferrari [74] created image clusters of body parts based on similar appearances using a color model. At the beginning, a pose detector scanned all images in a training set. After obtaining the estimated joint locations for the entire human body, sub-images

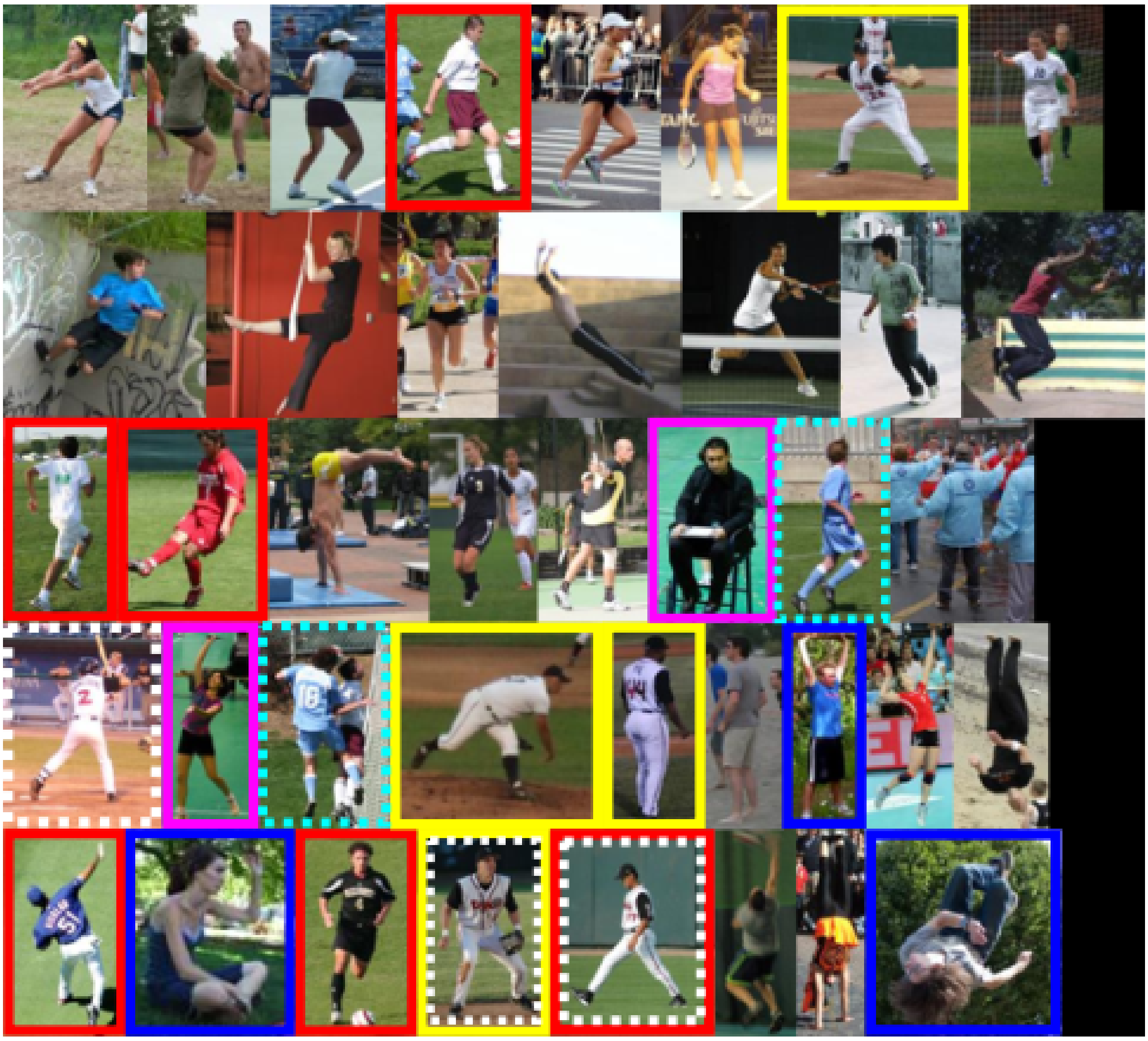


Fig. 2.14 Appearance Cluster (solid boxes for background, dashed for foreground with colors illustrate index of cluster) (adapted from [74]).

were extracted based on the estimated joint locations, and the system clustered similar body parts based on the color histograms of sub-images. Finally, color models for each part were estimated from the clusters, providing cues for refining the pose estimation. The above procedure generated a number of appearance mixtures as seen in Figure 2.14 where, red boxes show people with short trousers and yellow boxes show people wearing long trousers.

Poselets In part-based models, parts are often defined as basic rigid parts such as the head, torso, left arm, right leg, etc. However, parts in this basic definition do not always capture the

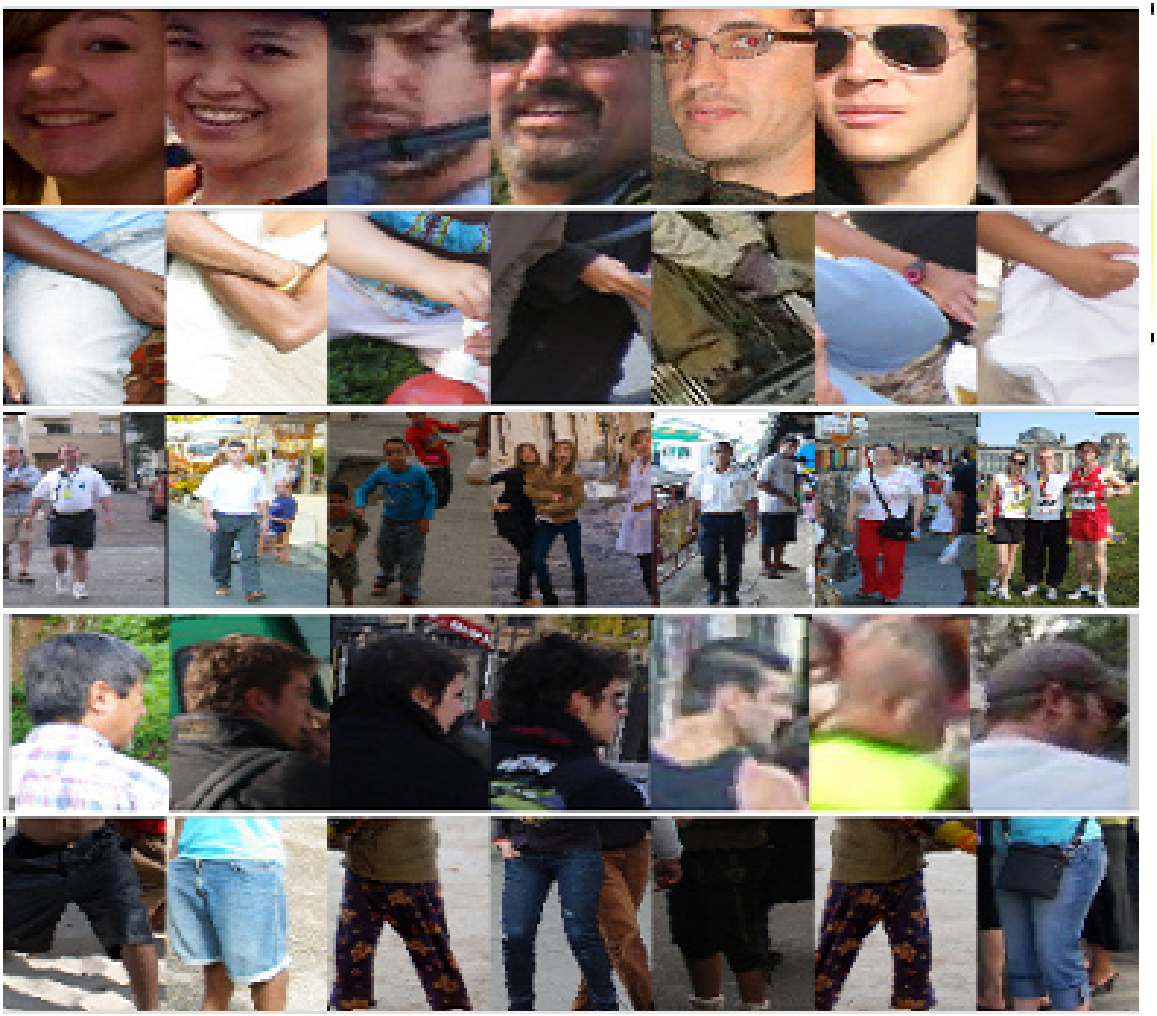


Fig. 2.15 Examples of poselets (adapted from [75]).

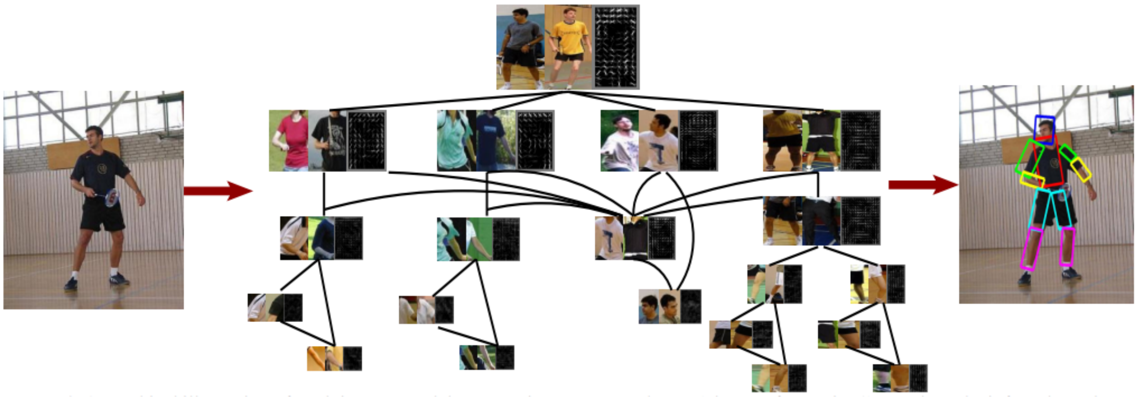


Fig. 2.16 Hierarchical poselets (adapted from [70]).

most important features for visual recognition. For example, limbs defined as rectangle or parallel lines can easily get confused with background objects. Another way of defining parts is image areas which cover large portions of the human body such as “a torso with crossed arms” or “a torso with kneeling legs”. Parts defined in the latter method are called “poselets”, which allow for modeling of dependencies between non-adjacent parts. Examples of poselets include the frontal face, right arm crossing torso, pedestrian, right profile and shoulder, as shown in Figure 2.15.

In a hierarchical representation proposed by [70], poselets were introduced to capture different levels of detail from small rigid parts to the whole body as seen in Figure 2.16. This presentation took into account both rigid parts and parts that captured large portions of a human body. Poselets and basic rigid parts could also be represented in a simple tree structure [76, 77], which enabled modeling large variations of human poses without losing the advantage of efficient inference.

2.3.2 CNN-based HPE

CNN as part detectors

Initial CNN-based frameworks for HPE are based on traditional PS structures [26] where CNNs function as part detectors [1, 78–80]. Jain et al. [78] were first to introduce an end-to-end approach for full-body HPE, where multiple convolutional networks were used for body-part classification instead of one network. The outputs were response-maps representing the probability distributions of body-parts. The resulting maps were then post-processed to remove false-positives using a high-level spatial model with simple body-pose priors. These priors were obtained by histograms of joint locations calculated over the training set. On the contrary, Chen et al. [1] used only one network for body part detectors, and human poses were modeled using a graphical model with novel pairwise relations. CNNs trained on local image patches of body parts not only encoded part appearance but also provided clues for pairwise relations. Figure 2.17 shows examples of pairwise relations between elbows and wrists. The left panel displays various possibilities of the elbow positions and the right panel contains different possibilities of the wrist positions. In the central panel of the figure, the local image patch of the elbow contributes to the pairwise relations by providing information for directions of its neighboring parts, which are the wrist and shoulder.

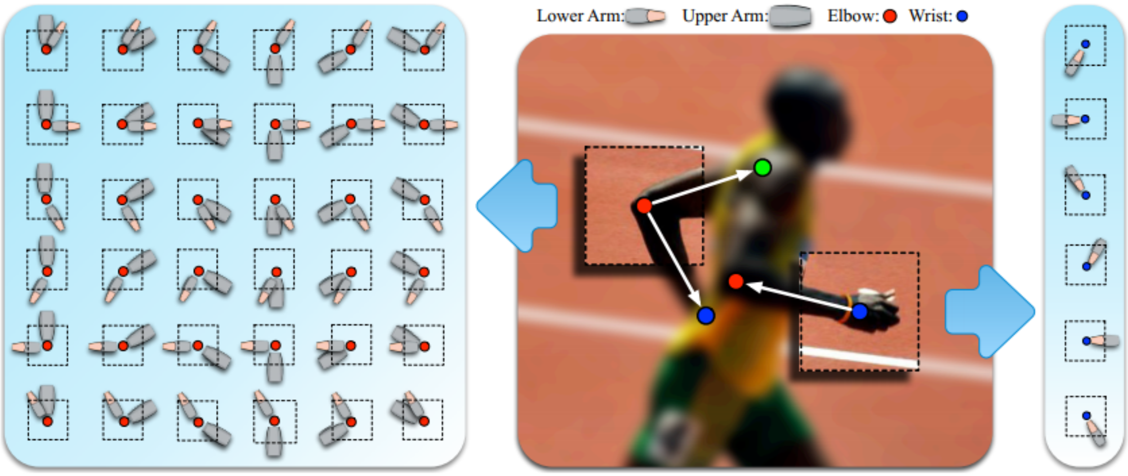


Fig. 2.17 Graphical model with novel pairwise relation (adapted from [1]).

CNN as part detectors and relational models

In the more recent HPE approaches, CNNs were applied to learn relationships between body parts. Tompson et al. [81] introduced a hybrid architecture combining a CNN and a Markov Random Field [82]. The CNN was designed to learn both part detectors and spatial relationships between body parts. To improve the scale-invariance, the CNN-based part detector was trained with two input image sizes 320x240 pixels and 160x120 pixels as shown in Figure 2.18. Feature sizes generated by these two image resolutions were different and a point-wise up-sampling was deployed to generate the same size features, so that they could be concatenated to form unified network inputs. In addition to the part detector module, the spatial relationship module applied the Markov Random Field [82] modelling to represent relationships between body parts' locations. In contrast to the hand-crafted spatial model proposed by [78], the spatial model by [81] was generated using DCNNs in combination with the message passing procedure conveying information generated by the part detectors. Although the training procedure for this highly-parametric model was computationally expensive, the model was capable of dealing with large spatial displacement of body joints when using sufficiently large convolutional kernels.

The issue of high computational complexity was addressed by Yang et al. [22], who introduced appearance mixtures and mixture of deformation constraints. This introduced system is shown in Figure 2.19 where, "T_i types" denotes a mixture of deformable constraints. This figure also illustrates two message passing layers (u^1 and u^2) encoding the spatial relationships between body joints in a loopy graph (or non-tree-based model). In the loopy graph, messages were passed simultaneously across every link at each iteration. The first iteration used the unary potential ϕ representing body appearance features as input to generate part belief u^1, which was then refined in the second iteration for the belief u^2.

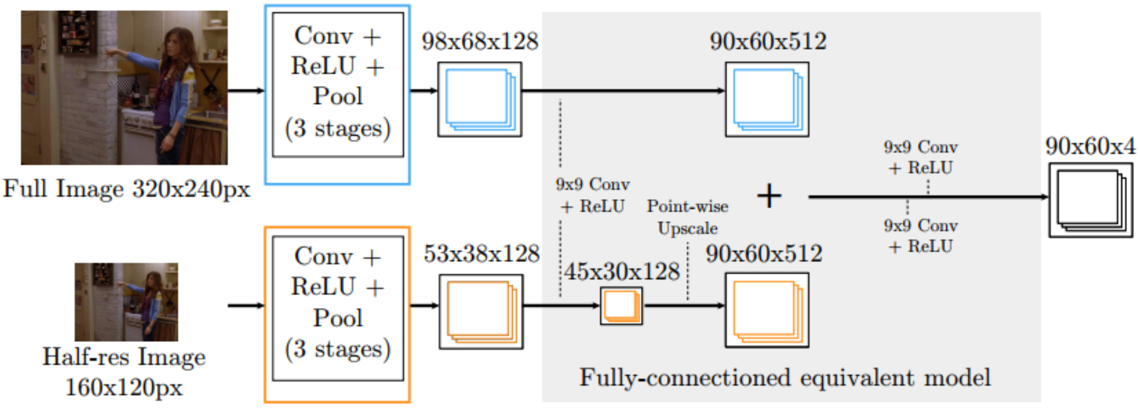


Fig. 2.18 Training with two input image resolutions (adapted from [81]).

The method described in [22, 81] learned pairwise relationships between body joints from score maps. This system first trained the part detectors separately and then stored the heat-map outputs, which were later used to train a spatial model. On the other hand, Chu et al. [2] learned the spatial model at the feature level as shown in Figure 2.20. Body part features were derived (Figure 2.20 (top)), and then refined through the structured feature learning module shown in Figure 2.20 (bottom), by being passed in a bi-direction tree. Both processes (feature calculation and refinement) were conducted at the feature level.

As opposed to the part-based approach proposed by [1, 22, 78], Toshev [83] estimated human poses in a holistic manner by introducing a cascade of DNN regressors. The task

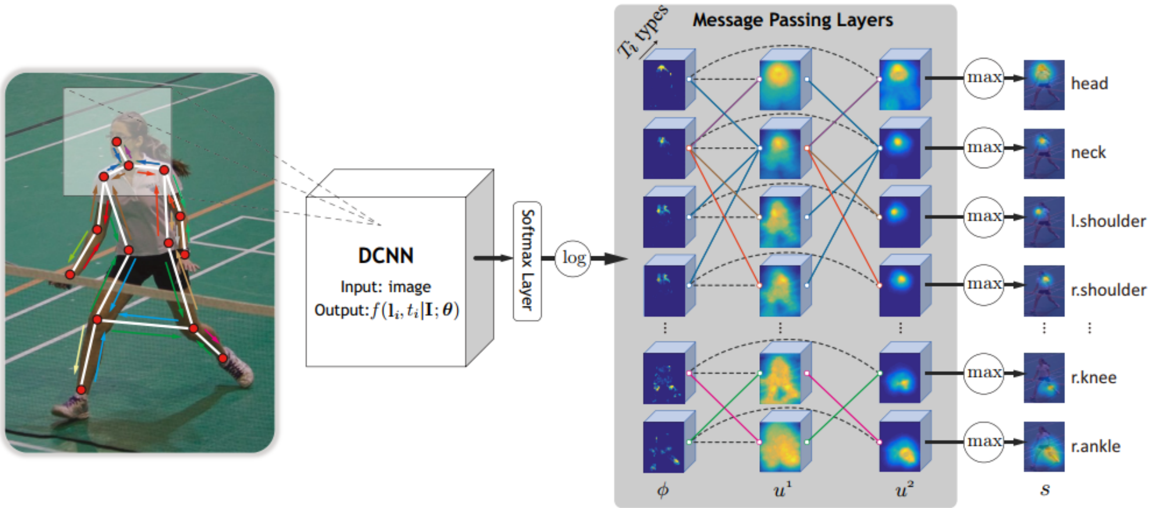


Fig. 2.19 Non-tree-based message passing (adapted from [22]).

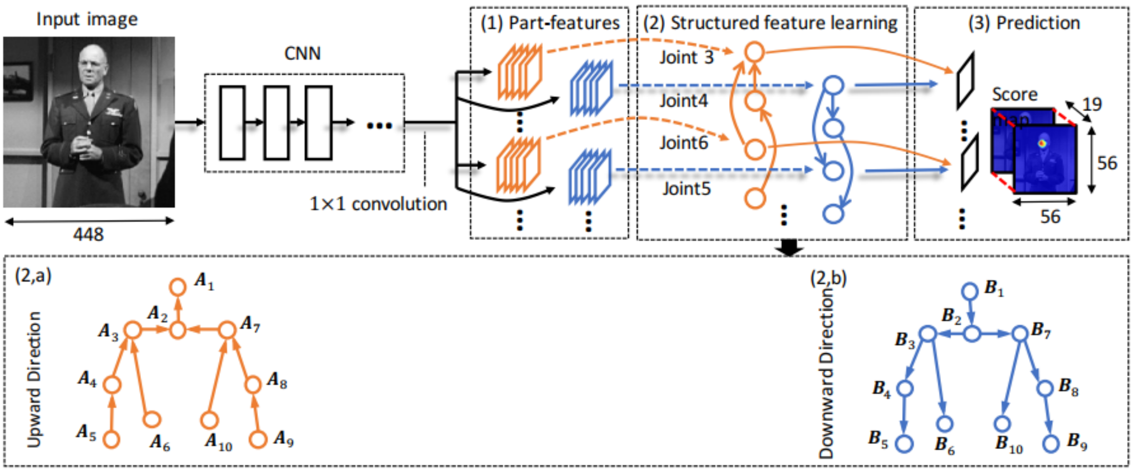


Fig. 2.20 Structured feature learning (adapted from [2]).

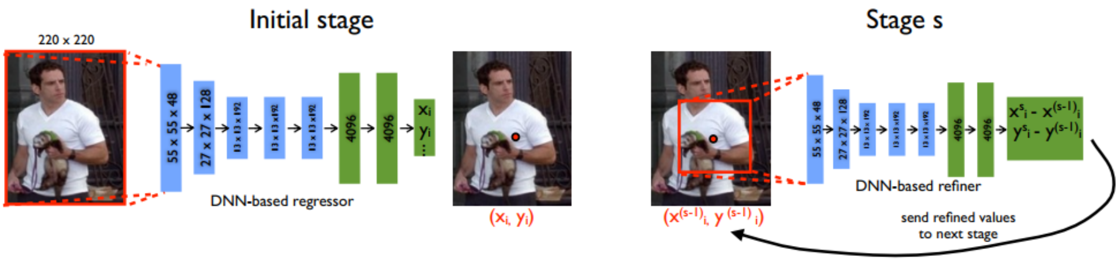


Fig. 2.21 DeepPose system (adapted from [83]).

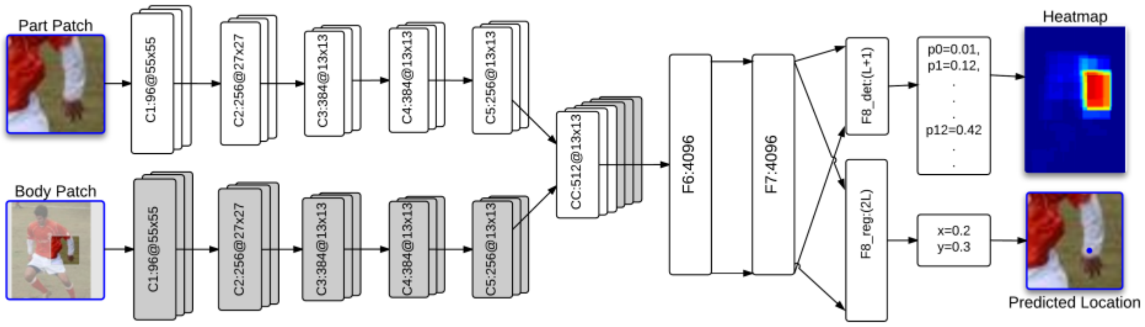


Fig. 2.22 Dual-source CNN (adapted from [84]).

of HPE was formulated as a joint regression problem with full-size images as input. The advantage of training whole images is the rich expressiveness as the full context of body parts can be captured. The DNN-based regressors are illustrated in Figure 2.21. Rough poses were estimated at the initial stage using the fixed input size of 220x220. Then, the pose regressors were trained on image patches cropped around predicted points of the previous stages so that they could learn displacements of joint locations from the previous stages to the ground-truth locations. Since the regressors were applied on sub-image regions, they saw higher image resolutions and thus, higher precision. However, this system did not consider local appearance in initial pose estimation. This limitation was addressed by Fan et al. [84] in a dual-source CNN for HPE taking into account the local appearance of each body part and the global view of the whole body. The complete CNN system as shown in Figure 2.22 consisted of two CNN sequences. One took input as part patch (image patches containing a body part) and the other as body patch (images which showed the whole body). In addition to the usual joint location task, the CNN was also designed for joint detection to obtain a complementary effect. These two sequences were then stacked together and both the joint regression and detection were applied to the whole network.

CNN with successive predictions and stacked networks

Recent CNN-based approaches propose successive prediction structures or making deeper CNNs to achieve higher expressive power. Although this research direction lead to very high HPE performance, the data and computational costs were extremely high, making the use of graphics processing units (GPUs) paramount and limiting possibilities of practical applications.

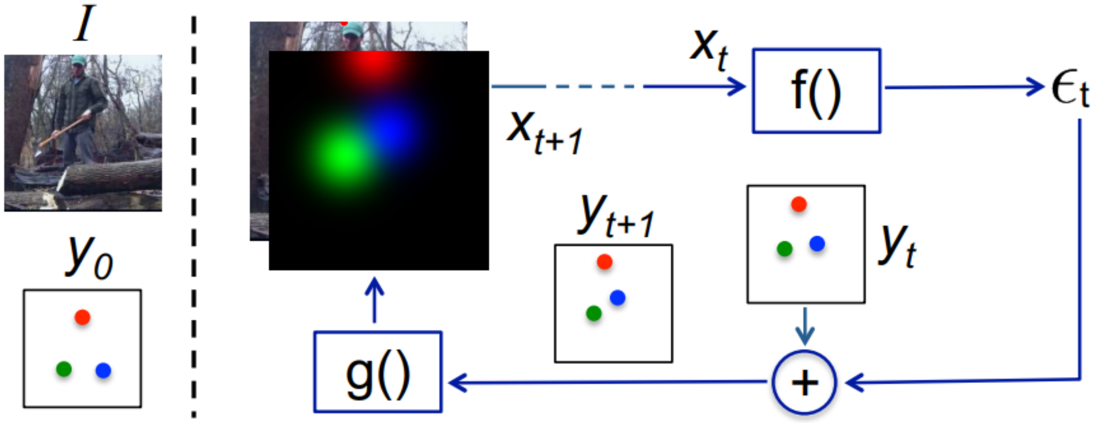


Fig. 2.23 Iterative error feedback (adapted from [85]).

Carerra et al. [85] expanded the expressiveness of a CNN by introducing a self-correcting model which encompassed both input and output space. The model contained a feedback loop that could iteratively refine previous estimations (Figure 2.23), thus creating successive predictions. In this figure, I and y_o represented the input image and ground-truth keypoint positions; x_t and y_{t+1} were the input data and keypoint positions of the iteration t; function $f()$ denoted a convolutional network; and function $g()$ worked as a converting function from 2D keypoint positions to Gaussian heatmap channels. At the iteration t, the function $f()$ generated a correction ε_t from input x_t stacked with Gaussian heatmap of keypoint positions y_t. Then, the keypoint positions for the next iteration y_{t+1} were obtained by adding y_t to the correction ε_t, and x_{t+1} was obtained by concatenating x_t and the Gaussian heatmap of y_{t+1}. Later iterations continued in this manner.

Similar to the successive predictions introduced by [85], Newell et al. [20] proposed a network consisting of a stack of several hourglass networks. An intermediate supervision was applied in-between individual networks. As stacking networks dramatically increases the depth of the CNN structure, and makes it more prone to the vanishing gradient, the incorporation of intermediate supervision tends to reduce the vanishing gradient effect.

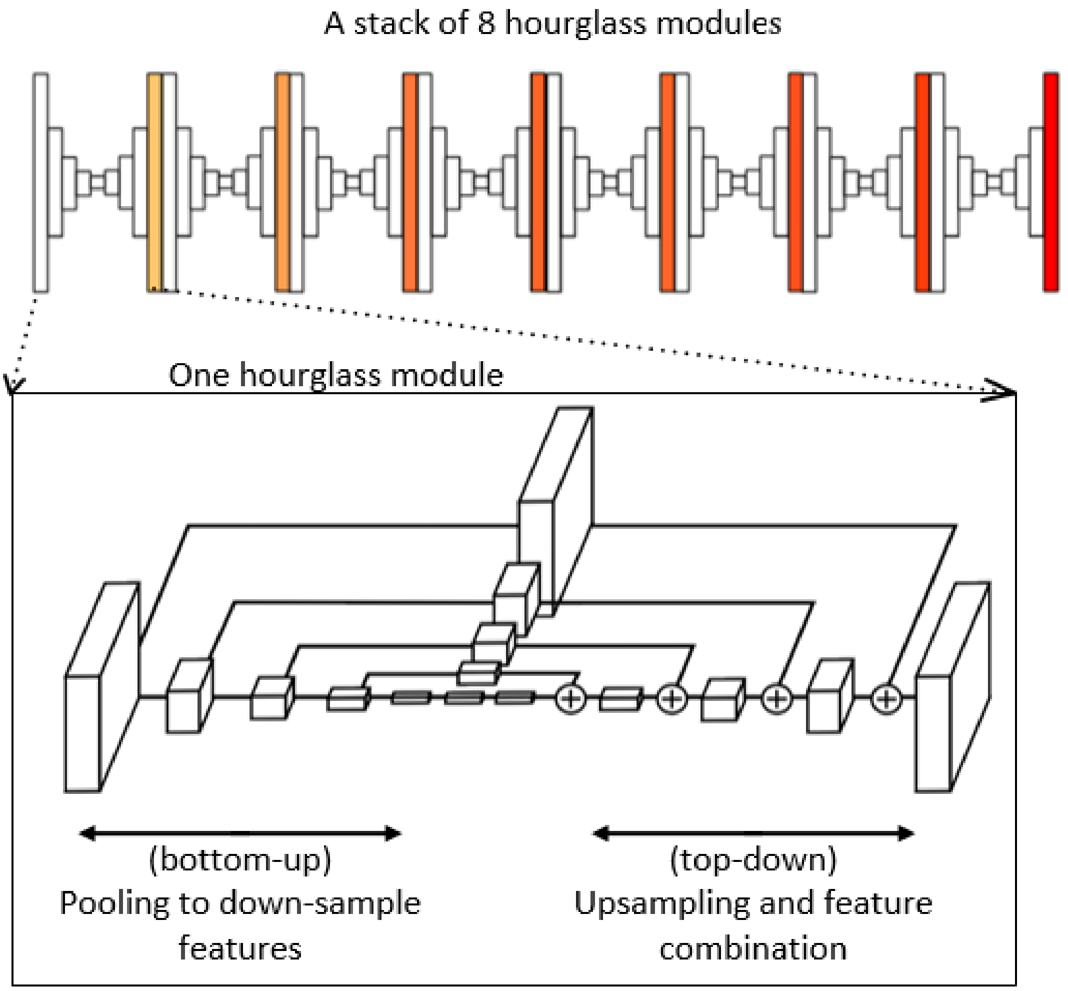

A stack of 8 hourglass modules
One hourglass module
(bottom-up)
Pooling to down-sample features
(top-down)
Upsampling and feature combination

Fig. 2.24 A stack of 8 hourglass modules (adapted from [20]).

Another feature in each of the hourglass networks was the symmetric design capable of capturing information at every scale. Figure 2.24 shows stacked hourglass modules, each of which demonstrates repeated bottom-up (using pooling for feature down-sampling) and top-down (with up-sampling and feature concatenation) processing. The repeated down-sampling and up-sampling, together with feature concatenation, enable features to be processed at various scales.

Butlat et al. [24] proposed a CNN cascaded architecture to effectively learn part relationships and the spatial context. The architecture includes a detection network followed by a regression network, as shown in Figure 2.25. The detection network generated part heatmaps

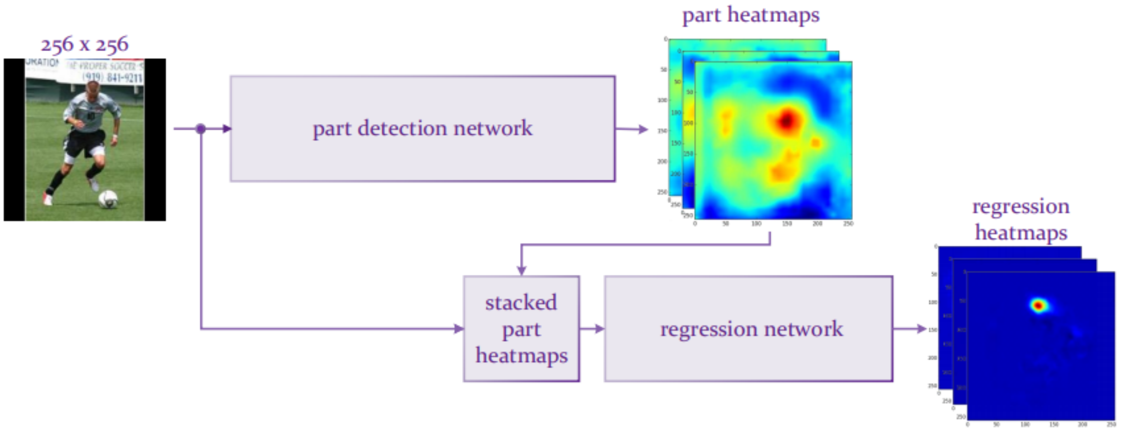


Fig. 2.25 Part heatmap regression (adapted from [24]).

showing the probabilistic distribution of body parts, while the regression network regressed these heatmaps concatenated with the input image. In the training process, the detection network was trained separately, then both the detection and regression networks were trained jointly afterwards. Although this system consisted of only two network components, its HPE accuracy was comparative to the eight hourglass networks proposed in [20].

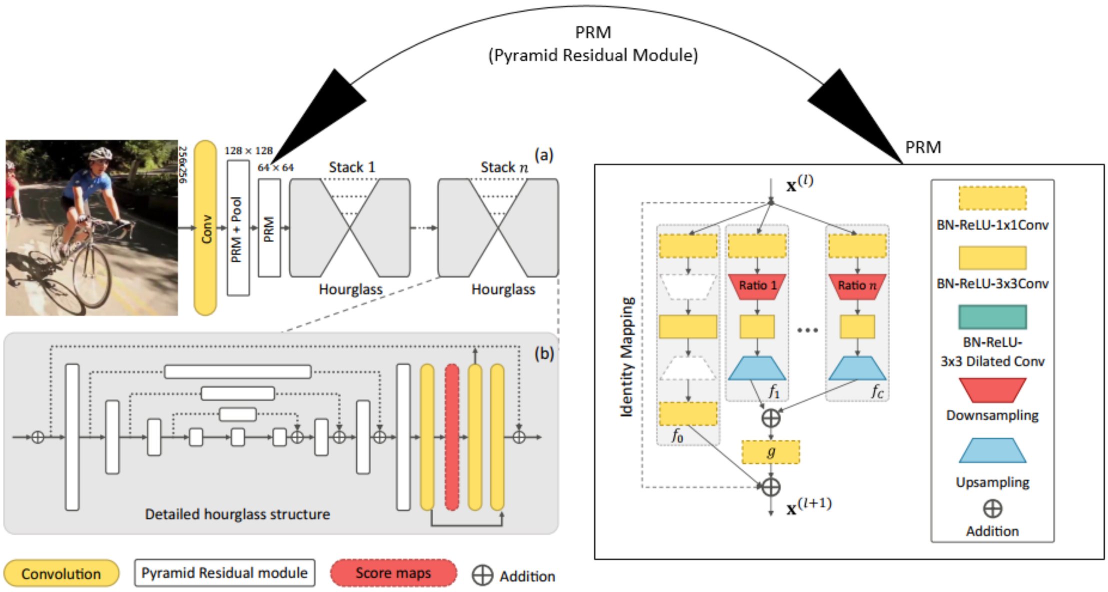


Fig. 2.26 Pyramid Residual Module (adapted from [32]).

Yang et al. [32] introduced a Pyramid Residual Module (PRM), which could be plugged into various CNNs to improve the network's scale-invariance. Figure 2.26 shows how two PRM modules were incorporated into the stacked hourglass network proposed by [20]. Each PRM module contained down-sampling and up-sampling of sub-modules capable of generating feature maps for various levels of pyramids. The integration of two PRM modules in Figure 2.26 enabled the original network to learn feature pyramids from low-level to high-level semantics. It improved its HPE accuracy by approximately 1% on the MPII dataset.

Adopting the idea of Generative Adversial Networks (GANs), Chou et al. [86] built an HPE framework including a generator and a discriminator, each of which shared the same architecture of 4-stack hourglass networks as shown in Figure 2.27. This framework aims to generate human poses that fit the distribution of training data. The generator maps input color images to keypoint heatmaps which show the confidence scores for each keypoint at all locations. On the other hand, the discriminator distinguishes the generated heatmaps from ground-truth ones and produced a different set of heatmaps. The training process continues until the generated heatmaps are indistinguishable from the ground-truth heatmaps.

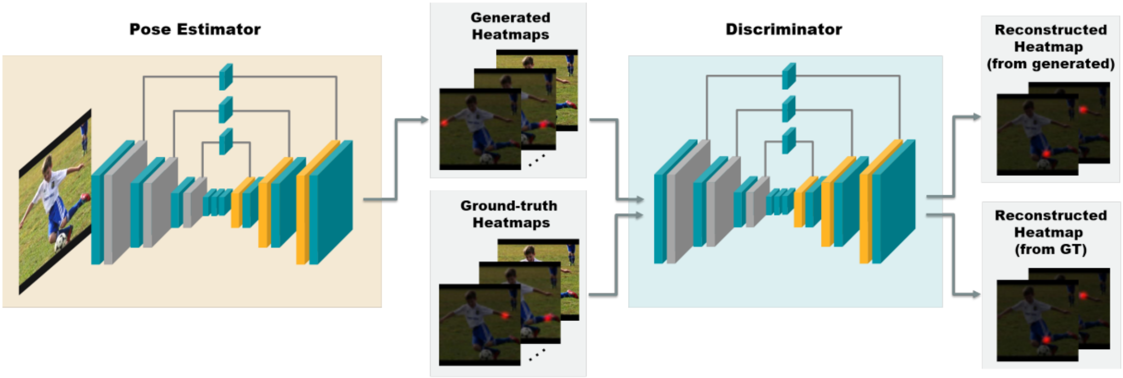

Pose Estimator
Generated Heatmaps
Ground-truth Heatmaps
Discriminator
Reconstructed Heatmap (from generated)
Reconstructed Heatmap (from GT)

Fig. 2.27 Adversarial networks for HPE. (adapted from [86]).

Belagiannis et al. [87] proposed an architecture for HPE by combining a feedforward module with a recurrent module which can be trained end-to-end. As shown in Figure 2.28, this architectue contains fusion layers where the output of Layer 3 and Layer 5 are concatenated and given as input to the recurrent module. The recurrent module can be run iteratively, containing several groups of layers supervised separately to create different numbers of iterations. The combination of fusion layers and the recurrent module was shown to improve the overall HPE performance.

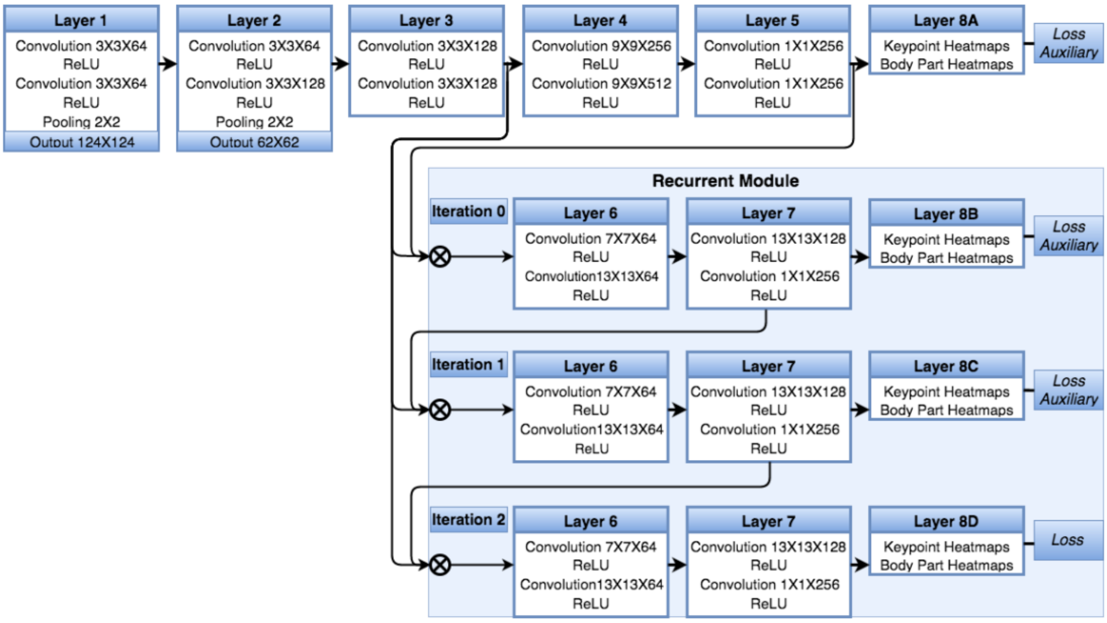


Fig. 2.28 A recurrent network for HPE (adapted from [87]).

Chu et al. [25] used an 8-stack hourglass network that incorporated holistic and part attention maps generated from features at multiple resolutions as shown in Figure 2.29. The holistic attention maps encode the global consistency of the whole body where the part attention maps looks at detailed information of different body parts. The integration of both types of attention maps enables the networks to focus at various scales from local regions to global spaces, resulting in an improved HPE accuracy.

2.3.3 CNN-based approaches in a difference view

Network structures

Network structures used in CNN-based HPE were largely inspired by successive outcomes of the annual ImageNet Large Scale Visual Recognition Challenge (ILSVRC) [88]. The ILSVRC was a competition that evaluated different algorithms and network structures for

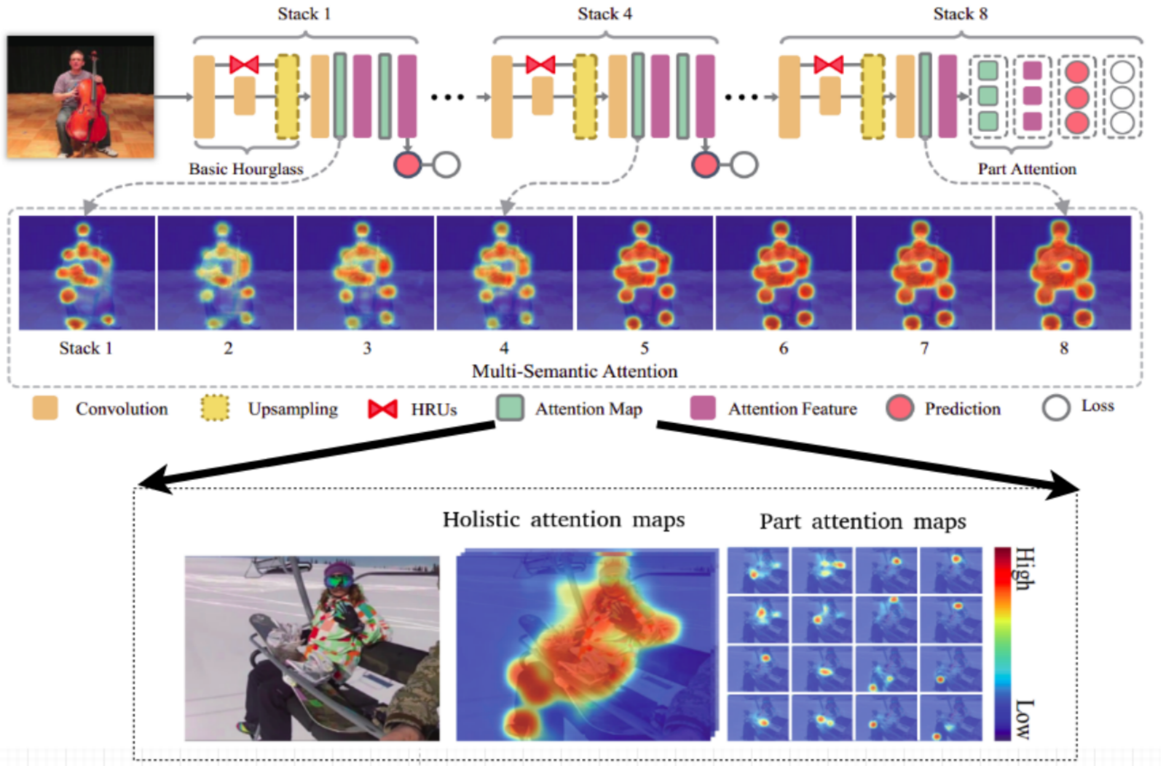


Fig. 2.29 A multi-context attention network for HPE (adapted from [25]).

large scale object detection, localization and classification. Winners of this competition over the years included: Alexnet (2012) [3], ZFnet (2013) [4], Googlenet [89] and VGGnet [90] (2014), Resnet (2015) [91] and Densenet (2017) [92]. Typically, the networks in the later years achieved higher performance or obtained lower error rates. Variants of these networks were adapted to the task of human pose estimation. For examples, variants of the Alexnet were used by [1, 78, 83, 93, 94], while the works by [18–20, 23, 24, 95, 96] deployed and adjusted the architectures of VGGnet and Resnet. The current state-of-the-art in HPE was proposed by [24]. It achieved HPE accuracy of 83.5% on the LSP dataset [28] using a variant of VGGnet, person-centric annotation, and PCK metrics. When using a variant of Resnet, it obtained up to 90.7% accuracy. In comparison, the popular Alexnet-based system for HPE proposed by [2] obtained an accuracy of 75% although the computational and data requirements were much lower than VGGnet-based systems.

Image scales

Training CNNs often requires input images of a fixed size. A network would have better knowledge of an object if it can see the object at different resolutions. Similarly, estimating human poses requires the understanding of the whole body structure, as well as the arrangement of body parts and their orientations. This information would be best captured if different scales of an image are available [20]. Inspired by this observation, a number of CNN architectures were designed to capture input objects at variable scales [20, 32, 81]. Tompson et al. [81] improved the scale-invariance properties of HPE by training CNNs with two different input image resolutions in a single framework. On the other hand, Newell et al. [20] addressed this problem in a different way by training the network using input images of a fixed size, but combining features of different levels through repeated downsampling, upsampling, and feature concatenation. Further extension of this idea was the aforementioned Pyramid Residual Module (PRM) introduced by Yang et al. [32]. This approach had a general character and could be incorporated into many network structures to improve their scale-invariance.

Successive predictions

Successive predictions have become popular in recent years. They help refining the estimation and improve localization performance in the high-precision range [18, 20, 24, 83, 85–87]. Toshev and Szegedy [83] introduced a cascade of DNN regressors, in which the keypoint predictions of the previous regressor were applied so that the following regressor could learn any displacements of joint locations with regard to the ground-truth locations in a repeated manner. By using a single network, Carreira, et al. [85] proposed the Iterative Error Feedback, where output predictions of the network were fed back and concatenated with its input through a number of iterations. Successive predictions were also deployed by Newell, et al. [20] in a stack of networks. Intermediate predictions were applied to each sub-network. Experiments were conducted to compare the accuracy of 2-, 4-, and 8-stack networks on the MPII dataset using PCK metrics. The HPE accuracy obtained for each case was 87.4%, 87.8%, and 88.1% respectively. In a different design, Bulat and Tzimiropoulos [24] combined a detection and a regression network for the HPE. The detection network predicted the locations of body parts and generated heatmaps showing probabilistic distributions. The heatmaps combined with input images were regressed by the other network for further joint prediction.

Intermediate supervision

In supervised CNN training, a loss function is applied to measure the level of difference between predictions and ground-truth. When different networks are stacked together, it causes the unified network to become deeper and more prone to the vanishing gradient. By observing histograms of gradient magnitude across training epochs at various depths in an architecture, Wei, et al. [18] discovered that intermediate supervision helped to reduce the effect of the vanishing gradient. Intermediate supervision was often used in combination with stacked networks and was applied for each sub-network. Newell, et al. [20] also discovered that intermediate supervision contributed to improvement of the HPE accuracy.

CNN-based message passing

Message passing is used by traditional PS and part-based approaches to learn the spatial relationships between body parts. Tompson et al. [81] formulated his spatial model as Markov Random Field using CNNs. The model was further enhanced by a mixture of deformation models introduced by Yang et al. [22] and by structured feature learning introduced by Chu et al. [2]. Yang et al. [22] constructed the message passing in a loopy graph (non-tree-based model), while it was built as a tree-based model by Chu et al. [2].

Gaussian heatmap label

The CNN-based regressors as proposed by Toshev and Szegedy [83] mapped input image pixels (e.g. 224x224x3) to body joint coordinates (e.g. 26x2). However, Jain, et al. [78], reasoned that the direct mapping worked very poorly, since the valid poses contributed just a small portion in the output space and came with a high number of invalid poses. Therefore, binary heatmap labels were proposed by [2, 78]. Using binary heatmap labels, input images would be mapped to heatmaps of joint positions created based on ground-truth keypoints. One heatmap was generated for each joint position. It was given as a binary square matrix with the value of 1 denoting pixels in the corresponding input image that contained a given joint and value of 0 denoting pixels where the joint was absent. Another type of heatmap label was introduced by [18–20, 23, 24, 97]. In this case, the heatmap label was given as a Gaussian probability density distribution of a joint position.

2.4 Research questions and conclusion

As shown in the above literature review, the current state-of-the-art technology in HPE is lead by the CNN-based approaches. Existing research gaps led to the following six research questions to be investigated in this thesis:

Research Question 1 How to efficiently apply the CNN modeling to maximize accuracy of the body part recognition for HPE?

Research Question 2 What is the effect of different tree-based human pose models on the CNN-based HPE? How do the conventional anatomy-based tree-based models compare with the data-driven tree-based models?

Research Question 3 How does the number of tree nodes used in modelling of human pose affect the CNN-based HPE accuracy?

Research Question 4 How do the tree-based models compare with the non-tree-based models in terms of HPE accuracy?

Research Question 5 What is the effect of different connections between body parts of the non-tree-based models on HPE accuracy?

Research Question 6 How to design an efficient hybrid structure combining both non-tree and tree-based models of the human pose?

Following **Section 2.3.2**, potential improvements of the CNN-based HPE can be achieved through the study of different more efficient network configurations. This observation lead to **Research Question 1**. As described in **Section 2.3.3**, an investigation of data-driven tree-based models as opposed to anatomy-based models could be beneficial. The effects of using different numbers and configurations of nodes, application of different inter-node connections and generation of complex CNN structures could also be investigated, and were proposed in **Research Questions 2, 3, 4, and 5**. The idea of successive predictions in **Section 2.3.2.** inspired **Research Question 6**.

Chapter 3

CNN optimization of the HPE based on a graphical model

3.1 Preview

By combining the representation flexibility of graphical models with the data-driven power of CNNs, Chen and Yuille [1] proposed an HPE system with significantly improved estimation accuracy. However, the CNN structure (ChenNet) suggested by Chen and Yuille [1] has not yet been explored; the classification accuracy of body parts of this network was approximately 41% when evaluated on the Leeds Sports Pose (LSP) dataset [8]. This chapter aims to research answers to the Research Question 1. Namely, methods of improving the ChenNet design are investigated by exploring different network configurations and applying transfer learning for the original ChenNet on the LSP data set. The modified ChenNet is referred to as the MChenNet in the remainder of the chapter.

3.2 Related work

Recent interest in CNNs for automatic object recognition was spurred on by the availability of the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) dataset and the CNN structure proposed by Krizhevsky, et al. [3], known as the AlexNet network. Various algorithms and architectures have been developed based on this network [4, 5, 90, 98, 99]. One typical architecture is the Znet network by Zeiler and Fergus [4], which used a 7x7 receptive field array and a stride of 2 in the first convolutional layer in contrast to the 11x11 receptive field array and a stride of 4 applied by the AlexNet. The choice of the receptive field and the stride size in the Znet network was determined through an innovative visualization technique. The proposed Znet structure outperformed the AlexNet on the ILSVRC classification task. The effects of the depth of convolutional layers and data augmentation schemes were investigated by Chatfield, et al. [5], who evaluated different CNN architectures using the same training and inference protocols. The experimental results indicated that the size of the network could be significantly reduced with only a minor performance degradation.

Another important improvement to the CNN based classification was achieved through the implementation of multi-scaling and sliding window, as proposed by Sermanet, et al. [99]. Through the sliding window, the CNN explored densely all input image regions at multiple scales while the AlexNet and the Znet network architectures applied a fixed structure of 8 layers (5 convolutional layers and 3 fully connected layers) at a single scale.

Deeper CNN architectures typically lead to a higher classification accuracy. This has been shown when using networks such as the VGGnet proposed by Simonyan and Zisserman [90] (with 16-19 layers) and the GoogLeNet proposed by Szegedy, et al. [89] (with 22 layers).

Transfer learning is another efficient approach used for training CNNs to improve their performance. Girshick, et al. [89] pre-trained a CNN on the large ImageNet dataset for an image object classification task to differentiate between 1000 object categories. The pre-trained network was then trained on a smaller, very scarce dataset to detect objects within a bounding box. That resulted in a higher classification accuracy compared to models withough pre-training. Experiments conducted by Agrawal, et al. [100] showed that even when the training data that used pre-trained weights was not in abundance, the transfer learning could still lead to an increased performance.

This chapter explores network configurations with different pooling schemes and varied depth of layers. The application of transfer learning from a model on the FLIC-full [30] dataset to the model on FLIC [8] dataset is investigated.

Table 3.1 Variants of AlexNet (adapted from [3],[4],[5])

. The convolution layer is denoted as "number of channels (receptive field size x stride)". The pooling is specified as "kernel size x stride".

layers	AlexNet [3]	ZNet [4]	CNN-M [5]	CNN-M-1024 [5]
input	224x224x3	224x224x3	224x224x3	224x224x3
conv1	96 (11x3)	96 (7x2)	96 (7x2)	96 (7x2)
pool1	3-2	3-2	2-2	2-2
conv2	256 (5x1)	256 (5x2)	256 (5x1)	256 (5x1)
pool2	3-2	3-2	2-2	2-2
conv3	384 (3x1)	384 (3x1)	512 (3x1)	512 (3x1)
conv4	384 (3x1)	384 (3x1)	512 (3x1)	512 (3x1)
conv5	256 (3x1)	256 (3x1)	512 (3x1)	512 (3x1)
pool3	3-2	3-2	2-2	2-2
fc6	4096	4096	4096	4096
fc7	4096	4096	4096	1024
fc8	1000	1000	1000	1000

Table 3.1 illustrates network variants of the AlexNet [3]. These structures showed that receptive field size, stride, and number of channels had an effect on a network's performance. For example, by using the receptive size and stride of 7x2 at the first convolutional layer (conv1), the ZNet performed better in comparison to the AlexNet. In the other variants, by applying a suitable number of channels in convolutions and fully-connected layers, the CNN-M and CNN-M-1024 [5] outperformed even the ZNet [4].

3.3 The original ChenNet

3.3.1 The Model

The system uses a graph G = (V, E) to model human poses where, V denotes vertices or positions of body joints, and the edges $E \subseteq V \times V$ specify the spatial relationships between the joints. Given an input image I, the full score $F(|)$ of a pose configuration, is given as

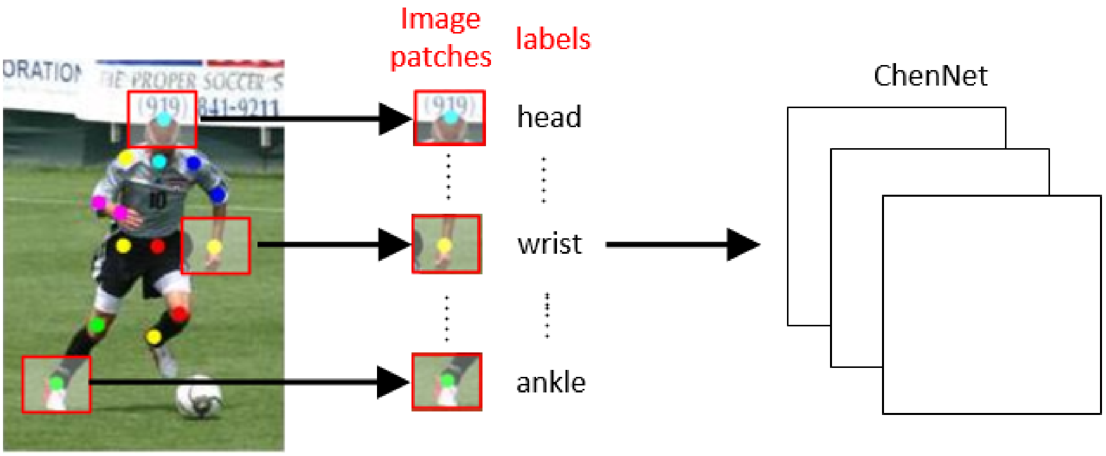


Fig. 3.1 The ChenNet input.

follows:

$$F(l,t|I;\theta,\omega) = \sum_{i \in V} \phi(l_i|I,\theta_i) + \sum_{i,j \in E} \psi(l_i,l_j,t_{ij},t_{ji}|I,\omega_{ij}) \tag{3.1}$$

where θ_i and ω_{ij} are model parameters; $k = |V|$ specifies the number of parts (nodes); $i \in 1, \ldots.K$ denotes the ith part; $l = \{l_i\}_{i=1}^{K}$ represents the pixel locations of parts; for each edge in the graph (i,j) $\in E$, t_{ij} denote the body-part types of spatial relationships.

In the formula given by Equation 3.1, the pose configuration probability $F(|)$ contains the part appearance term (or the unary term) $\phi(l_i|I,\theta)$ and the spatial relational term $\psi(l_i,l_j,t_{ij},t_{ji}|I,\omega)$. While the appearance term provides local confidence of the appearance of a part i located at l_i, the relational term, on the other hand, models the spatial relationship of two neighboring parts i and j.

3.3.2 The ChenNet structure

The ChenNet [33] was formulated to classify different body parts. To train this network, image patches containing body parts and corresponding labels were provided as demonstrated in Figure 3.1. To obtain heatmaps of each body part, after the training all fully connected layers were converted to fully convolutional ones by reshaping the weight matrices of the fully connected layers, resulting in a network consisting only of convolutional layers. This

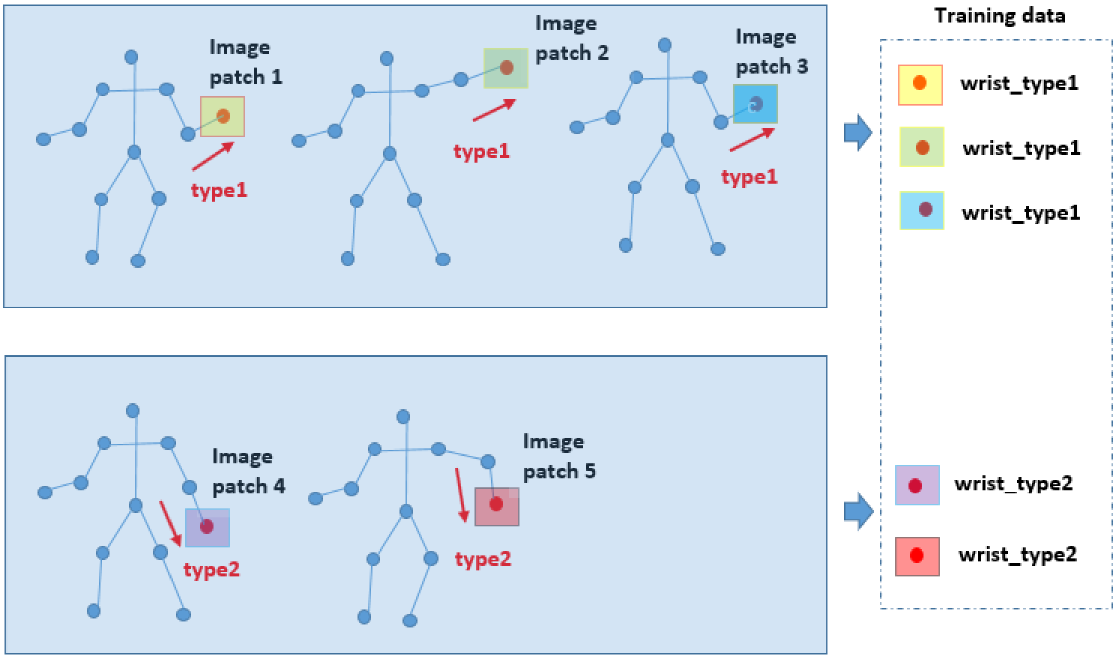


Fig. 3.2 The ChenNet body-part types.

fully convolutional network, containing the appearance model parameter (θ), would then be used to extract image features to obtain the appearance features for each body part.

To improve the feature representation, body-part types were taken into account. The types of body part were determined based on their relative orientations with respect to the neighboring parts. Hence, each body-part type was represented as a set of spatial relations with reference to the parent (or children) parts organized on a graphical tree structure. Taking the wrists on Figure 3.2 as an example, it can be seen that the Wrist_type1 denoted a group of wrist patches at the north-east of corresponding elbows. Similarly, the wrist_type2 features denoted a group of wrist patches positioned at the south-east with regard to the corresponding elbows.

Figure 3.3 shows the ChenNet configuration as originally proposed in [1]. The network input was given as a RGB image of size 36x36x3 pixels. Each input image was pre-processed and passed through a structure of five convolutional layers (layer 1 to layer 5) and three fully connected (FC) layers (layer 6 to layer 8). The first convolutional layer used a receptive field of size 5x5 pixels with stride of 2 pixels, whilst the remaining layers used a stride of 1 pixel

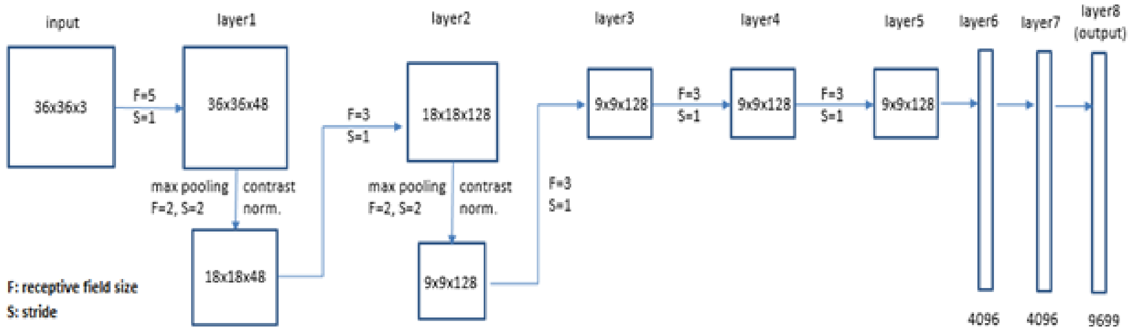


Fig. 3.3 The ChenNet configuration (adapted from [1]).

with a 3x3 pixels receptive field. Overlapping maximum pooling procedure was applied after the first and second convolutional layer with a window size of 3x3 pixels and a stride of 2 pixels.

Similar to the AlexNet architecture [3], every hidden layer (layer 1 to layer 7) of the ChenNet was followed by a Rectified Linear Unit (RELU) layer. The Local Response Normalization (LRN) was also applied in layer 1 and layer 2. As observed by Simonyan and Zisserman [90], the LRN did not significantly contribute to the network performance but consumed more memory and required longer training times.

3.4 Proposed modifications to the ChenNet configurations

Applications of popular CNN structures have shown that different pooling schemes (e.g. overlapping and non-overlapping pooling), weight initialization and layer depth can significantly affect the CNN performance. This chapter investigates and compares a series of different CNN configurations using a common HPE framework. These configurations, are listed in Table 3.2. The modified versions of the ChenNet are referred to as MChenNet. The differences between the MChenNet and the the original ChenNet network can be summarized as follows:

- Configuration (B) uses non-overlapping pooling.

- Configuration (C) uses initial weights from a pre-trained network.

- Configuration (D1), (D2), (D3), (D4) use varied layer depth resulting in different network sizes determined by the number of parameters.
- Configuration (B_D2_D6) combines Configuration (B), (D2) and (D6).

Table 3.2 Different network configurations.

CNN configurations	**Description**	**Number of parameters (millions)**
A (the ChenNet [33])	The number of channels in 8 layers are: 48, 128, 128, 128, 128, 4096, 4096 and 9699 respectively. Pooling scheme: overlapping with windows size z = 3, stride s = 2.	99.5
B	Pooling scheme: non-overlapping with window size z = 2, stride s = 2.	99.5
C	Same architecture as the A configuration. Use pre-trained weights from FLIC-full dataset.	99.5
D1	The number of channels in 8 layers are: 30, 48, 48, 48, 48, 48, 4096, 4096 and 9699 respectively.	72.5
D2	The number of channels in 8 layers are: 48, 96, 96, 96, 96, 4096, 4096 and 9699 respectively.	88.6
D3	The number of channels in 8 layers are: 48, 128, 256, 256, 256, 4096, 4096 and 9699 respectively.	1430
D4	The number of channels in 8 layers are: 48, 128, 128, 128, 128, 4096, 1024 and 9699 respectively.	57
B_D2_D6	Pooling scheme: non-overlapping with window size z = 2, stride s = 2. The number of channels in 8 layers: are 48, 96, 96, 96, 96, 4096, 1024 and 9699 respectively.	46

3.4.1 Configuration with different pooling schemes

Pooling is a process used by the convolutional layers of the CNN to down-sample the input images, and to turn them into input features for the fully connected layers. There are two popular pooling scheme: the overlapping pooling scheme that uses a window size of 3 pixels and a stride of 2 pixels, and the non-overlapping pooling scheme that uses a window size of 2 pixels and a stride of 2 pixels. The ChenNet [33] applied the overlapping pooling. The fully

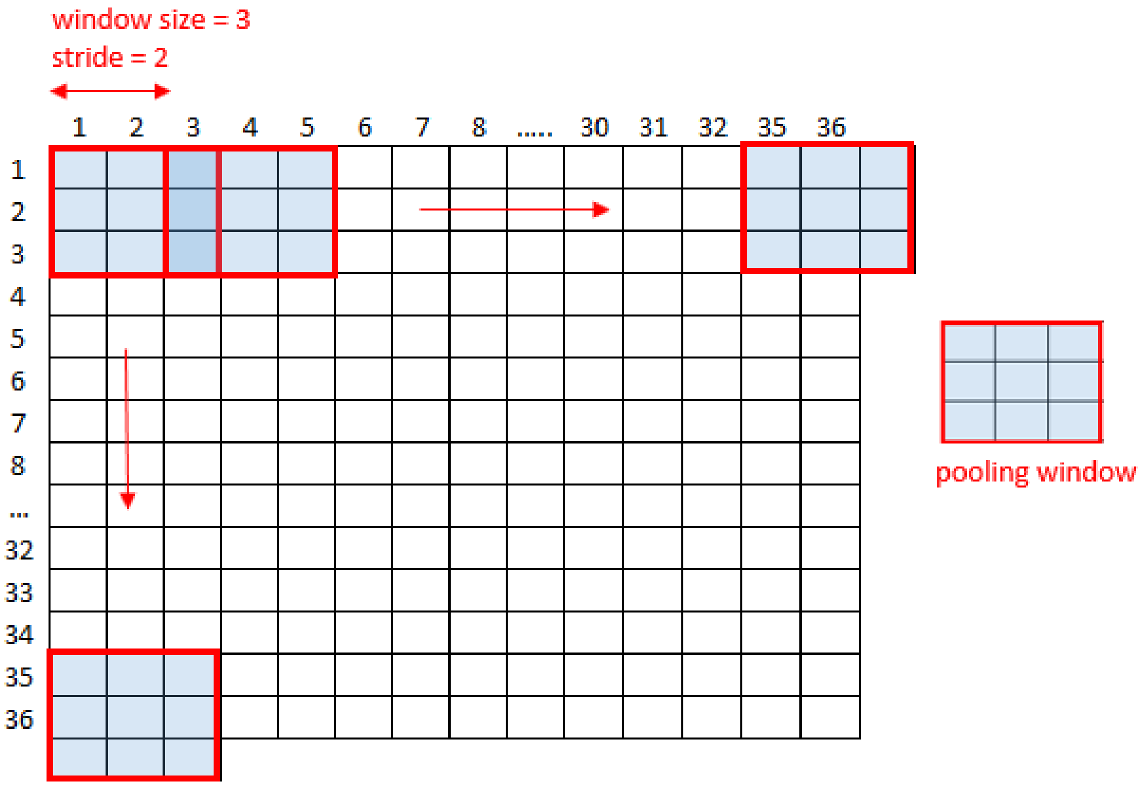


Fig. 3.4 The overlapping pooling of the ChenNet [1].

connected part of the network was trained on RGB input images of size 36x36x3 downsized by the convolutional layers to the image feature arrays of size of 9x9x128. Convolving along the 36x36 pixels image array, the 3x3 pixels window with the 2 pixels stride did not fit integer-multiple times into the image area creating a boundary estimation problem (Figure 3.5). On the other hand, the proposed in this chapter non-overlapping pooling method (Figure 3.5) fitted perfectly when operating along the feature dimension eliminating the boundary problem.

3.4.2 Configuration with variation in receptive field and stride

Popular CNN structures are often trained with large images of size 224x224x3 pixels. Therefore, large receptive field (F) and stride (S) are utilized in the first convolutional layer. For example, the Alexnet [3] used F = 11 and S = 4, while the Znet [4] used F = 7 and S = 2. The latter configuration resulted in a higher classification accuracy but required longer

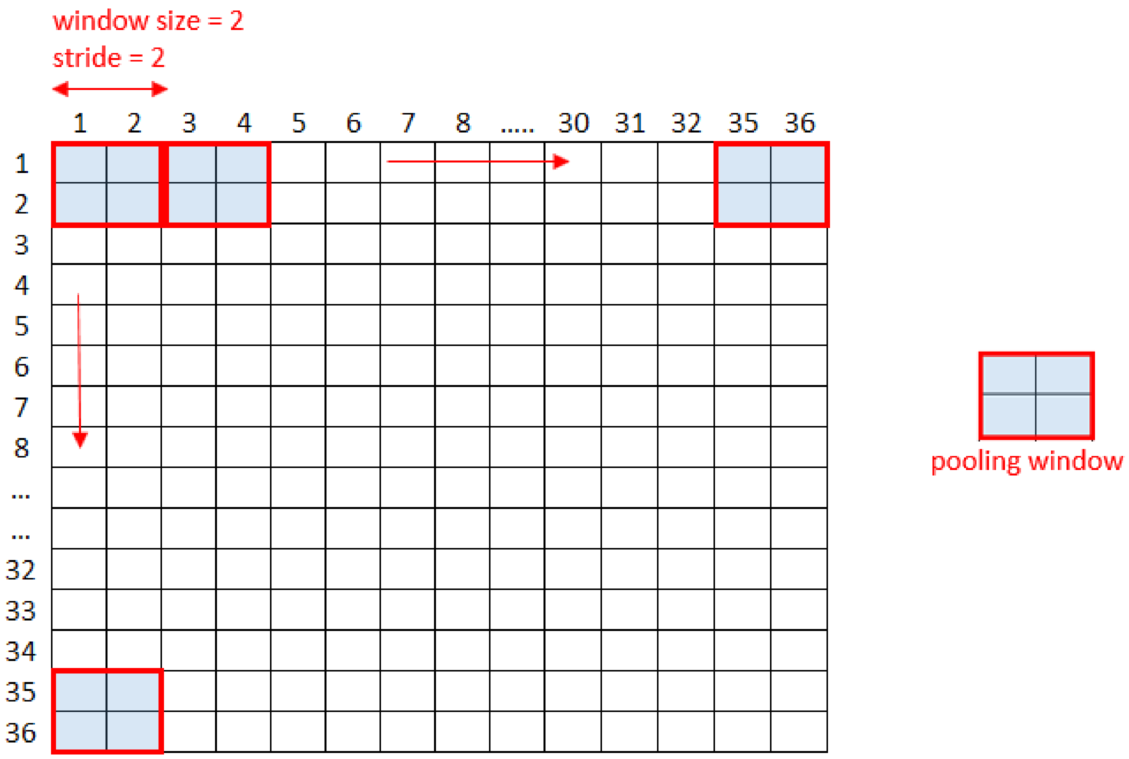


Fig. 3.5 The proposed non-overlapping pooling for MChenNet.

training time. In contrast, the ChenNet used a smaller input sample size of 36x36x3 pixels and thus used smaller receptive field (F = 5) and stride (S = 1) for the first convolutional layers. If large receptive field sizes and strides (F=11, S=4 or F=7, S=2) were to be applied to the ChenNet network, the spatial information would have been reduced significantly. Therefore, this chapter maintains the use of F=5 and S = 1, similar to those of the original ChenNet.

3.4.3 Transfer learning for the MChenNet

There are various pre-trained networks available such as the AlexNet [3] or the VGG net [90]. However, these networks were trained on large input images (224x224x3) and thus their , weight structures are much larger than the weight structures required by the ChenNet. Due to this incompatibility, the AlexNet or the VGG weights could not be transferred to the ChenNet [33] to train on the LSP dataset [8].

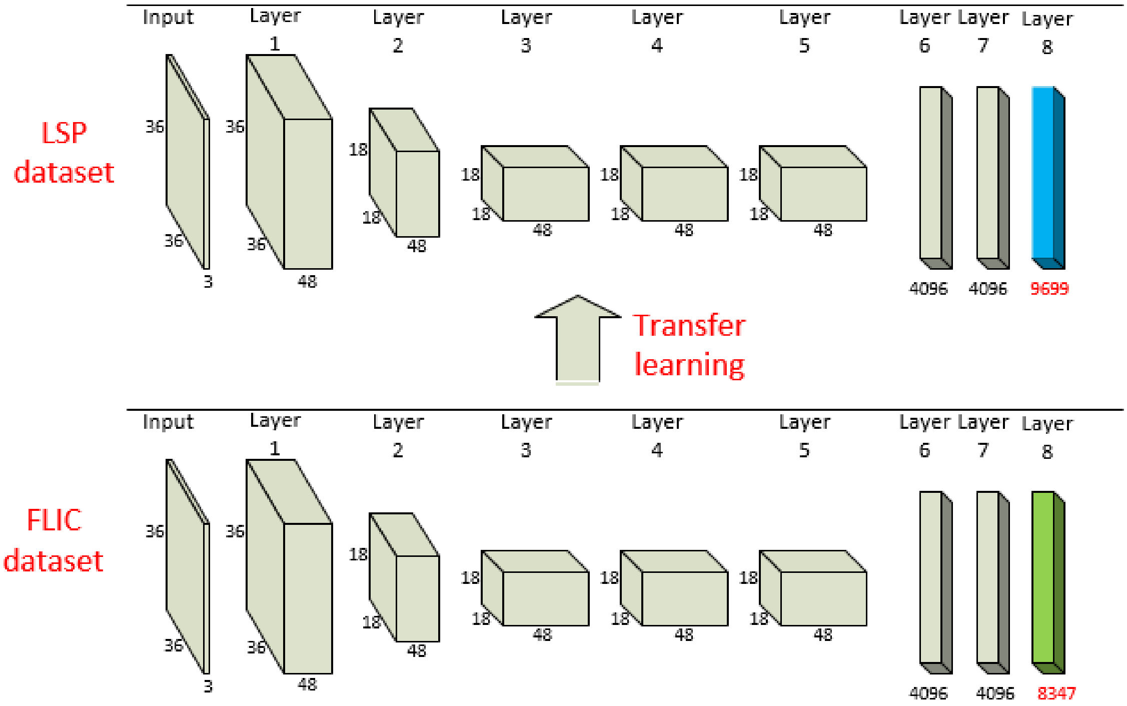


Fig. 3.6 Transfer learning for the MChenNet from the upper-body annotation in FLIC dataset [30] to the full-body annnotation in LSP dataset [8].

To be able to apply the transfer learning, a CNN was pre-trained on the FLIC-full dataset [30] containing 20000 annotated images of upper-body poses. The pre-trained CNN had the same architecture as the ChenNet, except that the last fully connected layer's (layer 8) dimension was set to 8347 nodes corresponding to the number of FLIC body part templates as proposed by [1] (Figure 3.6).

To meet the large training data requirements of the CNN, data augmentation was applied. The FLIC-full original 20000 training images of 18 annotated body parts (N = 18) were augmented through 22 different random rotations and horizontal flipping. This procedure increased the number of training images from 20000 to approximately 15 million. The hyper-parameters for the pre-training were similar to those used by Chen and Yuille [1]. Namely, the linear weight decay factor was equal to 0.0005; the momentum was 0.9; and the initial learning rate was initially setup to 0.001 and gradually decreased by the factor of 10 after 20000 iterations. The training batch size was 512 and the layer initialization was sampled from the Gaussian distribution with zero mean and variance of 0.01. Weights of the pre-trained network were then used to train the modified ChenNet (MChenNet) using the LSP dataset. The fine-tuning structure was the same as the structure used in the pre-training, except for the last fully connected layer (layer 8) dimension being replaced by a value of 9699. In the testing phase, a window is scanned through an input image, generating a series of small input images. These images (extracted from the original large images) generate 9699x2D heatmaps. Each heatmap represents a part type as shown in Figure 3.2. These heatmaps are post-processed to obtain refined heatmaps of size 12x2D, equivalent to 12 heatmaps of 12 joints. This value of 9699 is the output dimension of the ChenNet corresponding to the number of LSP body part templates as proposed by [1]. The learning rate schedule for the hidden layers / last layer $10^3/10^2; 10^4/10^3; 10^5/10^4$ is with reference from [90].

3.5 Experimental Results of the HPE

3.5.1 Evaluation of the proposed MChenNet configurations

The evaluation procedure applied the 4-fold leave-one-out cross validation method described by Refaeilzadeh, et al. [101] to evaluate the proposed MChenNet configurations (see Table 3.2) using the Caffe framework [102]. The evaluation was performed on the LSP training set as illustrated in 3.7. The training set was divided into four equal parts of 250 images each. Each CNN configuration was trained and evaluated using four leave-one-out folds, where each fold used a model trained on three of the four parts and tested on the remaining fourth part used only for testing.

During the evaluation procedure, each of the MChenNet configurations given in Table 3.2 was trained and tested four times using the four scenarios; the results were averaged across all four evaluations. The averaged results represented the network accuracy of each configuration.

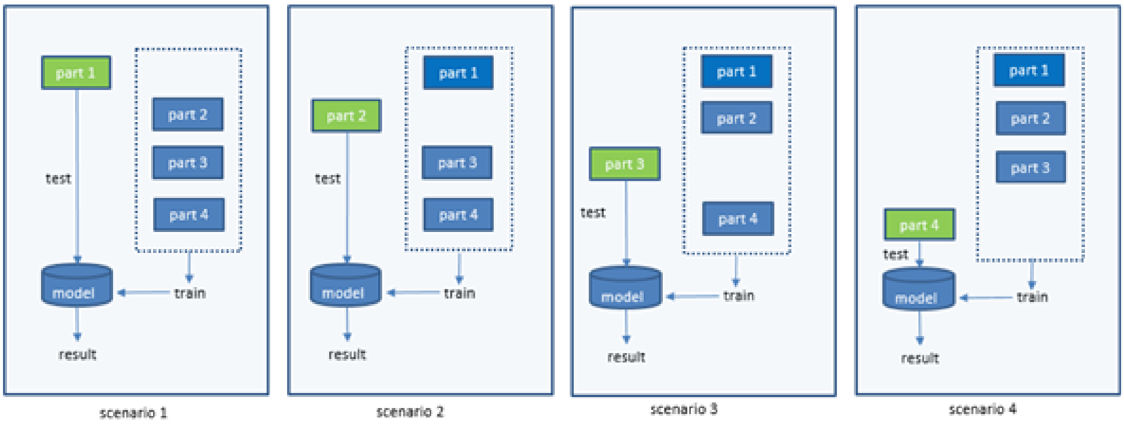


Fig. 3.7 Four-fold cross validation (adapted from [101]).

3.5.2 Experimental results

Table 3.3 shows the average network accuracy obtained for each of the proposed MChenNet configurations after 50000 training iterations. After 50000 iterations, the accuracy had plateaued; therefore, the HPE outcomes obtained after 50000 iterations were used to compare the MChenNet performance for different configurations.

Each of the MchenNet configurations was applied to perform the HPE as proposed in [1], and the resulting average accuracies are shown in Table 3.4.

Table 3.3 The average accuracy of the HPE after different numbers of iterations.

Configuration	**The accuracy using strict PCP after a number of training iterations (%)**								
	10k	**15k**	**20k**	**25k**	**30k**	**35k**	**40k**	**45k**	**50k**
A	36.42	38.29	38.35	41.01	40.99	41.02	40.93	41.16	41.13
B	37.58	39.44	39.9	42.51	42.38	42.45	42.32	42.58	42.58
C	39.42	40.01	39.68	43.3	43.29	43.18	43.08	43.54	43.5
D1	31.87	35.05	35.99	39.18	39.81	39.41	39.39	39.68	39.68
D2	35.79	37.81	38.4	41.19	41.19	41.24	41.19	41.41	41.4
D3	37.67	38.72	38.52	41.08	40.97	40.92	41.07	41.09	41.09
D4	35.64	37.72	37.98	40.91	40.89	40.87	40.79	41.04	41.04
B_D2_D4	35.58	38.33	39.46	42.07	42.1	42.11	42.03	42.3	42.3

Table 3.4 The HPE accuracy (using strict PCP (Section 2.2.4) evaluation protocol) on each body part for different MChenNet configurations.

Configuration	**The accuracy using strict PCP on each body part (%)**						
	Head	**Torso**	**Upper arm**	**Lower arms**	**Upper legs**	**Left legs**	**Mean**
A (the ChenNet [33])	85	93.1	70	58.1	83.7	77	77.81
B	85.5	93.4	71.4	58.5	84.4	77.8	78.5
C	86.4	94.3	72.8	60.7	85.2	79	79.7
B_D2_D4	84.3	92.6	71.4	58.2	84.2	78	78.11

3.5.3 Discussion

Pooling scheme

Krizhevsky, et al. [3] reported that the overlapping pooling scheme performed better than the non-overlapping one. However, in a number of applications, the CNNs have been reported to perform well with the non-overlapping scheme [5, 90]. In this chapter, the non-overlapping pooling scheme of the MChenNet (configuration (B) in Table 3.2) was compared with the overlapping scheme (configuration (A)) used by the original ChenNet. The results showed, that configuration (B) outperformed configuration (A) by a margin of 1.4% (42.58% vs 41.13%, of the average classification accuracy (as shown in Table 3.3). Similarly, when looking at the performance across different body parts shown in Table 3.4, it can be observed that the configuration B is again outperforming configuration A. There is an improvement of the mean accuracy of 0.69% (78.5% for configuration B vs 77.81% for configuration A). Given that the pooling scheme was the only difference between configurations A and B, it can be concluded that the non-overlapping pooling scheme leads to an improvement of the HPE accuracy. This outcome appears to be consistent with the results previously reported by Krizhevsky, et al. [3] when the non-overlapping pooling scheme was applied to the image object classification problem. The finding of non-overlapping pooling which outperformed overlapping pooling is specific to HPE only.

Transfer learning

The transfer learning approach used in configuration (C) achieved the highest overall performance. Configuration (C) surpassed configuration (A) by a margin of 2.36% (43.5% vs 41.13%, as shown in Table 3.3) of the average HPE accuracy. Similarily, when looking at the performance across different body parts in Table 3.4, it can be observed that the configuration C is again outperforming configuration A. A mean accuracy gain of 2% from 77.81% to 77.9%, as shown in Table 3.4). In particular, difficult-to-detect body parts, such as the upper and lower arms, obtained greater accuracy improvement compared to the head or torso, which are easier to detect. Table 3.4 shows that the improvements for the upper arms and lower arms were 2.8% and 2.6% respectively. Given that the only difference between configuration A and configuration C was the application of the transfer learning, it can be concluded that this type of learning leads to an improvement of the HPE.

Layer depth

The depth (or the number of channels) of the convolution layers (e.g., the layer 1,2,3,4 and 5, as shown in Figure 3.3) and the fully connected layers (e.g., the layer 6, 7 and 8 as shown in Figure 3.3) affected the network size and hence the network performance. Configurations (D1), (D2), (D3) were different from one another with regard to the depth of convolution layers. Then, configuration (D4) used a smaller depth of fully-connected layers. Given a fixed amount of training data, reducing layer depth can lead to under-fitting, while increasing the layer depth can cause over-fitting. Table 3.3 shows that an improved accuracy is exhibited (41.4% vs 41.13%) when using configuration (D2). In contrast, Configuration (D1), which had a smaller layer depth, and Configuration (D3), which had a larger layer depth, both decreased accuracy (Table 3.4). The reduced accuracy indicated that the (D1) network was under fitting while the (D3) network was over fitting. As a result, experiments with configurations having smaller layer depth than (D1) or larger layer depth than (D3) were not considered. Although the accuracy of (D4) was slightly lower than (A) (41.04% vs 41.13%, as seen in Table 3.3), Configuration (D4) required considerably fewer parameters (57 million vs 99.5 million, as shown in Table 3.2).

Combined network configuration

By observing that Configuration (B) obtained good HPE accuracy but required high number of network parameters, the Configuration (B_D2_D4) was proposed, combining Configuration (B), (D2) and (D4), resulting in a very compact network of 46 million parameters (shown in Table 3.2). This configuration obtained an estimation accuracy of 42.3%, as seen in Table 3.3), which was a 1% improvement compared to the original Chenet, configuration (A).

3.6 Conclusion

This chapter investigated various CNN configurations to improve the accuracy of the previously proposed ChenNet approach that trained a body part classifier using CNNs. Experimental results demonstrated that the network's pooling scheme, transfer learning as well as the depth of the output layers have an effect on the HPE results. In particular, by training the ChenNet with pre-trained weights from a large dataset, the CNN accuracy was improved by 2%. Future work will explore ChenNet by integrating very deep structures such as the VGGnet, GoogLeNet, ResNet and DenseNet.

Chapter 4

Tree-based models

4.1 Preview

The focus of the previous chapter was to investigate the effects of different CNN classifier configurations on the HPE results disregarding the human pose model. The same basic tree-based model representing body parts and the connections between them was used in all experiments. In this chapter the focus moves towards the human pose models. In particular different tree-based modelling approaches are tested and compared within the same HPE benchmark setup. The chapter aims to provide answers to research questions 2 and 3 by investigating what is the effect of different tree-based human pose models on the CNN-based HPE, how the conventional anatomy-based tree-based models compare with the data-driven tree-based models, and how the number of tree nodes used in modelling of human pose affects the CNN-based HPE accuracy.

Tree-based structures are commonly used to model relationships between body parts for articulated HPE. Tree-based structures can be applied to model relationships between feature maps of joints in a structured learning framework using CNN. This chapter proposes new data-driven tree-based models for HPE. In data-driven tree models, the connections between tree nodes are obtained based on the distribution of joints using the ground-truth joint labels of the LSP dataset. On the other hand, the connections between tree nodes in the anatomy-based model is formed as referred to the anatomy of the human body. The data-driven tree-based structures were obtained using the CLRG algorithm representing the joint distribution of human body joints and tested using the LSP dataset. The chapter also analyzes the effect of the variation of the number of nodes on the accuracy of the HPE. Experimental results showed that the data-driven tree-based model obtains 1% higher HPE accuracy compared to the traditional anatomy-based model. A further improvement of 0.5% was obtained by optimizing the number of nodes in the traditional anatomy-based model.

4.2 Introduction

Most systems that model human poses use the part-based approach, which represents the human body as a collection of rigid parts constrained in different ways. One such constraint is the kinematic constraint among neighboring body parts arranged in a tree-based structure. A tree-based structure consists of nodes and edges, where nodes correspond to body joints (or parts) and edges represent the pairwise relationship between parts. Tree-based structures can model both basic rigid parts, e.g. arms, legs, torso or head, and combined parts that cover large areas of the body and that contain more than one rigid part, e.g. torso and arms [71, 77].

A tree-based structure based on CNN to model human poses as first proposed by Chu, et al [2]. This tree-based structure was an anatomy-based tree with 26 nodes corresponding to 26 human joints on the LSP dataset [8]. Structural dependencies between feature maps of body joints in this framework were learned using CNN. However, there is no evidence to substantiate that this particular tree-based structure and the 26-node tree are optimal.

The goal of this chapter is to find an optimal tree-based representation to model human poses for the LSP dataset by applying the CLRG-based data-driven tree-based model [103] and exploring the optimal number of nodes in a tree-based model. To achieve this, this research proposed a new data-driven tree-based model for an existing structured learning framework to be tested against traditional anatomy-based approaches. The proposed structure was optimized with respect to the number of nodes to provide further improvement in HPE accuracy.

4.3 Related works

Tree-based models of human poses were first proposed by Felzenszwalb and Huttenlocher [68] and used in part-based approaches to model pair-wise relationships between adjacent parts. To capture a greater range of pose variations, a global mixture of trees [69] or a mixture of local parts for each tree node [33] was introduced. One disadvantage of the tree-based representation is its inability to model complex pose space, as only the pairwise interactions between nearby parts are captured. To solve this problem, Wang, et al. [70] proposed a non-tree-based structure (or a loopy graph) to model high-order relationships between body parts. However, loopy graphs use approximate inference, which sacrifice the exact inference benefits of tree-based structures. This limitation can be overcome by the hierarchical tree-based structure with latent nodes introduced by Tian et al. [71].

The majority of the current tree-based structures used for human pose estimation are based on the anatomy of the human body [33, 71, 78]. Tree-based structures can be learned from observable variables to find tree approximations for joint distributions of body parts [77]. Choi, et al. [103] introduced two algorithms to automatically build latent tree-based structures from observations: the recursive grouping and the CLRG algorithm. Using the CLRG algorithm, Wang and Li [77] learned a tree-based model from the pose space of the LSP dataset, where body joint positions play the role of observable variables. Different tree-based configurations are proposed as shown in Table 4.1. These resultant configurations were tested on the structured learning framework introduced in [2].

4.4 Tree-based models

4.4.1 Obtaining a tree-based model

A tree-based structure consists of nodes and edges. In the context of HPE, nodes correspond to body joints (or parts) and edges represent the pairwise relationship between parts. Given the LSP dataset [8] with 14 joints annotation for each pose or each image, a 14-node tree can be established from the annotation data to represent human poses. To improve the representation power and increase training data, most works use the 26-node tree for this dataset. Therefore, additional joints are added to the original 14-joint annotation to model 26-node trees,as illustrated in Fig. 4.1). Fig. 4.2 shows how added joints are formed by

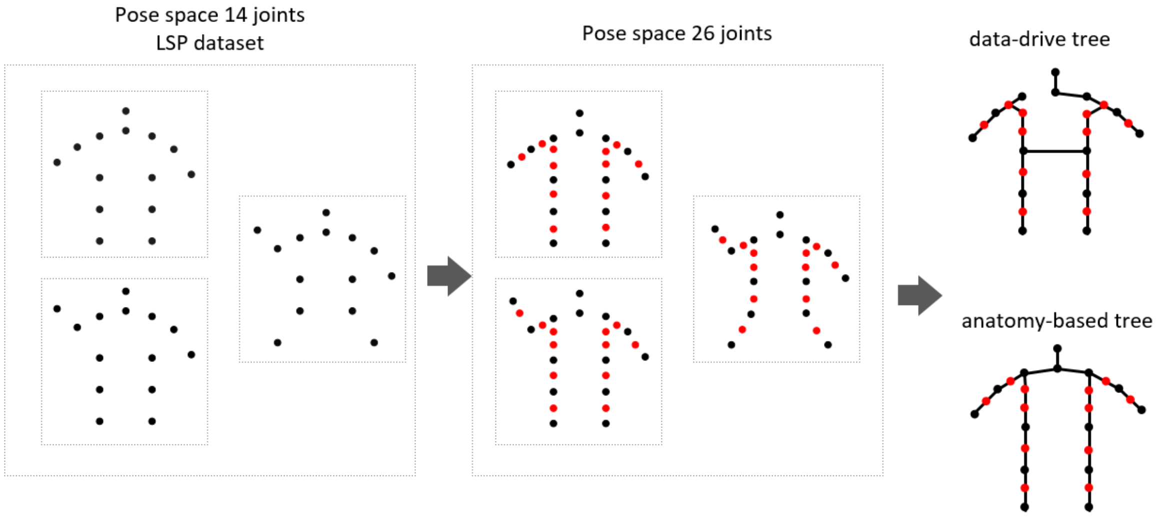


Fig. 4.1 The tree formation in an anatomy-based tree-based HPE framework with black points denotes originals joints and red points represent added joints.

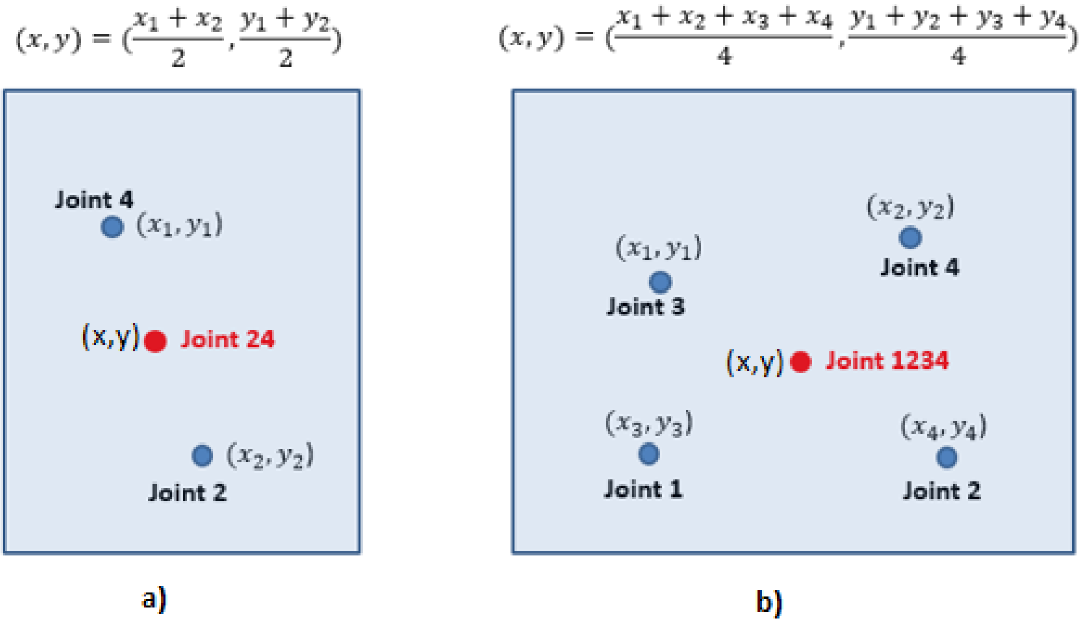


Fig. 4.2 Added joints in a tree-based HPE framework.: (a) Joint 4 or Joint 2 denotes the name of a general joint; Joint 24 is formed as the midpoint of Joint 2 and Joint 4. (b) Joint1234 (Figure 4.4) is formed as the centroid of Joint 1, Joint 2, Joint 3 and Joint 4.

calculating the arithmetic mean positions of neighboring joints. Fig. 4.2a) shows that joint 24 is obtained by taking the mean position of Joint 2 and Joint 4. Similarly, Fig. 4.2b) demonstrates the formation of joint 1234 as the mean position of joint 1, joint 2, joint 3 and joint 4.

In a tree-based model, nodes on the same edge show the parent-children or pairwise relationships. One way of determining edges is referred to the anatomy of human body. For example, a person has his left elbow connected to the left shoulder; therefore, these two body parts (or joints) would be on the same edge with regard to the human anatomy. Edges can also be formed using grouping methods to construct data-driven trees.

Data-driven trees presented in this chapter used the CLRG algorithm [103]. The algorithm first group observed nodes that were likely to be close to each other and then followed a process of recursive grouping. During the recursive grouping process, the algorithm used distance information to obtain sibling groups and recursively build a tree-based structure. Given x_i and x_j as observed random variables, the correlation coefficient is defined as

$$p_{ij} = \frac{Cov(x_i,x_j)}{\sqrt{Var(x_i),Var(x_j)}} \tag{4.1}$$

and the information distance is defined as

$$d_{ij} = -log(p_{ij}) \tag{4.2}$$

The relationship between each triple i,j,k ∈ V is determined based on the result of $\phi_{ijk} = d_{jk} - d_{ik}$. In case that $\phi_{ijk} = d_{ij}$, j is set as the parent of i. On the other hand, if for all $k \in V \setminus \{ij\}$, a hidden node is added as the parent of i and j. In this way, a latent tree is recursively built.

4.4.2 Proposed tree-based models

Table 4.1 shows the different tree-based configurations tested in this chapter's experiments, which are either based on the anatomy of the human body (the TA_14, TA_26, TA_30, TA_34_A, TA_34, TA_38, TA_50 configurations) or learned from the pose space of the LSP dataset using the CLRG algorithm (the TD_26 and TD_26_C configurations).

The anatomy-based configurations (the TA_14, TA_26, TA_30, TA_34_A, TA_34, TA_38, TA_50) contain different numbers of nodes: 14 nodes, 26 nodes, 30 nodes, 34 nodes, 38 nodes and 50 nodes. The 26-node tree (or the TA_26 configuration) refers to the tree proposed by [2]. In the case of the 14-node tree, the average distances between neighboring joints are large compared to the size of the geometric transform kernels. Thus, to model the relationships between the feature maps of these joints, the network requires large geometric transform kernels that are difficult to train [2]. As large kernels increase the network size, intermediate joints are introduced to reduce the distance between neighboring joints. The effect of these added joints (or added tree nodes) will then be investigated.

Table 4.1 Human pose models tested in the HPE experiments.

Name	**Pose Model**			
	Tree or Non-tree	**Anatomy or Data-driven**	**Number of Nodes**	**Fig.**
TA_14	Tree	Anatomy	14	4.3
TA_26 [2]	Tree	Anatomy	26	4.3
TA_30	Tree	Anatomy	30	4.3
TA_34	Tree	Anatomy	34	4.3
TA_34_A	Tree	Anatomy	34	4.3
TA_38	Tree	Anatomy	38	4.3
TA_50	Tree	Anatomy	50	4.3
TD_26	Tree	Data-driven	26	4.4a
TD_26_C	Tree	Data-driven	26	4.4b

The data-driven configurations (the TD_26 and TD_26_C configurations) have 26 nodes, in which 14 nodes represent the original 14 joints of the dataset and the other nodes represent additional joints formed as midpoints or centroids of existing joints (See Figure 4.2). These centroid-type joints are only used in the TD_26_C configuration, inspired by the tree-based representation described in [77].

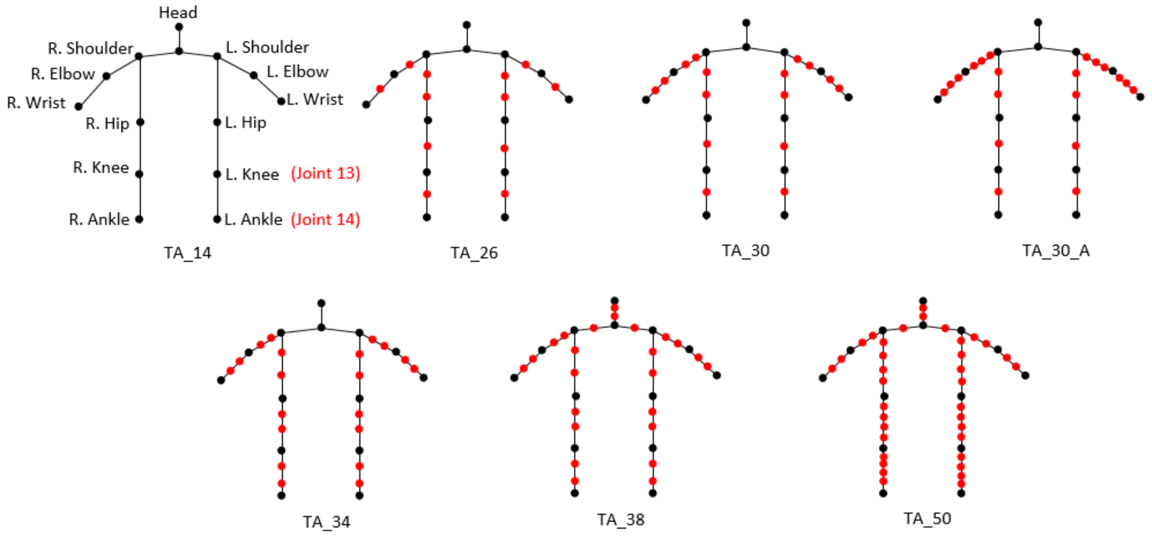


Fig. 4.3 The anatomy-based tree-based configurations.

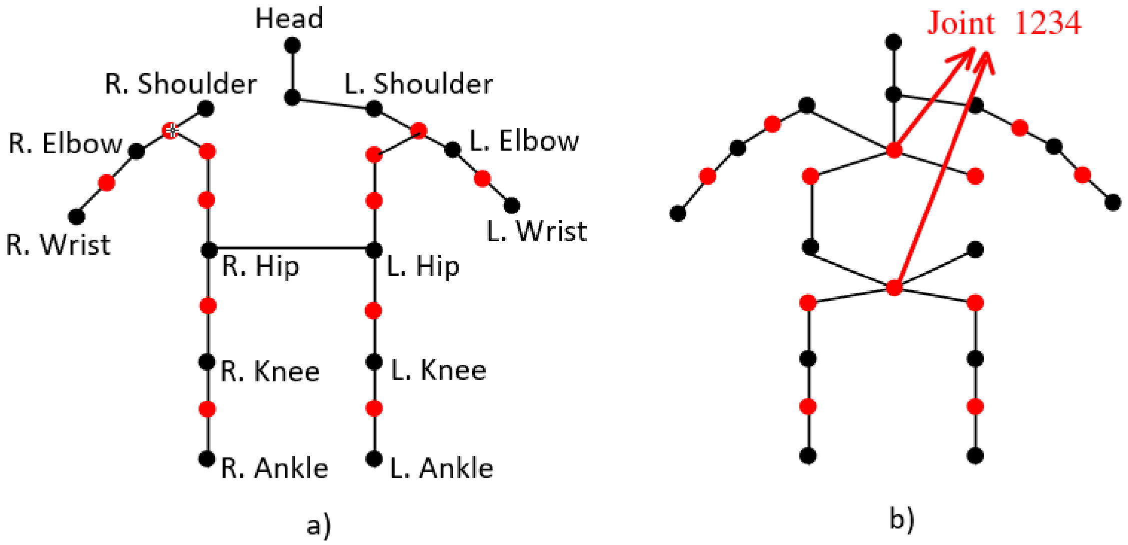


Fig. 4.4 The data-driven tree-based configurations.

4.5 System overview

4.5.1 The HPE framework

The system uses a graph G = (V, E) to model human poses where, V denotes vertices or positions of body joints, and the edges $E \subseteq V \times V$ specify the spatial relationship between the joints. Given an input image I, the full score $F(|)$ of a pose configuration is given as follows:

$$F(l,t|I;\theta,\omega) = \sum_{i\in V} \phi(l_i,t_i|I,\theta) + \sum_{i,j\in E} \psi(l_i,l_j,t_i,t_j|I,\omega_{i,j}^{t_i,t_j}) \tag{4.3}$$

where θ and $\omega_{i,j}^{t_i,t_j}$ are model parameters, $k = |V|$ specifies the number of parts (nodes); $i \in \{1,....K\}$ denotes the ith body joint; $l = \{l_i\}_{i=1}^{K}$ represents the pixel location of a part; $t = \{t_i\}_{i=1}^{K}$ denotes the mixture types of spatial relationships.

In the formula given by Equation 4.3, the pose configuration probability $F(|)$ contains the part appearance term (or the unary term) $\phi(l_i,t_i|I,\theta)$ and the spatial relational term $\psi(l_i,l_j,t_i,t_j|I,\omega_{i,j}^{t_i,t_j})$. While the appearance term provides local confidence of the appearance of a part i located at l_i, the relational term models the spatial relationship of two neighboring parts i and j.

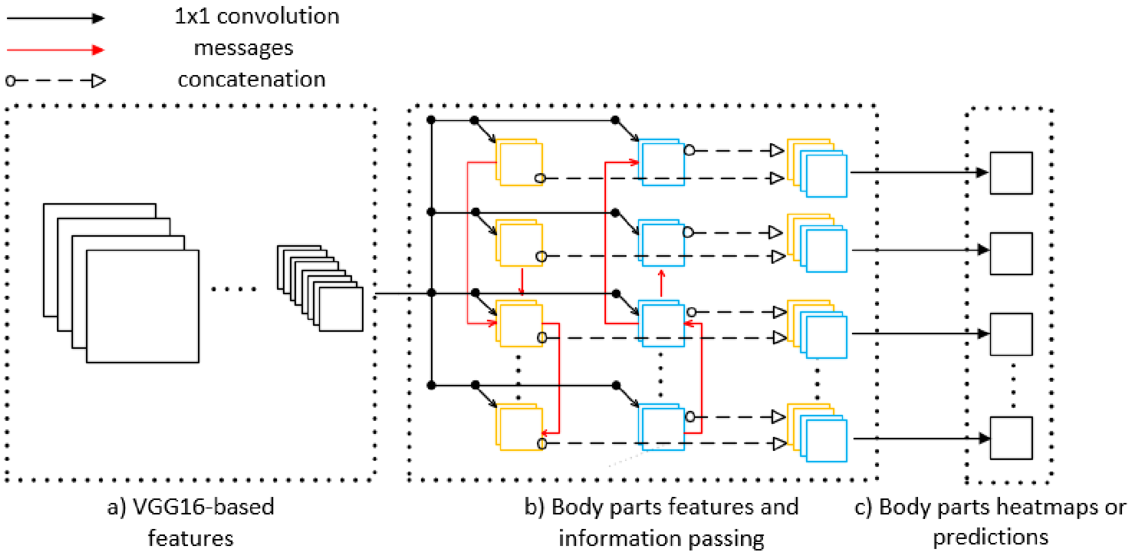

1x1 convolution
messages
concatenation
a) VGG16-based features
b) Body parts features and information passing
c) Body parts heatmaps or predictions

Fig. 4.5 The tree-based HPE framework (adapted from [2]): (1) VGG16-based features obtained using layers similar to VGG16. (2) Body parts features, as well as the refinement of these features, and information passing. (3) Body parts heatmaps or predictions: Yellow rectangles specify refined part features in the downward information passing, blue rectangles denote features in the upward information passing, and red lines indicate the direction of information passing.

The experiments described in this chapter are based on the HPE system proposed in [2], as illustrated in Figure 4.5. It consists of a pre-trained VGG16 image classification network [90] producing VGG16 features and a message passing network (MPN).

4.5.2 The VGG16-based network

The VGG16-based structure was converted from the VGG16 (VGG with 16 weight layers) as proposed by [90]. The conversion included removing the fully-connected pool4 layer and pool5 layer. The two pool layers were removed to keep prediction maps at a high resolution. The VGG16-based network inputted images of size 336x336 pixels and produced output feature maps of size 42x42 pixels. These feature maps played the role of appearance term (ϕ) as seen in Equation 4.3.

4.5.3 The tree-based message passing network

The function of the information passing network is to learn structural relations between feature maps of joints. This is achieved by passing feature messages (or shifted feature maps) through a tree-based structure in both upward and downward directions using geometric transform kernels. In a tree-based structure, messages are passed in a serial scheme; one message is passed at a time. The refined part-features obtained after message passing in upward and downward directions are next concatenated and convolved by 1x1 convolution layers to obtain part detection heatmaps (Fig. 4.5b)). These heatmaps predict the most likely positions of joints (Fig. 4.5c)).

The relationships between feature maps of joints were modeled using a tree-based structure as seen in Figure 4.3 (TA_26). Each body joint was represented by a set of 128 feature maps. All joints shared the fconv6 layers of the VGG16 network, which had 1024 feature channels. Feature maps of joints were passed from the leaf nodes to the root node (upward direction) and from the root node to the leaf nodes (downward direction). The refined feature maps in the upward direction would then be concatenated with those in the downward direction, generating 256 feature maps to predict the score map of one joint.

Let U_k denote the 128 feature map vectors of joint k in (1) and $U_k^{'}$ denote a vector of the refined feature maps of joint k after message passing in an upward direction in (2):

$$U_k = f(\phi_{fcn6} \otimes w^{k,up}) \tag{4.4}$$

$$U_k^{'} = f(U_k + \sum_{i \in children(k)} (U_j^{'} + w^{j,k})) \tag{4.5}$$

Where ϕ_{fcn6} denotes the feature maps of the fconv6 layer, $w^{k,up}$ is the filter banks for joint k in an upward message passing direction, f is the Rectified Linear Unit (RELU) and $w^{j,k}$ is the geometric transform kernels between joints j and k.

Taking nodes 13 and 14 of the tree in Figure 4.3 (TA_14) as an example, the feature maps of joint 13 and joint 14 can be represented by U_{13} and U_{14} as follows:

$$U_{13} = f(\phi_{fcn6} \otimes w^{13,up}) \tag{4.6}$$

$$U_{14} = f(\phi_{fcn6} \otimes w^{14,up}) \tag{4.7}$$

Since joint 14 is the leaf node, the node does not receive information from other joints in the upward direction. Therefore, the refined feature map of joint 14 ($U_{14}^{'}$) is equal to its original feature map (U_{14}).

$$U_{14} = U_{14}^{'} \tag{4.8}$$

In the upward direction tree, joint 13 receives information from its child, joint 14. Hence, the refined feature map of joint 13 is given as follows:

$$U_{13}^{'} = f(U_{13} + U_{14}^{'} \otimes w^{13,14}) \tag{4.9}$$

Similar to the downward direction, D_k represent the 128 feature vectors of joint k and $D_k^{'}$ denotes the updated feature maps of joint k after message passing in a downward direction.

The concatenation of two sets of updated feature maps for one joint k is then represented by $[U_k^{'},D_k^{'}]$, which serves to predict the score map of joint k.

4.6 Experiments

Table 4.2 HPE accuracy (using strict PCP evaluation protocol) for different tree-based configurations.

Configurations	Head	Torso	Upper arm	Lower arm	Upper leg	Lower leg	Mean HPE
TA_14	88	87.7	71.5	59.5	78.2	72.2	73.9
TA_26 ([2])	89.2	93.9	76.4	63.9	85.7	80.3	**79.6**
TA_30	88.6	93.6	77.5	65.9	85.9	81.1	80.3
TA_34_A	90.6	93.3	76.8	64.3	84.9	79.8	79.6
TA_34	89.5	92.6	76.6	65.5	87	82.3	80.5
TA_38	89.5	93	77.6	66.3	87	81.7	80.8
TA_50	90.5	94.1	76.6	65.7	87.8	82.9	81.1
TD_26	89	94.5	77	64.8	87.1	81.7	80.5
TD_26_C	87	93.6	74.4	63.5	85.7	80.6	79.1

4.6.1 Data setup

The HPE experiments were conducted on the LSP dataset [8]. The LSP dataset is a popular benchmark dataset containing 2000 images: 1000 images for training and 1000 images for testing. These images capture sports activities with full-body annotations. The annotations use the Person Centric (PC) style, where the left/right sides of body parts are labeled according to the viewpoint of the person being depicted. The PC annotations are converted to the Observer Centric (OC) style in the experiments following previous work by [2]. In addition to the LSP dataset, the INRIAPerson dataset [27], which does not contain people, was also used to provide negative training images to increases the system robustness to noise.

The results obtained were benchmarked against results obtained in the previous work by [2] using the metric of strict Percentage of Correct Part (strict PCP). Strict PCP only accounts for the single highest scoring estimation, and a body part is considered correct if its endpoints are within 50% of the length of the ground-truth endpoints [1].

4.6.2 Data augmentation

Since 1000 LSP images are insufficient for the training and thus have the potential for over-fitting, existing pre-trained VGG16 weights [90] were used to initialize the system and perform fine-tuning for the task of human pose estimation. To increase the number of images for the fine tuning, additional images were created by augmenting the original LSP pictures. Each LSP image was flipped horizontally and rotated 39 times with angles sampled incrementally in the range from -171° to 180°. As a result, an additional 78000 training samples were generated from the original 1000 LSP images.

4.7 Results and discussion

4.7.1 Data-based- and anatomy-based tree-based models

As seen in Table 5.2 and Figure 4.6, for the same set of joints, the proposed data-driven representation (TD_26) obtains 0.9% higher HPE accuracy compared to the anatomy-based representation (TA_26, the original result [2]). This result demonstrated that given the same set of tree nodes, the way in which the nodes were connected could have a significant effect on the learned structure between joints.

4.7.2 Different data-driven representations

In contrast to TD_26, the TD_26_C configuration has 2 tree nodes that represent centroid-type joints formed as centroids from a subset of existing joints. The TD_26_C representation (data-driven model) obtains a mean HPE accuracy 1.4% lower than the TD_26 representation (79.1% vs 80.5%, illustrated in Table 5.2 and Figure 4.6). One of the possible explanations for the decreased HPE accuracy was the relatively large distance between some of the neighboring joints, where the distance between joints was calculated based on the distance in

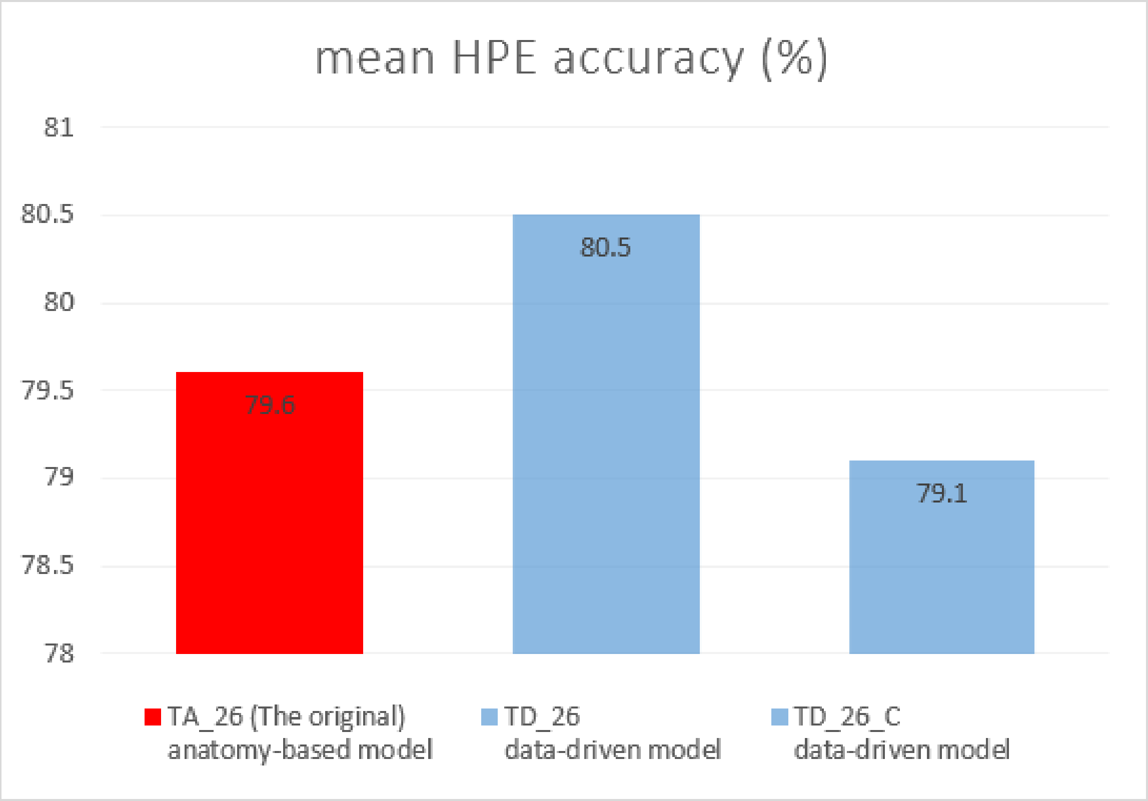


Fig. 4.6 Mean HPE accuracy of anatomy-based and data-driven tree-based models.

high-level features. In the TD_26_C representation, the distance between some neighboring joints whose nodes were on the same edge was larger than 9 pixels. Meanwhile, only two consecutive 7x7 geometry transform kernels (which was equivalent to one 9x9 kernel) were used to learn the deformation model between two neighboring joints. These 2 kernels were targeted for a high-level joint distance of less than 9 pixels. It was therefore possible that given a joint distance of more than 9 pixels, the kernels were not able to learn effectively, thus leading to decreased HPE accuracy.

4.7.3 Varying the number of tree nodes

To find an optimal number of tree nodes given the average distance among neighboring body joints (referred to as AD), experiments with different numbers of joints on the upper arm and lower arm were conducted. Added joints (corresponding to added tree nodes) to a body part reduced the distance between neighboring joints (joints represented by nodes on the same

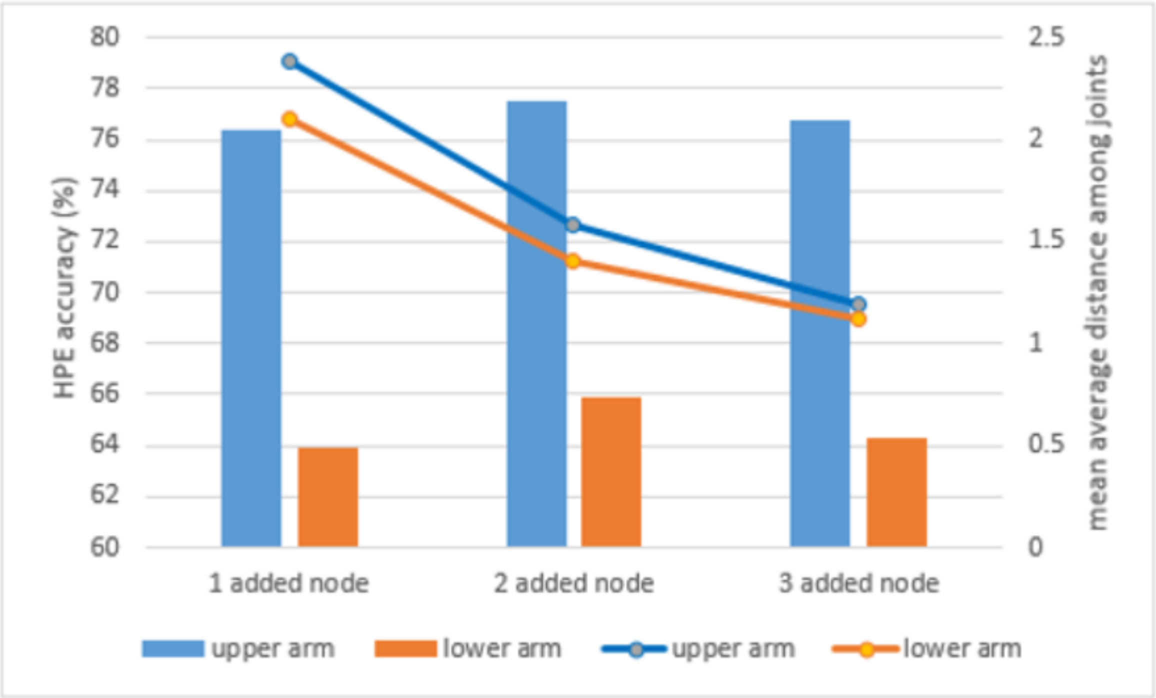


Fig. 4.7 HPE accuracy of lower arm and upper arm when more tree nodes are added.

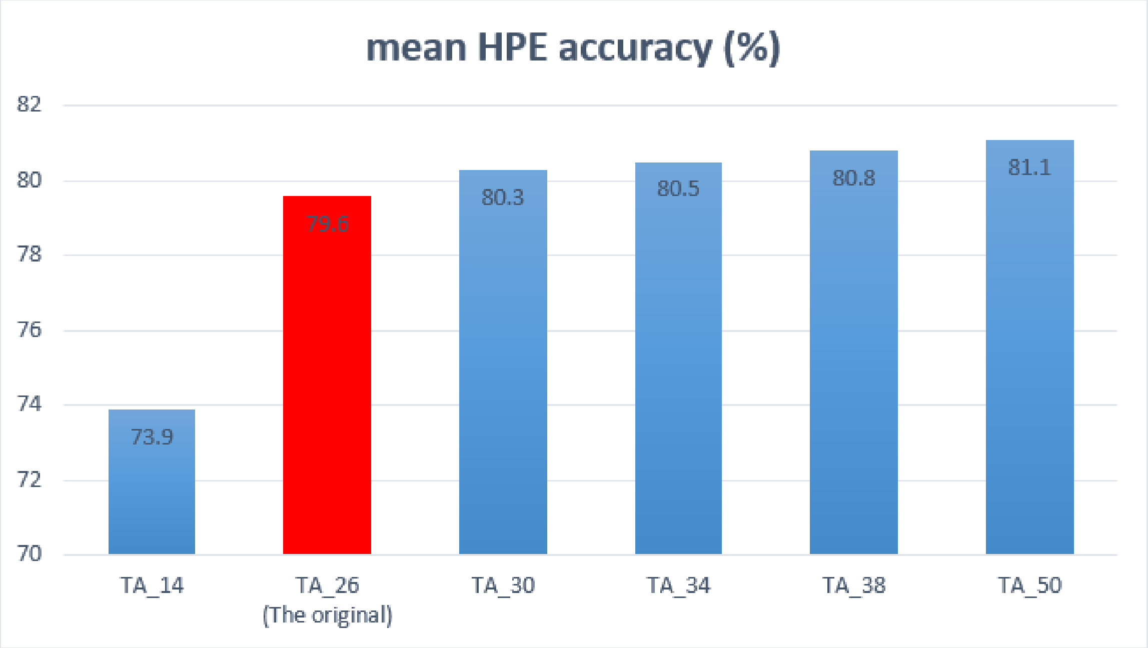


Fig. 4.8 Mean HPE accuracy for trees with varied numbers of nodes.

tree edges). As seen in Figure 4.7, both the upper arm and lower arm achieve the highest HPE accuracy when the AD on the arm is approximately 1.5 and the HPE accuracy decreases

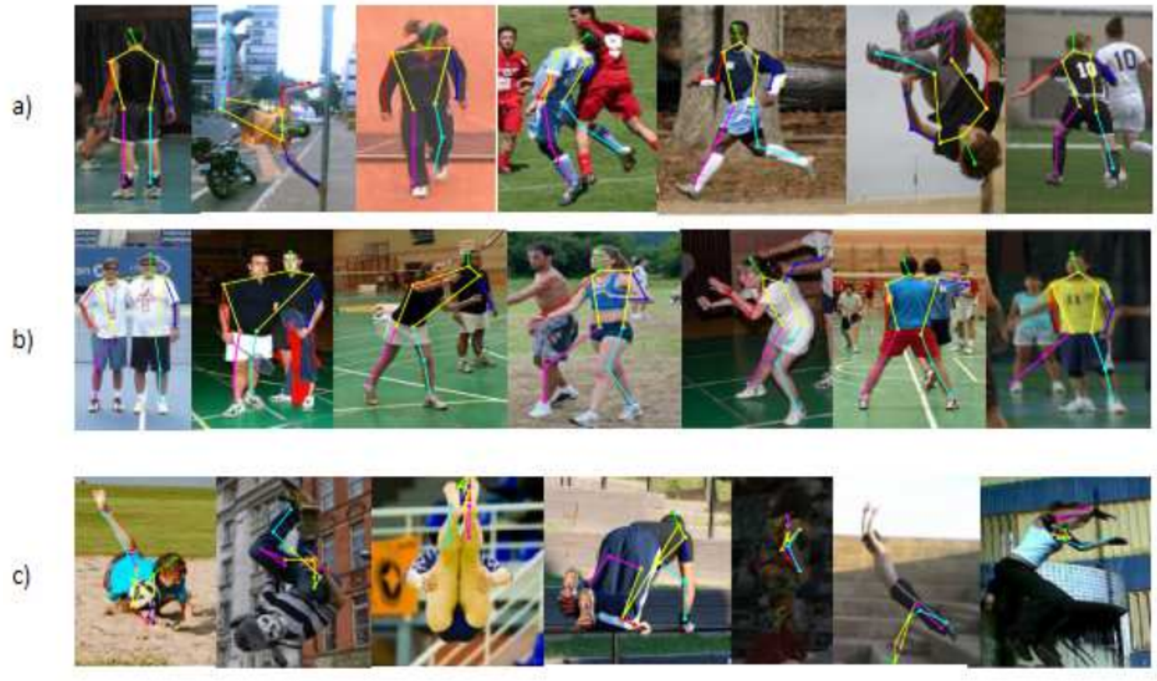


Fig. 4.9 Intuitive estimation results on the LSP dataset. a) Good estimation results. b) Bad estimation results caused by surrounding people. c) Bad estimation results caused by strong pose articulation and low image quality.

when AD is close to 1. When the average distance between two neighboring body joints was less than 1, the transform kernels between these two joints failed to learn as a kernel stride of 1 was used. Therefore, when the AD approached 1, transform kernels were not efficiently trained, thus leading to decreased HPE accuracy. On the other hand, with a large AD, the addition of intermediate joints generated more data for a network to learn, which resulted in an increased HPE accuracy.

In summary, experiments suggested that the AD value of approximately 1.5 provided an optimal average distance between neighboring joints leading to an optimal tree-based representation. The optimal tree-based structure contains 50 nodes (the TA_50 configuration in Figure 4.8. Figure 4.8 shows that the 50-node representation obtains a mean HPE accuracy of 81.1%, that is 1.5% higher than the HPE accuracy obtained when using the original TA_26 representation (79.6%). With four added nodes for both lower legs (AD = 1.44), the lower leg accuracy is increased by 2.6% (from 80.3% to 82.9%). With four added nodes for both upper legs (AD = 1.4), the lower leg accuracy is increased by 2.1% (from 85.7% to 87.8%).

Figure 4.8 shows that the lowest mean accuracy of 73.9% is achieved for the 14-node representation. This example illustrates a tree having a small number of nodes and thus, a large number of neighboring joints with a joint distance larger than 9 pixels that cannot be covered by geometric transform kernels.

4.7.4 Discussions on HPE results

Figure 4.9 shows as an example intuitive HPE results for the LSP dataset using the model of TD_26 configuration. Good estimations are displayed on Figure 4.9a. In contrast, the estimations in Figure 4.9b are poorer for some images that contain more than two people. In these images, the estimated body parts of one person are mixed up with the body parts of other people in close proximity. The number of these erroneous estimations is high, effectively reducing the average HPE accuracy. These estimation errors are due to the simple post-processing techniques used in the existing framework [2] and can be overcome by applying post-processing techniques for multi-person estimation as suggested by [19]. Figure 4.9c also illustrates erroneous estimations, where the human poses are either infrequently seen in practice or the image quality is poor.

4.8 Conclusion

This chapter compares data-driven tree-based models with existing human anatomy-based tree-based models for CNN-based HPE. Experimental results showed that tree-based models learning from data using the CLRG algorithm obtained approximately 1% higher accuracy than human body anatomy-based models. In addition, the optimal number of nodes was analyzed, establishing the distance between neighboring joints as an influencing factor. The optimal average distance was determined to be approximately 1.5 on the LSP dataset, which resulted in the 50-node tree achieving a mean HPE accuracy 1.5% higher compared to the original 26-node tree.

Chapter 5

Non-tree-based models

5.1 Preview

As opposed to tree-based structures which can only model pairwise interactions among nearby body parts, non-tree-based structures are able to model high-order relationships among body parts. In recent times, non-tree-based structures have been formulated by CNNs; however, different non-tree-based models were not investigated. This chapter aims to answer research questions 4 and 5. It evaluates different CNN-based non-tree-based structures and compares them with tree-based models using a common framework and a benchmark dataset. In addition, the effect of different connections between body parts of the non-tree-based models on HPE accuracy is investigated. Experimental results showed that proposed non-tree-based structures obtained approximately 0.8% higher mean HPE accuracy compared to the original CNN-based non-tree-based model.

5.2 Introduction

Tree-based and non-tree-based models are parts of graph representation used for modeling structural information. In human poses modeling, body joints are denoted by nodes in a graph and edges connected among nodes specify relationships among the joints.

In a tree-based representation, pairwise relationships are established among body parts. The advantage of this representation is the exact inference and simple implementation. However, the pairwise relationship only encodes information among nearby body joints; therefore, it fails to model complex pose space [104, 77]. As a result, non-tree-based representation is proposed to model high-order relationship among body parts [71, 70]. This modeling uses approximate search scheme to optimize the output. The main drawback of non-tree-based models is the difficult implementation and the optimization procedure which may not converge.

Recent works have modeled human poses using CNN-based non-tree-based structures [21, 22]. However, different non-tree-based structures have not been evaluated. This absence of evaluation motivates the research to analyze different CNN-based non-tree-based configurations for HPE and compare them with CNN-based tree-based models using a common framework and a benchmark dataset.

5.3 Related works

Before the introduction of CNN to the HPE, several non-tree-based representations extended the body part modeling beyond pairwise links. Jiang and Martin [72] combined tree-based and non-tree-based structures in a graph representation with strong (tree) edges to enforce arbitrary constraints and with weak (non-tree) edges to express the mutual exclusivity of inter-part occlusions and symmetric conditions. To further encapsulate the complexity of relations between body parts, Tran and Forsyth [73] proposed a full-relation modeling of body parts by creating a comprehensive set of the dependencies of body parts. Another important representation presented the hierarchical structure of body parts [70, 71], which included single rigid parts such as the torso, head, wrist and parts that contain more than one rigid element. Finally, a number of recent studies apply CNN to model the structural relationships using non-tree-based models [22, 21, 105].

The non-tree-based modeling described in this chapter is based on the framework proposed by Yang et al. [22] and Chu et al. [21]. Yang et al. [22] formulated the spatial model for body parts as Markov Random Field and learned it using the max-sum algorithm. Chu et al. [21], on the other hand, used Conditional Random Field and the sum-product algorithm to build the spatial model at feature level.

5.4 Proposed non-tree-based configurations

Table 5.1 contains three non-tree-based configurations: NT_26_A, NT_26_B, and NT_26_C. The configuration NT_26_A is similar to the one proposed by [21], while the NT_26_B and NT_26_C added more edges to the left and right of non-tree-based models corresponding to left and right body parts. As observed by Yang, et al. [21], a cascade of two or three message passing layers was sufficient to produce good results; therefore, two message passing layers were applied for these configurations, equivalent to two iterations of the message passing procedure. In addition, these proposed non-tree-based configurations were learned by the sum-product algorithm at feature level of CNNs, as opposed to the sum-product algorithm by [22] at heat-map level.

Table 5.1 Human pose models tested in the HPE experiments.

Configurations	Pose Model	
	Description	**Fig.**
NT_26_A [8]	2 iterations No loopy connections between left and right body parts, 26-node tree	8a
NT_26_B	2 iterations, 2 loopy connections between left and right body parts, 26-node tree	8b
NT_26_C	2 iterations, 5 loopy connections between left and right body part, 26-node tree	8c

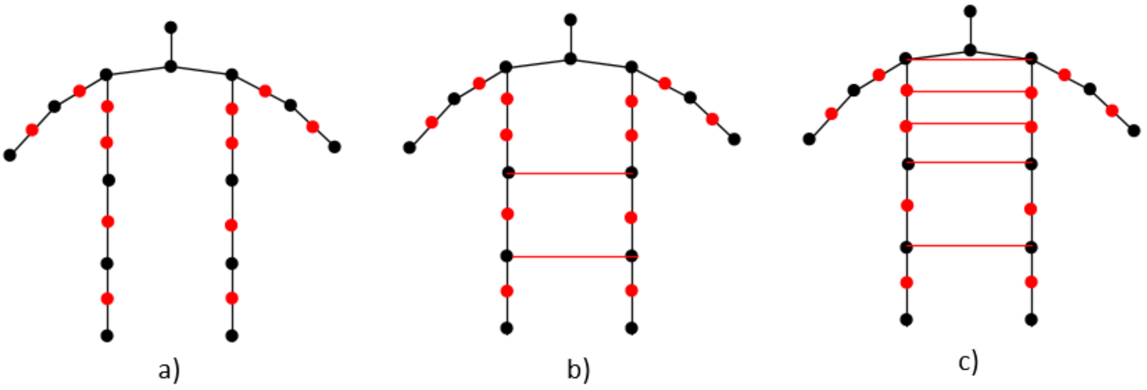


Fig. 5.1 Non-tree-based configurations.

5.5 System Overview

5.5.1 The HPE framework

The system uses the graph G = (V, E) to model human poses, where V denotes vertices or positions of body joints, and the edges $E \subseteq V \times V$ specify the spatial relationships between the joints. Given an input image I, the full score $F(|)$ of a pose configuration is given as follows:

$$F(l,t|I;\theta,\omega) = \sum_{i\in V} \phi(l_i,t_i|I,\theta) + \sum_{i,j\in E} \psi(l_i,l_j,t_i,t_j|I,\omega_{i,j}^{t_i,t_j}) \tag{5.1}$$

where θ and $\omega_{i,j}^{t_i,t_j}$ are model parameters, $K = |V|$ specify the number of parts (nodes); $i \in 1,\ldots.K$ denotes the ith part; $l = \{l_i\}_{i=1}^{K}$ represents the pixel locations of parts; $t = \{t_i\}_{i=1}^{K}$ denotes the mixture types of spatial relationships.

In the formula given by Equation 5.1, the score $F(|)$ contains the part appearance term (or the unary term) $\phi(l_i,t_i|I,\theta)$ and the spatial relational term $\psi(l_i,l_j,t_i,t_j|I,\omega_{i,j}^{t_i,t_j})$. While the appearance term provides local confidence of the appearance of a part i located at l_i, the relational term, on the other hand, models the spatial relationship of two neighboring parts i and j.

The experiments described in this study were based on the HPE system proposed by [2]. It consists of a pre-trained VGG16 image classification network [90] producing VGG16 features and a message passing network (MPN). The VGG16 network generated the appearance features while the MPN learned the spatial relationship features. In the VGG16 network structure [90] pool4 and pool5 layers were removed to keep the prediction maps at a high resolution level. The sizes of the input images and the corresponding output feature maps were 336x336 pixels and 42x42 pixels, respectively. In the MPN, both tree-based and non-tree-based representations applied the sum-product algorithm. Denoting C as a message sent from part i to part j by $m_{i},_{j}\,(l_i,l_j)$ and the belief of part j as $u_i(l_i,t_i)$, the algorithm proceeded as follows:

$$m_{i,j(l_i,l_j)} \leftarrow \sum_{l_i,t_i} u_i(l_i,t_i) \otimes \omega_{i,j}^{t_i,t_j} \tag{5.2}$$

$$u_i(l_i,t_i) \leftarrow \phi(l_i,t_i) + \sum_{k \in N(i)} m_{ki}(l_i,t_i) \tag{5.3}$$

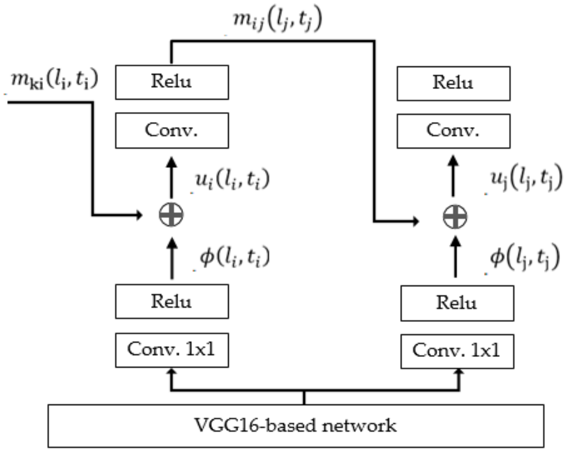


Fig. 5.2 A message passing from part i to part j within a CNN structure.

A flowchart of the message passing procedure between two adjacent body parts i and j is illustrated in Figure 5.2. Starting at the bottom of the graph and moving upwards, the output features from the VGG16 network (replicated for each body part) were convolved with the convolution layer 1x1 (conv. 1x1) to obtain the corresponding appearance term (ϕ). The belief parameter of each body part feature (u) was then updated by adding the appearance term (ϕ) to messages $m_{ki}(l_i,t_i)$ coming from the neighboring parts and sharing the same edge with the current part, as given by (5.3). This was next followed by the convolution with the updated belief to form the part message $m_{ij}(l_j,t_j)$, as given by (5.2). It is worthwhile to notice that the tree-based and non-tree-based representations used different mechanisms to pass messages. Namely, the tree-based structures used a serial message passing scheme in which one message was passed at a time, while the non-tree-based representations applied the flooding scheme where messages were passed simultaneously across every link at each time [22].

5.5.2 The non-tree-based message passing network

The non-tree-based HPE framework used the flooding message passing scheme where messages were passed simultaneously across every link. Suppose that in a given graph structure, the head and the neck share the same edge, and so do the neck and the left shoulder. This means that messages from head to neck, neck to head, neck to left shoulder, and left shoulder to neck were sent simultaneously. This scheme generated only approximate results and the message passing procedure needed to be iterated a number of times to obtain converged results [22].

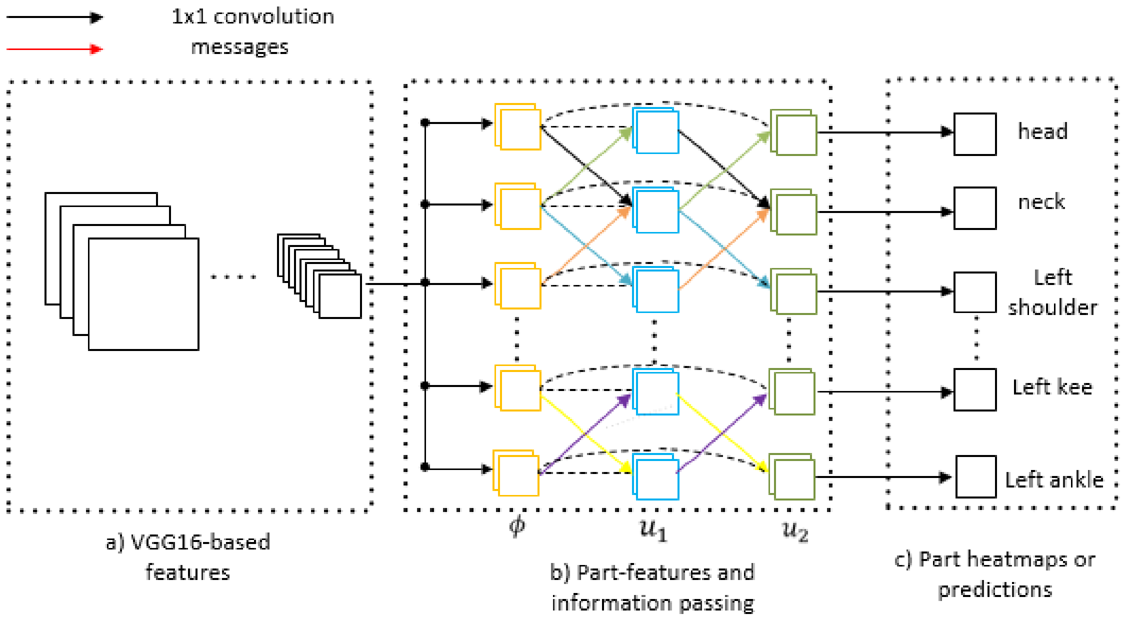


Fig. 5.3 The non-tree-based HPE framework (adapted from [22]): a) VGG16-based features obtained using layers similar to VGG16. b) Body parts features and the refinement of these features by information passing. c) Body parts heat maps (predictions).

The non-tree-based HPE framework used in this study is shown in Figure 5.3. It used the VGG network structure (with reference to the VGG 16 weight layers proposed by [90]) to obtain appearance features for each body part.

To learn spatial models using these body part appearance features, a non-tree-based message passing network was used. It included a cascade of two messaging layers equivalent to two iterations of the message passing procedure. Figure 5.3b) demonstrates the belief u_1 and u_2 corresponding to part beliefs after the first and second iteration respectively. In each iteration, nodes sent messages to their neighbors simultaneously. These messages are denoted by solid lines as demonstrated in Figure 5.3b). If the network converged after n iterations, the achieved belief of each body part u_n was considered to be the final result.

5.5.3 The implementation of non-tree-based message passing

Figure 5.5 illustrates the implementation of non-tree-based messaging for three body parts, including the head, neck and left shoulder. Modules in this implementation share the same architecture as shown in Figure 5.4. The construction of this module was based on the diagram in Figure 5.2. In Figure 5.5, modules of the same color illustrate data for the same body part but with different network weights, input and output.

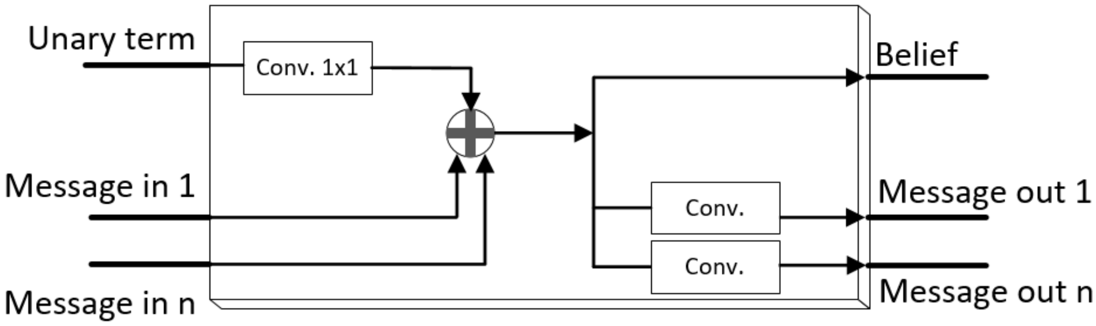


Fig. 5.4 The architecture of a module used in the implementation of non-tree-based message passing as seen in Figure 5.5.

At the beginning of each iteration, modules in block 0 were initialized with messages ’0’. At the iteration 1, messages from the previous block, block 0, were sent to block 1 (from head (block 0) to neck (block 1), neck (block 0) to head (block 1), neck (block 0) to left

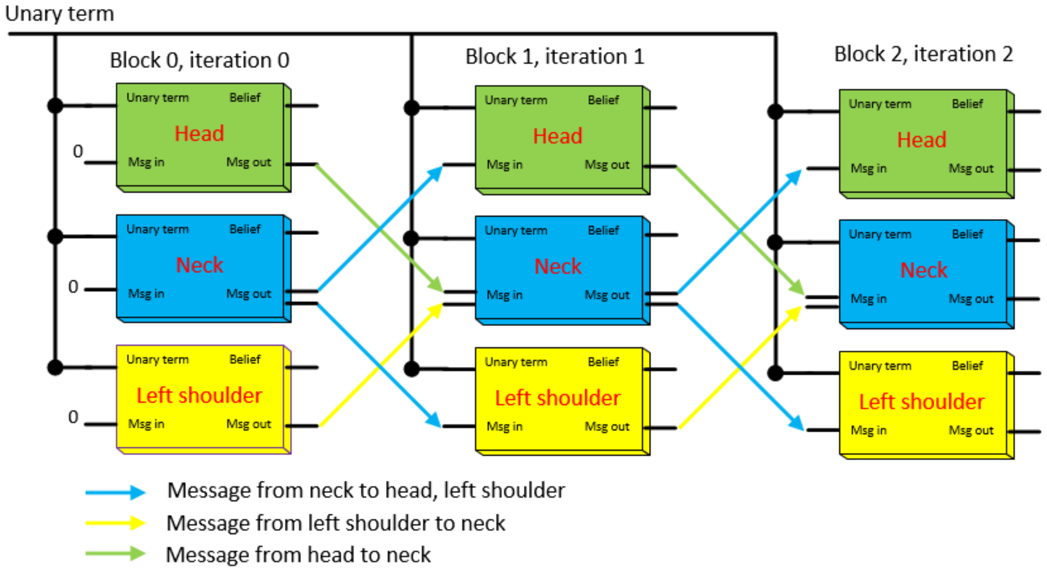


Fig. 5.5 Implementation of non-tree-based message passing with 9 modules, each of which shares the same architecture as seen in Figure 5.5.

shoulder (block 1) and left shoulder (block 0) to neck (block 1)) simultaneously. In the real implementation, the message updated in each module of block 0 were conducted in a serial manner. To achieve the simultaneous message passing from block 0 to block 1, after all modules of block had updated their output messages, output messages of block 0 would then be connected to input messages of block 1 at the same time. This procedure was repeated for messages from block 1 to block 2.

5.6 Results

5.6.1 Database

The HPE experiments were conducted on the LSP benchmark dataset [8], which contains 2000 images: 1000 images for training and 1000 images for testing. These images captured sports activities and came with full-body annotations. The annotations used the Person Centric (PC) style, where the left/right sides of body parts were labeled according to the viewpoint of the person being depicted. The PC annotations were converted to the Observer

Centric (OC) style following the previous study by [2]. In addition to the LSP dataset, the experiments also used the INRIAPerson images, which did not depict people [27]. The addition of the LSP data provided "negative" training and increased the system robustness to noise.

5.6.2 Performance measure and benchmarks

The HPE performance was assessed using the strict Percentage of Correct Part (strict PCP) measure [1]. It accounted only for the highest scoring estimation, and a body part was considered to be correctly identified if the relative distance between its estimated endpoints and the ground-truth endpoints was less than 50%. The ground-truth distance between the head and the neck nodes was used as the reference. The experimental results were benchmarked against results obtained in [2] and [22].

Since having only 1000 LSP training images is insufficient to train the network without the risk of over-fitting, transfer learning was applied. It was achieved by initializing the HPE network with the weights of an existing VGG16 network [90] pre-trained on a very large number of images depicting a large range of different objects. Thus, the experiments in this chapter only required a relatively short training (fine-tuning) and a small dataset to train the HPE network. To increase the number of images for the fine-tuning, additional images were generated by augmenting the original LSP pictures. Each LSP image was flipped horizontally using the Matlab function flipdim and rotated 39 times around the picture's centre point using the Matlab function imrotate with the rotation angles changed incrementally within the range from -171° to 180°. As a result, additional 78000 training images were generated and added to the original 1000 LSP images.

5.6.3 Comparison between different non-tree-based models

Table 5.2 shows the mean HPE accuracy for different non-tree-based configurations including NT_26_A, NT_26_B and NT_26_C. The original non-tree-based model proposed by [22] obtained a mean HPE accuracy of 77.6%. The proposed non-tree-based configurations

Table 5.2 HPE accuracy (using strict PCP evaluation protocol) for different non-tree-based configurations.

Configurations	**Head**	**Torso**	**Upper arm**	**Lower arm**	**Upper leg**	**Lower leg**	**Mean HPE**
TA_26 ([2])	89.2	93.9	76.4	63.9	85.7	80.3	**79.6**
NT_26_A [22]	88.6	93.5	74.1	59.7	84.2	79	**77.6**
NT_26_B	88.2	93.1	74.1	63.1	84.5	79.4	78.3
NT_26_C	87.4	94.3	74.6	62.3	84.6	79.8	78.4

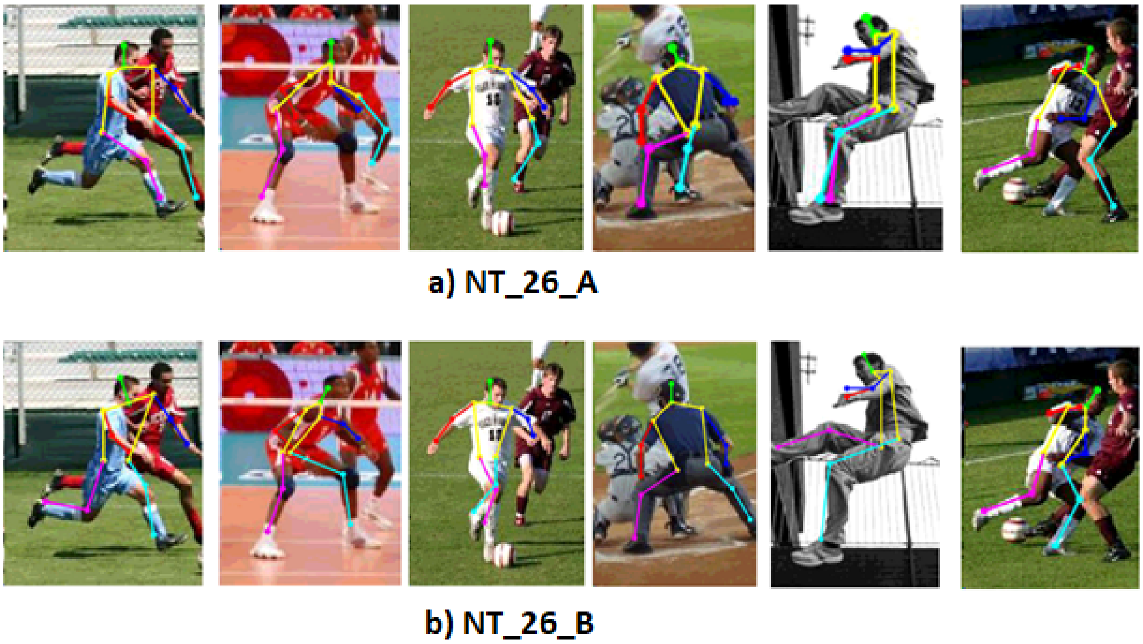
with additional modeling of left and right body parts (NT_26_B and NT_26_C) achieved the mean HPE accuracy of 78.3% and 78.4% respectively, which was approximately 0.8% higher compared to the original non-tree-based configuration. The experimental results demonstrated that additional modeling of the left and right body parts improved the spatial models of body parts and increased the overall mean HPE accuracy.

5.6.4 Comparison between tree-based and non-tree-based models

The tree-based configuration (TA_26) and non-tree-based configurations (NT_26_A, NT_26_B, NT_26_C) were tested on a single framework modeling spatial models of the body parts at feature level and applying sump-product algorithm for message passing. Table 5.2 shows that non-tree-based configurations obtained lower mean HPE accuracy of approximately 1% compared to the tree-based configuration. As [21] reasoned, the serial message passing for the tree-based models enabled each tree node to receive messages from all other nodes in an efficient way, resulting in higher mean HPE accuracy compared to the flooding message passing scheme for non-tree-based models.

5.6.5 Discussion on non-tree-based configurations

Compared to the original configuration (NT_26_A), the proposed configurations (NT_26_B and NT_26_C) added more edges to the left and right of non-tree-based models corresponding to left and right body parts. Visual results of the NT_26_A and NT_26_B are shown in Figure 5.6. In Figure 5.6 a), the legs of one person are confused with those of the other



a) NT_26_A

b) NT_26_B

Fig. 5.6 Visual HPE results of two non-tree-based configurations (NT_26_A and NT_26_B).

person nearby. This problem is not detected in Figure 5.6 b). Additionally, 5.6 b) captures the left and right legs of one person, including the skin and clothing, in the same color -which is different from that of the nearby person wearing different clothing. Thus, it is likely that apart from establishing a relational knowledge between the left and right body parts, the NT_26_B configuration also obtained color awareness. On the other hand, as there was no established connection among the left and right legs in the NT_26_A configuration, the system had mistaken the legs of one person with those of the other person.

5.7 Conclusion

This chapter evaluates different CNN-based non-tree-based structures and compares them with tree-based models using a common framework and a benchmark dataset. Both tree-based and non-tree-based configurations were modeled as Conditional Random Field and used the sum-product algorithm for message passing. Experimental results demonstrated that the proposed non-tree-based structures obtained lower mean HPE accuracy compared to tree-based models but achieved approximately 0.8% higher mean HPE accuracy as compared to the original CNN-based non-tree-based model.

Chapter 6

Hybrid models

6.1 Preview

Hybrid models such as dual-source CNNs, stacked and multitasking networks have been gaining in popularity. It has been shown that these complex structures can lead to an outstanding performance in many classification tasks. This chapter aims to answer research question 6. It proposes two original CNN-based hybrid models for the HPE. The first model is a double tree-based CNN (2T-CNN) structure, whereas the second model is a double-non-tree-tree-based CNN (2NT-CNN) configuration. The 2T-CNN configuration was trained with one CNN input, and each individual tree-based model was supervised separately, even though both shared the same base network. The 2NT-CNN configuration applied stacking for spatial models, in which a non-tree-based structure with two message passing layers was followed by a tree-based structure. Experimental results showed that the 2NT-CNN configurations obtained a mean HPE accuracy nearly 1% higher compared to tree-based or non-tree-based CNN structures alone.

6.2 Related works

Fan et al. [84] proposed a dual-source CNN consisted of two CNN sequences - one takes input as part patch (image patches containing a body part) and the other as body patch showing the whole body for HPE. In addition, the system was trained to perform two tasks - the joint location task and the detection task - in a unified network to achieve complementary effect to each individual task. Tompson et al. [81] introduced a hybrid architecture combining a CNN and a Markov Random Field. To improve scale invariance, this CNN was trained with two input image resolutions. Feature sizes generated by the two image resolutions were different and a Point-wise Upscale was deployed to generate the same size features so that they could be concatenated to form a unified network. Different from these two CNNs, the 2T-CNN configuration proposed in this chapter was trained with only one CNN input. In this configuration, each individual tree-based model was supervised separately, but both tree-based model shared the same base network, the VGG, as shown in Figure 6.1.

Jiang and Martin [72] introduced a novel global pose estimation modeled in a graph with strong (tree-based) edges to enforce arbitrary constraints and weak (non-tree-based) edges to express exclusive constraints from inter-part occlusion and symmetric conditions. However, this system used handcrafted features which did not achieve high expressive power. On the other hand, the 2NT-CNN framework proposed in this chapter used CNN-based features and the whole training, in addition to the optimization procedure performed using CNNs.

Another popular combined structure is the network stacking in which the output of one network is used as the input of other networks in a unified framework. Toshev and Szegedy [83] were the first to introduce stacking for the task of CNN-based HPE. They combined three consecutive networks as the three stages of the estimation. Initial poses were estimated in the first stage; networks would be used to refine the initial estimation in the following stages. In an end-to-end approach, Newell, et al. [20] performed pose estimation using a stack of eight networks. Bulat and Tzimiropoulos [24] also proposed stacking; they combined a detection and a regression network. An image was feedforwarded through the detection network to obtain a detection heatmap. The heatmap would then be concatenated with the input image to generate input for the regression network. Inspired by the idea of network stacking, the research proposes 2NT-CNN configurations to stack a nontree-based network with a tree-based network. Different from the whole network stacking by Newell et al.

[20], this chapter proposes 2NT-CNN configurations which only stacked spatial models (the nontree-based model stacked with the tree-based model), as seen in Figure 6.2).

6.3 The double tree-based CNN (2T-CNN) configuration

The network diagram for the 2T-CNN configuration is shown in Figure 6.1. This network is based on the structure learning framework proposed by [2] containing three main building blocks. The first block (Figure 6.1 (1)) is the pre-trained VGG-16 [90] with two pooling layers removed to keep features in high resolution. In the second block (Figure 6.1(2)), a combination of tree-based structures is introduced where two different tree-based models are trained in parallel and supervised separately. Single-tree-based model (tree1 in Figure 6.1) represents human anatomy [22], while the other tree-based model (tree2) is obtained using the CLRG algorithm [103] applied on the LSP dataset. Because these two tree-based models are supervised separately, the third building block contains two groups of heatmaps (Figure 6.1 (3)) corresponding to the two output of each tree-based model. The final pose is obtained using only the heatmaps of tree1 (Figure 6.1 (3)). Both tree1 and tree2 shared the same VGG16-based networks; therefore, the back-propagation mechanism possibly creates a complementary effect on the heatmap results of the both trees

6.4 The double-non-tree-tree-based CNN (2NT-CNN) configurations

6.4.1 Proposed configurations and diagram

The three 2NT-CNN configurations in Table 6.1 include H_26_1, H_26_2A and H_26_2B. The H_26_1 configuration contains a non-tree-based structure with two message passing layers followed by a tree-based structure with a single loss function applied to the whole network (Figure 6.2). The H_26_2A configuration (Figure 6.3) has a similar structure to the previous configuration, except that two loss functions are used -one for the non-tree-based

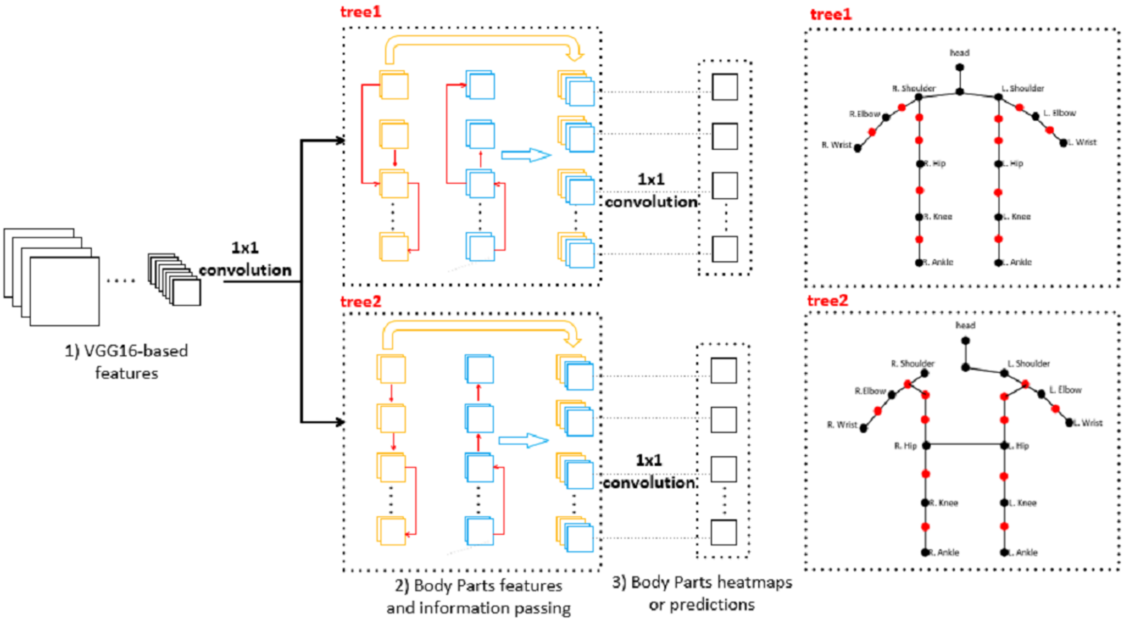


Fig. 6.1 The double tree-based CNN (2T-CNN) diagram: (1) VGG16-based features obtained using layers similar to VGG16 [90]. (2) Body Parts features and the refinement of these features by information passing. (3) Body Parts heatmaps or predictions.

Table 6.1 The double-non-tree-tree-based CNN (2NT-CNN) configurations.

Configurations	**Pose Model**			
	Types	**Number of Nodes**	**Descriptions**	**Fig.**
H_26_1	Hybrid	26	A single loss function	6.2
H_26_2A	Hybrid	26	Two loss functions	6.3
H_26_2B	Hybrid	26	Two loss functions and feature concatenation	6.4

part of the network and the other for the entire network- instead of a single loss function. Moreover, instead of passing the output from the non-tree-based network to the tree-based network input, the input and output of the non-tree-based network in the H_26_2B are concatenated to form a combined input to the tree-based network (Figure 6.4). This feature is inspired by the dense network proposed by [92] where all layers were connected to each other.

Diagrams for the three 2NT-CNN configurations are shown in Figure 6.2, 6.3 and 6.4, all of which contain three main building blocks. The first block uses the VGG-based structure

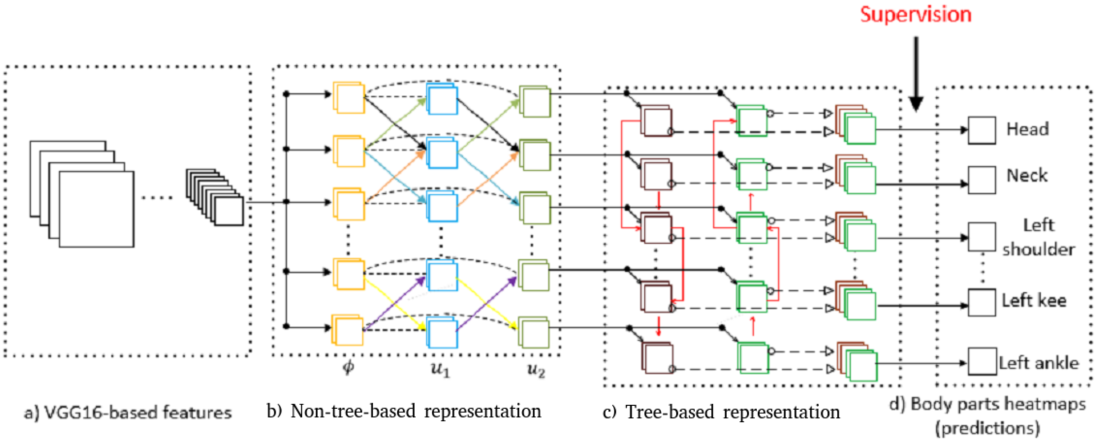

Supervision
Head
Neck
Left shoulder
Left kee
Left ankle
ϕ
u_1
u_2
a) VGG16-based features
b) Non-tree-based representation
c) Tree-based representation
d) Body parts heatmaps (predictions)

Fig. 6.2 The double-non-tree-tree-based CNN (2NT-CNN) diagram (the H_26_1 configuration): a) VGG16-based features obtained using layers similar to VGG16 [90]. b) non-tree-based representation c) tree-based representation d) Body Parts heatmaps or predictions.

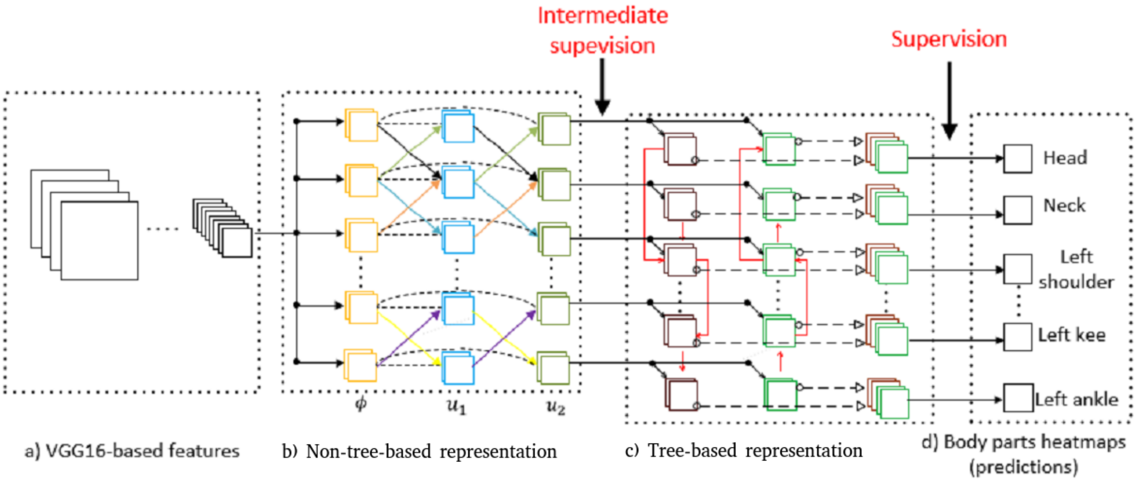


Fig. 6.3 The double-non-tree-tree-based CNN (2NT-CNN) diagram (the H_26_2A configuration): a) VGG16-based features obtained using layers similar to VGG16 [90]. b) non-tree-based representation c) tree-based representation d) Body Parts heatmaps or predictions.

(with reference to the VGG 16 weight layers proposed by [90]). The weights of this part of the network were generated during the pre-training process. During the training, the initial pre-trained weights were updated at a lower speed (a tenth of the pre-training rate). The inputs to the first building block were training images of size 336x336x3 pixels. The output features of the first building block (of size 42x42) were considered to be the appearance terms

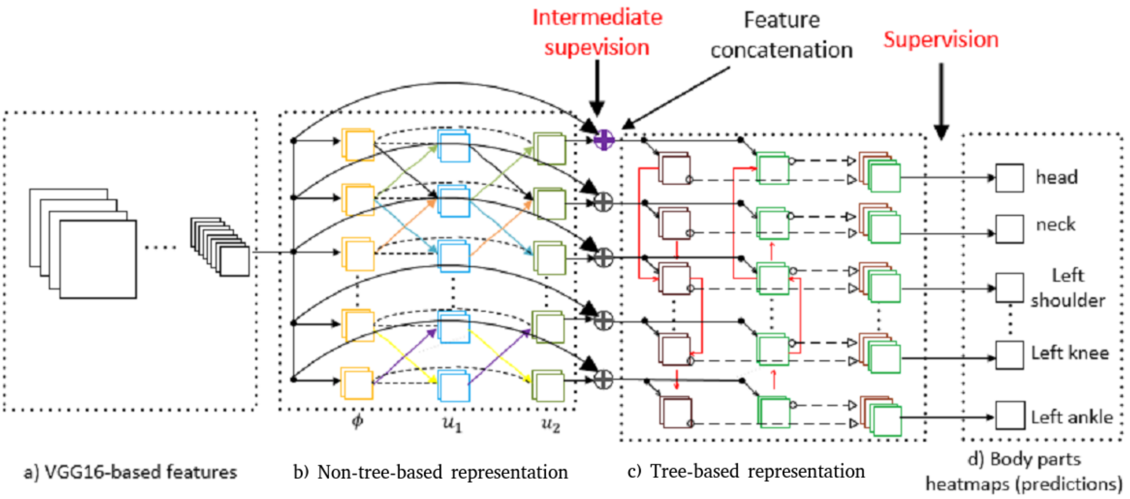


Fig. 6.4 The double-non-tree-tree-based CNN (2NT-CNN) diagram (the H_26_2B configuration): a) VGG16-based features obtained using layers similar to VGG16 [90]. b) non-tree-based representation c) tree-based representation d) Body Parts heatmaps or predictions.

providing local confidence values for each body part. In the second building block, feature maps of body parts were updated and refined through two iterations of the non-tree-based message passing network. The belief outputs of the second block were considered as the appearance features for the third building block, the tree-based message passing network proposed by [2]. The three building blocks were placed one after another. The proposed framework was trained using both a single loss function (for the configuration H_26_1) and two loss functions (for the configuration H_26_2A and H_26_2B).

6.5 Results

6.5.1 Comparison between different 2NT-CNN configurations

Table 6.2 shows the mean HPE accuracy for different 2NT-CNN configurations, which are hybrid models combining tree-based and non-tree-based structures. Since the depth of the combined network was significantly increased compared to a single network configuration, the system became prone to the vanishing gradient problem [20]. Therefore, it was understandable that the H_26_1 configuration with a single loss function (or one supervision) obtained a low accuracy of 78.35%. However when an intermediate supervision was additionally applied in H_26_2A and H_26_2B, the mean HPE accuracy increased to 80.2% and 80.5% respectively, approximately 2% higher than the single-loss configuration. In addition, the concatenation of features from different layers in the H_26_2B configuration led to an HPE accuracy 0.3% higher compared to the H_26_2A configuration (80.5% vs 80.2%). The hybrid configuration (H_26_2B) obtained an accuracy of 80.5%, which was nearly 1% higher compared to the HPE accuracy of either structure alone (i.e. the non-tree-based structure NT_26_A (77.6%, Chapter 5) and the tree-based structure TA_26 (79.6%, Chapter 4).

Table 6.2 HPE accuracy for single (1tree) and combined tree-based (2tree) configurations.

Configurations	Head	Torso	Upper arm	Lower arm	Upper leg	Lower leg	Mean HPE
TA_26 [2]	89.2	93.9	76.4	63.9	85.7	80.3	79.6
NT_26_A [21]	88.6	93.5	74.1	59.7	84.2	79	77.6
H_26_1	88.0	92.7	74.6	62.7	84.7	79.4	78.35
H_26_2A	88.9	94.5	77.2	64.8	86.1	81.2	80.2
H_26_2B	89.3	94.8	77.8	65.5	85.9	81.4	80.5

6.5.2 A comparison between the 2T-CNN and single tree-based configurations

Table 6.3 shows the mean HPE accuracy for the 2T-CNN and single tree-based (TA_26, Chapter 4) configurations. The 2T-CNN configuration obtained a mean HPE accuracy of 79.53%, slightly lower than the single tree-based configuration of 79.6%. In this experiment, the two-tree-based models in the 2T-CNN configuration was supervised separately. This can be improved by establishing correlation in learning of these two models so that knowledge learned in one-tree-based model can be complementary to the second one.

Table 6.3 HPE accuracy for a single tree-based and double tree-based (2T-CNN) configurations.

Configurations	Head	Torso	Upper arm	Lower arm	Upper leg	Lower leg	Mean HPE
TA_26 [57]	89.2	93.9	76.4	63.9	85.7	80.3	79.6
2T-CNN	88.2	93.9	75.4	64.8	86.2	80.2	79.53

6.6 Conclusion

Inspired by the popularity of CNN-based multitasking and network stacking, this chapter proposes the 2T-CNN and 2NT-CNN configurations. Sharing some characteristics with CNN-based multitasking networks, the 2T-CNN configuration contained two supervisions for two CNN-based tasks, one for an anatomy-based tree-based and the other for an data-driven tree-based structure. This dual-tasking configuration obtained a mean HPE accuracy of 79.53%, slightly lower than the single-tasking configuration of 79.6%. Future works would establish correlation in learning of these two models so that knowledge learned in one tree-based-based model can be complementary to the second one. The proposed 2NT-CNN configuration applied stacking for spatial models (a nontree-based model stacked with the tree-based model), in which a non-tree-based structure with two message passing layers was followed by a tree-based structure. Experimental results showed that the 2NT-CNN configuration obtained a mean HPE accuracy nearly 1% higher compared to the HPE accuracy of either the non-tree-based structure or the tree-based structure alone.

Chapter 7

Reflection on Research Questions, Future Works and Conclusions

7.1 Preview

This chapter discusses to what extent the study was able to answer the underlying research questions. It summarizes the thesis contributions, gives final conclusions, and outlines possible future research directions introduced in Section 2.4.

7.2 Reflection on research questions

The study was able to provide a number of insights into the various computational aspects of the HPE. It was able to provide at least partial answers to the research questions listed in Chapter 1. The following paragraphs summarize the findings corresponding to each of the 6 research questions.

Research Question 1 How to efficiently apply the CNN modeling to maximize accuracy of the body part recognition for HPE? The research found that the application of transfer learning at the body part recognition stage of the HPE can improve the overall HPE accuracy. As described in **Chapter 3**, a pre-trained CNN on a large FLIC dataset of general images

[30] was fine-tuned on a smaller LSP dataset. The application of transfer learning improved the HPE accuracy by 2% in comparison with the system proposed by [1] using a "freshly trained" body part recognition CNN. The research also showed that moderate improvements can be achieved through the optimization of the CNN pooling scheme and depth of network layers.

Research Question 2 What is the effect of different tree-based human pose models on the CNN-based HPE?

A new data-driven tree-based model was proposed and compared with the conventional anatomy-based model using the LSP dataset [8] on a structured CNN learning system proposed by [2]. As shown in **Chapter 4**, two versions of the proposed data-driven tree-based model, the TD_26 and TD_26_C, were trained using the CLRG algorithm. The TD_26, which applied only the original LSP nodes, achieved a mean HPE accuracy 1% higher than the anatomy-based configuration. The TD_26_C, which had additional nodes generated as centroids of the LSP nodes, displayed slightly lower performance than the anatomy-based model.

Research Question 3 How does the number of tree nodes used in modelling of human pose affect the CNN-based HPE accuracy?

The effect of the number of tree nodes on the HPE accuracy was examined in **Chapter 4** using the structured learning framework proposed in [2]. In general, it was observed that the addition of tree nodes increased the HPE accuracy at the same time that it increased the computational cost. This could potentially make the application impractical when taking into consideration the availability of hardware resources and computation time. In addition, the research found that the 50-node tree achieved an HPE accuracy of 81.8%, which was only 1.5% higher than the 79.6% of the frequently used 26-node tree-based model.

Research Question 4 How do the tree-based models compare with the non-tree-based models in terms of the HPE accuracy?

The advantage of the non-tree-based models is their ability to model complex relationships between body joints going beyond the pairwise links assumed by the tree-based models. A number of novel non-tree-based models were proposed and examined in **Chapter 5**. What differentiated the proposed models from previous models was the former's introduction of additional connections between the left and right body parts to improve the representation of the human structure. The proposed non-tree-based configurations resulted in higher HPE accuracy compared to the non-tree-based model proposed in [11]. Experimental results demonstrated that the proposed non-tree-based structures obtained lower mean HPE accuracy compared to tree-based models.

Research Question 5 What is the effect of different connections between body parts of the non-tree-based models on the HPE accuracy?

The introduction of additional connections to the non-tree-based models proposed in **Chapter 5** made possible the observation of effects of different connections to the HPE. Visual results showed that the non-tree-based models with additional edges among the left and right body parts (the proposed configuration NT_26_B and NT_26_C) do not confuse body parts of a person with body parts of a different person wearing different clothing nearby. With these additional connections, these proposed non-tree-based models have knowledge of the relationship between left and right body parts, which is missing in the orginal non-tree-based model (NT_26_A), resulting in an improved HPE accuracy.

Research Question 6 How to design an efficient hybrid structure combining both non-tree and tree-based models of the human pose?

Experimenting with both tree and non-tree representations of human pose showed that both types of models have their advantages and disadvantages. To compensate for the disadvantages of these models, an efficient hybrid structure was proposed in Chapter 6. This structure introduced a non-tree-based model with two messaging layers, which was followed by a tree-based model. By incorporating feature concatenation and intermediate supervision, the proposed hybrid structure obtained higher HPE accuracy than either individual structure alone.

7.3 Future work

This section describes several new concepts to be tested in future research.

7.3.1 Multi-tasking network structure for the HPE

Results of Figure 7.1 were obtained by superimposing on the original input images using a structural learning framework given in [2]. Each of the three images in Figure 7.1 shows people wearing similar color T-shirts. It indicates that these conditions were highly confusing and the system was not able to recognize that the pictures show two persons and not just one. It appears that in all of these examples, the joint-distance ratios of the detected body parts were not realistic. Hence, the outcomes did not represent valid human body configurations. For example, in all three examples of Figure 7.1, there is a clear disproportion between the unnaturally large distance from the neck to the left shoulder compared to the distances from the neck to the right shoulder. These errors can be attributed to the lack of sufficient distance-ratio constraints built into the model. Future studies could investigate the addition of these constraints to extend the HPE applications from pictures of a single person to pictures showing multiple people.

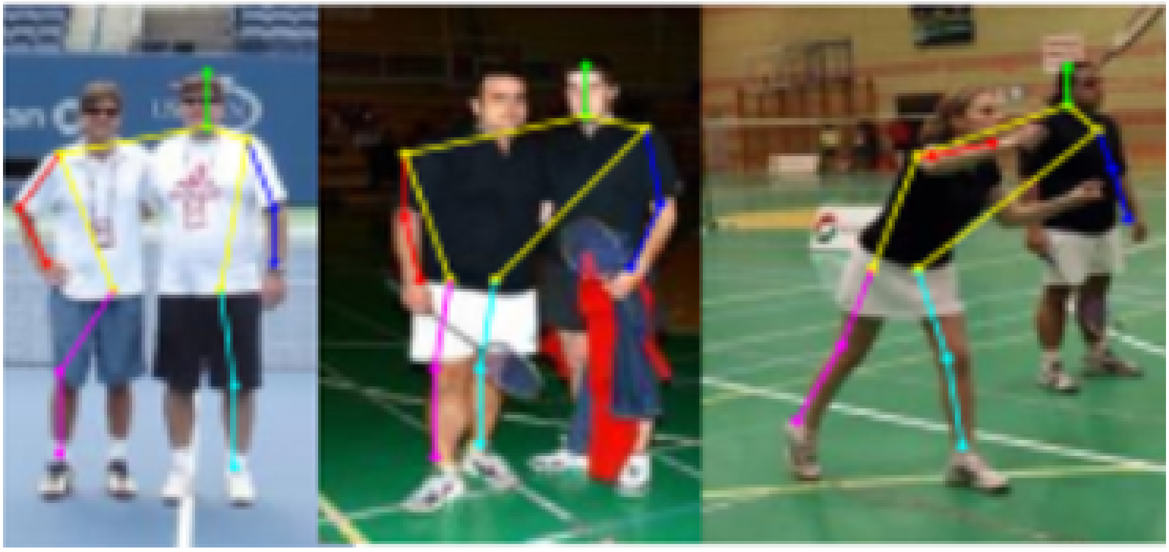


Fig. 7.1 Examples of incorrect HPE. Results were obtained by superimposing on the original input images using a structural learning framework given in [2]

As an example of potentially more powerful approach, a multi-tasking network that combines a detection and an estimation module could be investigated. The function of the detection module would be to detect individual people depicted in an image and outline the area occupied by each person. The HPE could then be applied within the outlined areas.

An example block diagram of a multi-tasking approach is shown inFigure 7.2. The estimation sub-network in Figure 7.2 is based on the structured learning framework by [2] while the detection sub-network was added to constrain the body parts to lie within the bounding box of a person. Both of the sub-networks share the same VGG16 [90] network features but were supervised separately.

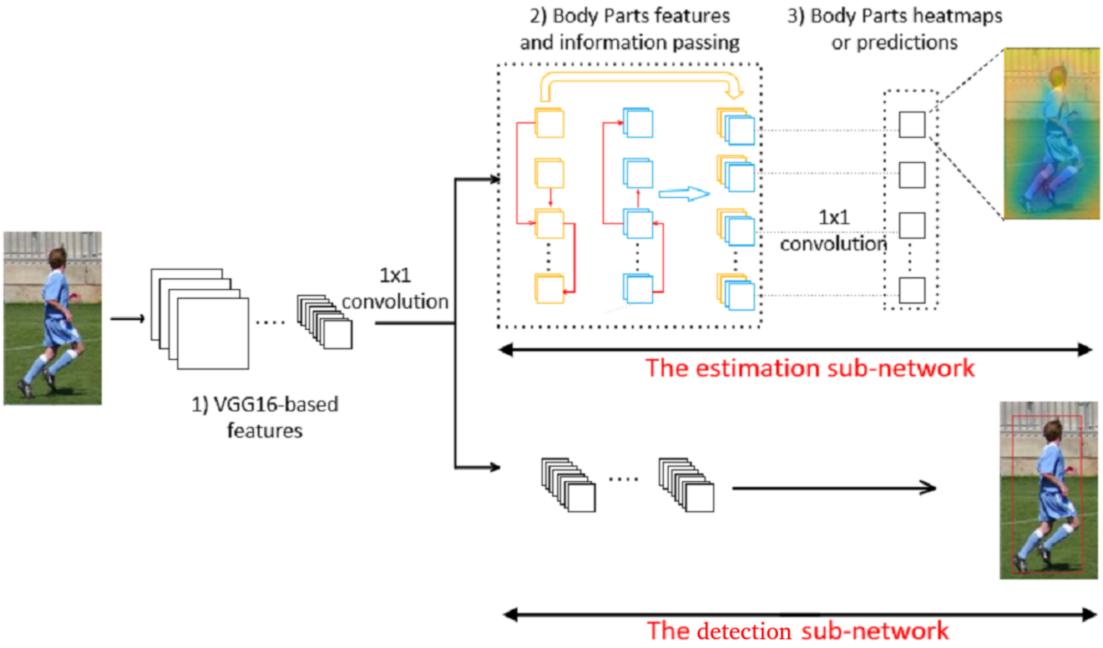


Fig. 7.2 Block diagram of a multi-tasking network for the HPE.

The detection-estimation network (DEN) was applied to perform preliminary tests of the multi-task training for the detection and estimation tasks in a unified framework. To train both tasks at the same time, two types of training labels were required: the labels for the estimation task given by body joint locations and the labels for the detection task given by the bounding box. Due to the limited number of training images, augmentation techniques were employed (image rotation and flipping) to increase the number of training data. When an image was augmented, new body part locations were obtained by augmenting the original body part locations the same way as the image. Some parts of the augmented bounding boxes were placed in the black background as shown in Figure 7.3. As the number of augmented images was relatively large, the black background problem had a significant contribution to the reduction of the HPE accuracy.

Solutions to the black background problem

There are three types of potential solutions to the black background problem. Examples of these solutions are illustrated in Figure 7.4. The first solution fills up the black background sections of the image with random noise (Figure 7.4 a)). The second solution applies padding to the colored area outside of the box outlining a person with the black pixels (Figure 7.4 b)). The third solution rotates and scales the bounding box to move it out of the black background area (Figure 7.4 c)).

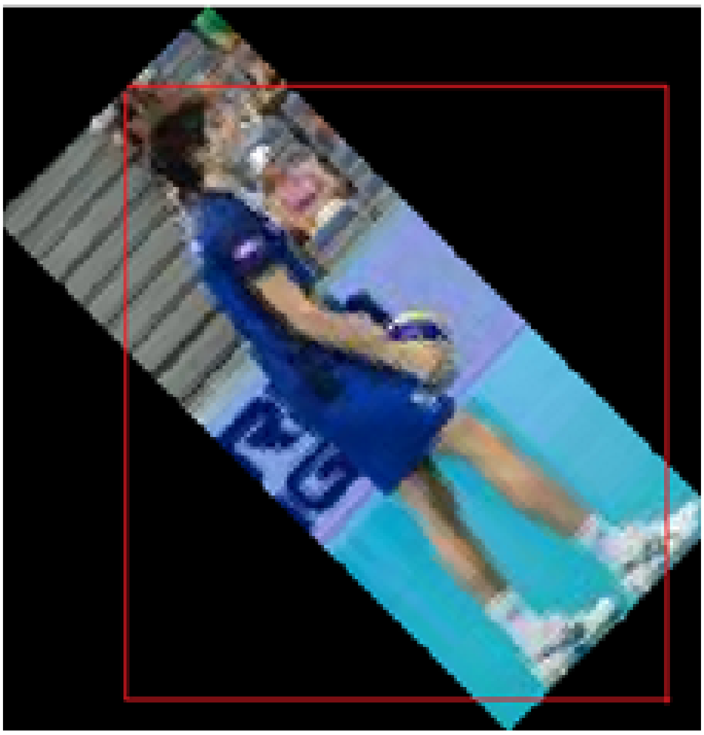


Fig. 7.3 The illustration of a bounding box of a person.

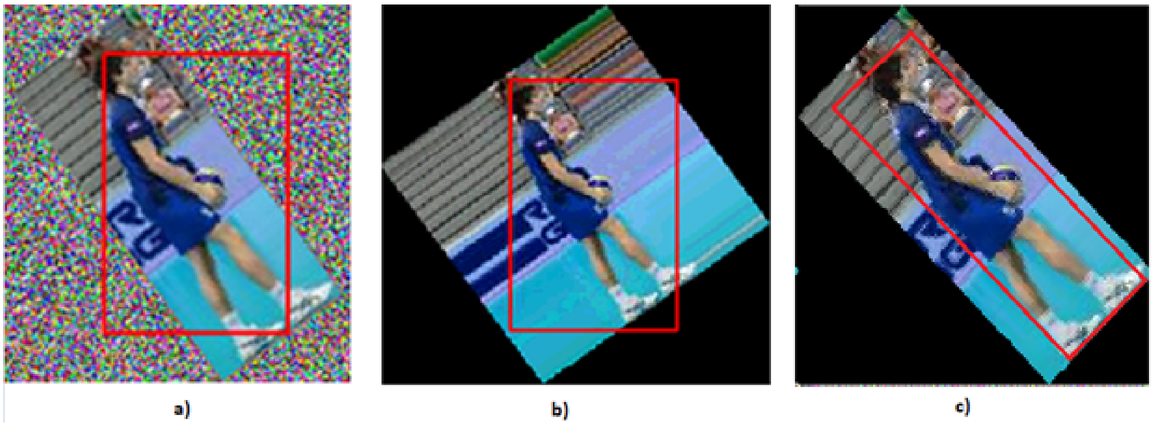


Fig. 7.4 Solutions for the problem of obtaining bounding box of a person.

Scale invariance

To improve the network scale invariance, Tompson et al. [81] introduced a hybrid architecture trained with two input image resolutions of size 320x240 pixels and 160x120 pixels. Incorporating scale invariance was further investigated by Newell et al. [20] through a symmetric network design with repeated downsampling, upsampling and feature concatenation. Motivated by these works, the research proposes new configurations listed in Table 7.1 to incorporate scale invariance to the existing structure learning framework by [2].

Table 7.1 Proposed configurations for scale invariance.

Configuration	Description	Figure
C1	Combine feature of different layers	7.5
C2	Combine feature of different layers	7.6
B1	Use Densenet [92] as based layers	-

The structured learning framework [2] described in Table 7.2 contains VGG16 layers [90] (conv1 to conv5) and a messaging passing layer. It is observed that the spatial resolution of the input (conv5_3) and output (fconv9) of the message passing layer is 42x42. The Configuration B1 is proposed to incorporate scale variance to this layer. (Table 7.1, Figure 7.5) concatenated fconv9 with conv4_1 and conv3_1 of different resolutions. Similarly, fconv9 can be combined with pool2 and conv3_1 as in Configuration C2 (Table 7.1, Figure

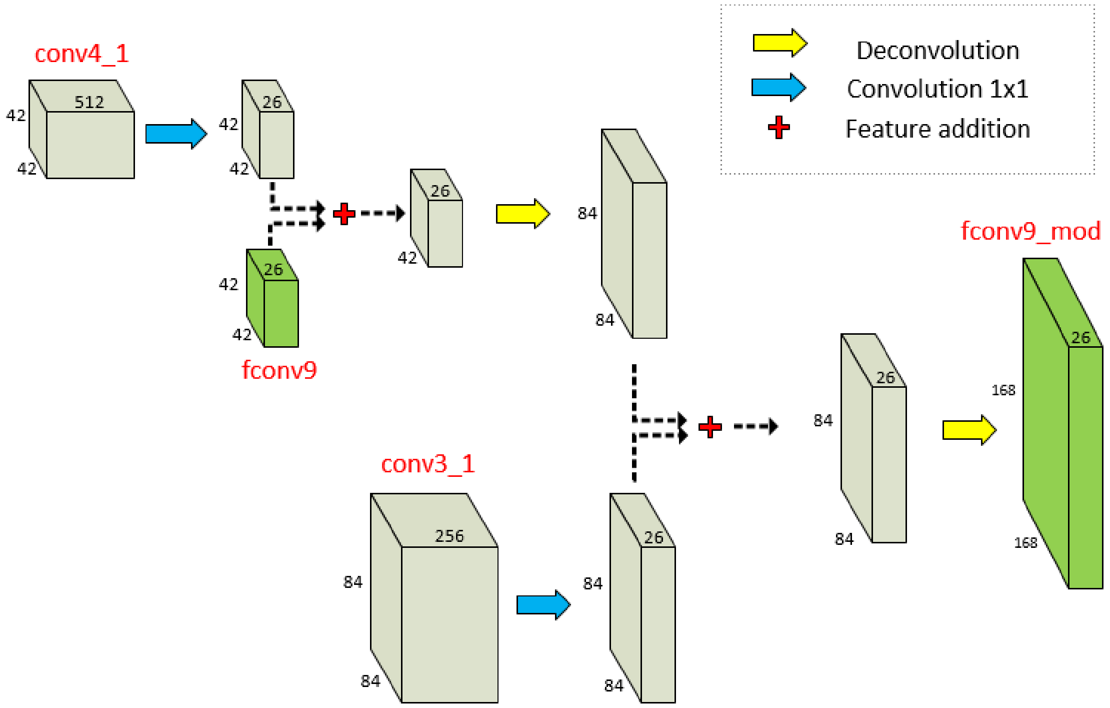


Fig. 7.5 The feature combination proposed for the HPE framework in [2](configuration C1 (Table 7.1)).

7.6). Configuration B1 is proposed to replace the base network VGG [90] by the popular Densenet [92].

7.4 Conclusion

This thesis investigated the incorporation of prior knowledge into CNNs through graph structures including tree-based and non-tree-based models. It was observed that both of the proposed data-driven tree-based models and hybrid approaches obtained higher HPE accuracy compared to the benchmark anatomy-based and non-tree-based models. The best overall HPE results were obtained when using the anatomy-based benchmark with an increased number of nodes. Non-tree-based and tree-based models were analyzed and compared using the same structured learning framework. A few proposed non-tree-based configurations obtained higher HPE accuracy compared to the existing non-tree-based models. Finally, a novel HPE framework that combined both a non-tree and tree-based network was introduced.

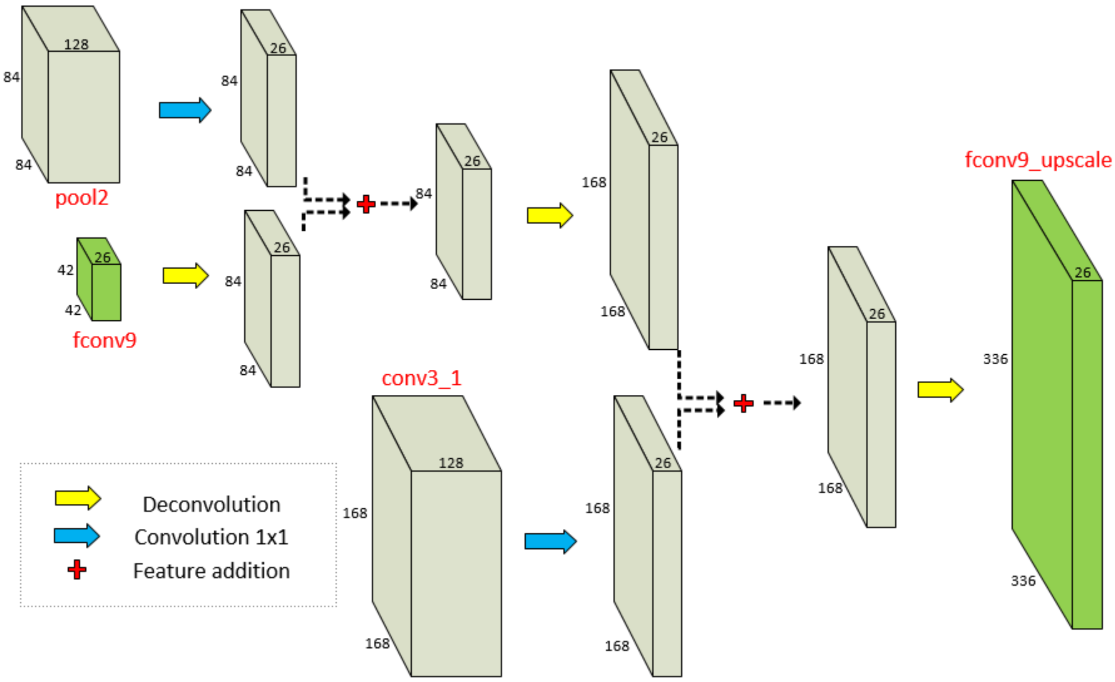


Fig. 7.6 The feature combination proposed for the HPE framework in [2] (configuration C2 (Table 7.1)).

This hybrid network obtained higher HPE accuracy compared to the HPE accuracy of either tree-base or non-tree-based structure alone. Future work will investigate network designs with feature concatenation from different levels of network hierarchy to improve scale invariance networks.

Table 7.2 Network structure of a structured learning framework [2].

layers	kernel	stride	output feature
input			3x336x336
conv1_1	3	1	64x336x336
conv1_2	3	1	64x336x336
pool1	2	2	64x168x168
conv2_1	3	1	128x368x168
conv2_2	3	1	128x168x168
pool2	2	2	128x84x84
conv3_1	3	1	256x84x84
conv3_2	3	1	256x84x84
conv3_3	3	1	256x84x84
pool3	2	2	256x42x42
conv4_1	3	1	512x42x42
conv4_2	3	1	512x42x42
conv4_3	3	1	512x42x42
conv5_1	3	1	512x42x42
conv5_2	3	1	512x42x42
conv5_3	3	1	512x42x42
message passing layers	-	-	
fconv9	-	-	26x42x42

References

[1] 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*, 2014, pp. 1736–1744.

[2] X. Chu, W. Ouyang, H. Li, and X. Wang, “Structured feature learning for pose estimation,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 4715–4723.

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

[4] M. D. Zeiler and R. Fergus, “Visualizing and understanding convolutional networks,” in *European conference on computer vision*. Springer, 2014, pp. 818–833.

[5] K. Chatfield, K. Simonyan, A. Vedaldi, and A. Zisserman, “Return of the devil in the details: Delving deep into convolutional nets,” *arXiv preprint arXiv:1405.3531*, 2014.

[6] A. Kolahi, M. Hoviattalab, T. Rezaeian, M. Alizadeh, M. Bostan, and H. Mokhtarzadeh, “Design of a marker-based human motion tracking system,” *Biomedical Signal Processing and Control*, vol. 2, no. 1, pp. 59–67, 2007.

[7] T. B. Moeslund, A. Hilton, and V. Krüger, “A survey of advances in vision-based human motion capture and analysis,” *Computer vision and image understanding*, vol. 104, no. 2-3, pp. 90–126, 2006.

[8] S. Johnson and M. Everingham, “Clustered pose and nonlinear appearance models for human pose estimation.” in *BMVC*, vol. 2, no. 4, 2010, p. 5.

[9] M. Ye, X. Wang, R. Yang, L. Ren, and M. Pollefeys, “Accurate 3d pose estimation from a single depth image,” in *2011 International Conference on Computer Vision*. IEEE, 2011, pp. 731–738.

[10] M. Andriluka, S. Roth, and B. Schiele, “Monocular 3d pose estimation and tracking by detection,” in *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. IEEE, 2010, pp. 623–630.

[11] R. Poppe, “Vision-based human motion analysis: An overview,” *Computer vision and image understanding*, vol. 108, no. 1-2, pp. 4–18, 2007.

[12] S. Ranasinghe, F. Al Machot, and H. C. Mayr, “A review on applications of activity recognition systems with regard to performance and evaluation,” *International Journal of Distributed Sensor Networks*, vol. 12, no. 8, 2016.

[13] D. Chen, A. J. Bharucha, and H. D. Wactlar, “Intelligent video monitoring to improve safety of older persons,” in *Engineering in Medicine and Biology Society, 2007. EMBS 2007. 29th Annual International Conference of the IEEE*. IEEE, 2007, pp. 3814–3817.

[14] F. Brémond, M. Thonnat, and M. Zúniga, “Video-understanding framework for automatic behavior recognition,” *Behavior Research Methods*, vol. 38, no. 3, pp. 416–426, 2006.

[15] M.-C. Chang, N. Krahnstoever, S. Lim, and T. Yu, “Group level activity recognition in crowded environments across multiple cameras,” in *Advanced Video and Signal Based Surveillance (AVSS), 2010 Seventh IEEE International Conference on*. IEEE, 2010, pp. 56–63.

[16] F. Fusier, V. Valentin, F. Brémond, M. Thonnat, M. Borg, D. Thirde, and J. Ferryman, "Video understanding for complex activity recognition," *Machine Vision and Applications*, vol. 18, no. 3-4, pp. 167–188, 2007.

[17] Y. Abe, C. K. Liu, and Z. Popović, "Momentum-based parameterization of dynamic character motion," in *Proceedings of the 2004 ACM SIGGRAPH/Eurographics symposium on Computer animation*. Eurographics Association, 2004, pp. 173–182.

[18] 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*, 2016, pp. 4724–4732.

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

[20] A. Newell, K. Yang, and J. Deng, "Stacked hourglass networks for human pose estimation," in *European Conference on Computer Vision*. Springer, 2016, pp. 483–499.

[21] X. Chu, W. Ouyang, X. Wang *et al.*, "Crf-cnn: Modeling structured information in human pose estimation," in *Advances in Neural Information Processing Systems*, 2016, pp. 316–324.

[22] 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*, 2016, pp. 3073–3082.

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

[24] A. Bulat and G. Tzimiropoulos, “Human pose estimation via convolutional part heatmap regression,” in *European Conference on Computer Vision*. Springer, 2016, pp. 717–732.

[25] X. Chu, W. Yang, W. Ouyang, C. Ma, A. L. Yuille, and X. Wang, “Multi-context attention for human pose estimation,” *arXiv preprint arXiv:1702.07432*, vol. 1, no. 2, 2017.

[26] M. A. Fischler and R. A. Elschlager, “The representation and matching of pictorial structures,” *IEEE Transactions on computers*, vol. 100, no. 1, pp. 67–92, 1973.

[27] N. Dalal and B. Triggs, “Histograms of oriented gradients for human detection,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, vol. 1, 2005, pp. 886–893.

[28] S. Johnson and M. Everingham, “Learning effective human pose estimation from inaccurate annotation,” in *Computer vision and pattern recognition (CVPR), 2011 IEEE conference on*. IEEE, 2011, pp. 1465–1472.

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

[30] B. Sapp and B. Taskar, “Modec: Multimodal decomposable models for human pose estimation,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 3674–3681.

[31] G. Gkioxari, P. Arbelaez, L. Bourdev, and J. Malik, “Articulated pose estimation using discriminative armlet classifiers,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 3342–3349.

[32] W. Yang, S. Li, W. Ouyang, H. Li, and X. Wang, “Learning feature pyramids for human pose estimation,” in *The IEEE International Conference on Computer Vision (ICCV)*, vol. 2, no. 7, 2017.

[33] Y. Yang and D. Ramanan, “Articulated human detection with flexible mixtures of parts,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 12, pp. 2878–2890, 2013.

[34] V. Ferrari, M. Eichner, M. Marin-Jimenez, and A. Zisserman, “Buffy stickmen v3. 01: Annotated data and evaluation routines for 2d human pose estimation,” 2013.

[35] “Mpii human pose dataset,” http://human-pose.mpi-inf.mpg.de, accessed: 2017-01-01.

[36] M. Fieraru, A. Khoreva, L. Pishchulin, and B. Schiele, “Learning to refine human pose estimation,” *arXiv preprint arXiv:1804.07909*, 2018.

[37] 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*, 2018, pp. 7103–7112.

[38] 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)*, 2018, pp. 417–433.

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

[40] B. Xiao, H. Wu, and Y. Wei, “Simple baselines for human pose estimation and tracking,” *arXiv preprint arXiv:1804.06208*, 2018.

[41] G. Ning, P. Liu, X. Fan, and C. Zhang, “A top-down approach to articulated human pose estimation and tracking,” *arXiv preprint arXiv:1901.07680*, 2019.

[42] W. Yang, W. Ouyang, X. Wang, J. Ren, H. Li, and X. Wang, “3d human pose estimation in the wild by adversarial learning,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, vol. 1, 2018.

[43] C. Zimmermann, T. Welschehold, C. Dornhege, W. Burgard, and T. Brox, "3d human pose estimation in rgbd images for robotic task learning," *arXiv preprint arXiv:1803.02622*, 2018.

[44] G. Rogez and C. Schmid, "Image-based synthesis for deep 3d human pose estimation," *International Journal of Computer Vision*, pp. 1–16, 2018.

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

[46] G. Mori and J. Malik, "Estimating human body configurations using shape context matching," in *European conference on computer vision*. Springer, 2002, pp. 666–680.

[47] J. M. Keller, M. R. Gray, and J. A. Givens, "A fuzzy k-nearest neighbor algorithm," *IEEE transactions on systems, man, and cybernetics*, no. 4, pp. 580–585, 1985.

[48] W. S. Cleveland, "Robust locally weighted regression and smoothing scatterplots," *Journal of the American statistical association*, vol. 74, no. 368, pp. 829–836, 1979.

[49] G. Shakhnarovich, P. Viola, and T. Darrell, "Fast pose estimation with parameter-sensitive hashing," in *null*. IEEE, 2003, p. 750.

[50] C. Desai and D. Ramanan, "Detecting actions, poses, and objects with relational phraselets," in *European Conference on Computer Vision*. Springer, 2012, pp. 158–172.

[51] D. Tran, Y. Wang, and D. Forsyth, "Human parsing with a cascade of hierarchical poselet based pruners," in *Multimedia and Expo (ICME), 2014 IEEE International Conference on*. IEEE, 2014, pp. 1–6.

[52] M. Dantone, J. Gall, C. Leistner, and L. Van Gool, "Human pose estimation using body parts dependent joint regressors," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 3041–3048.

[53] M. Kiefel and P. V. Gehler, "Human pose estimation with fields of parts," in *European Conference on Computer Vision*. Springer, 2014, pp. 331–346.

[54] A. Cherian, J. Mairal, K. Alahari, and C. Schmid, "Mixing body-part sequences for human pose estimation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 2353–2360.

[55] W. Ouyang, X. Chu, and X. Wang, "Multi-source deep learning for human pose estimation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 2329–2336.

[56] V. Kazemi, M. Burenius, H. Azizpour, and J. Sullivan, "Multi-view body part recognition with random forests," in *2013 24th British Machine Vision Conference, BMVC 2013; Bristol; United Kingdom; 9 September 2013 through 13 September 2013*. British Machine Vision Association, 2013.

[57] V. Ferrari, M. Marin-Jimenez, and A. Zisserman, "Progressive search space reduction for human pose estimation," in *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*. IEEE, 2008, pp. 1–8.

[58] P. F. Felzenszwalb and D. P. Huttenlocher, "Pictorial structures for object recognition," *International journal of computer vision*, vol. 61, no. 1, pp. 55–79, 2005.

[59] M. Eichner, V. Ferrari, and S. Zurich, "Better appearance models for pictorial structures." in *BMVC*, vol. 2, 2009, p. 5.

[60] L. Sigal. Human pose estimation. [Online]. Available: https://www.cs.ubc.ca/~lsigal/Publications/SigalEncyclopediaCVdraft.pdf

[61] A. Agarwal and B. Triggs, "Recovering 3d human pose from monocular images," *IEEE transactions on pattern analysis and machine intelligence*, vol. 28, no. 1, pp. 44–58, 2006.

[62] D. Ramanan, "Learning to parse images of articulated bodies," in *Advances in neural information processing systems*, 2007, pp. 1129–1136.

[63] L. Bo, C. Sminchisescu, A. Kanaujia, and D. Metaxas, "Fast algorithms for large scale conditional 3d prediction," in *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on.* IEEE, 2008, pp. 1–8.

[64] L. Bo and C. Sminchisescu, "Twin gaussian processes for structured prediction," *International Journal of Computer Vision*, vol. 87, no. 1-2, p. 28, 2010.

[65] M. Andriluka, S. Roth, and B. Schiele, "Pictorial structures revisited: People detection and articulated pose estimation," in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on.* IEEE, 2009, pp. 1014–1021.

[66] Y. Freund and R. E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," *Journal of computer and system sciences*, vol. 55, no. 1, pp. 119–139, 1997.

[67] B. Sapp, A. Toshev, and B. Taskar, "Cascaded models for articulated pose estimation," in *European conference on computer vision.* Springer, 2010, pp. 406–420.

[68] P. F. Felzenszwalb and D. P. Huttenlocher, "Efficient matching of pictorial structures," in *Computer Vision and Pattern Recognition, 2000. Proceedings. IEEE Conference on*, vol. 2. IEEE, 2000, pp. 66–73.

[69] Y. Wang and G. Mori, "Multiple tree models for occlusion and spatial constraints in human pose estimation," in *European Conference on Computer Vision.* Springer, 2008, pp. 710–724.

[70] Y. Wang, D. Tran, and Z. Liao, "Learning hierarchical poselets for human parsing," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2011, pp. 1705–1712.

[71] Y. Tian, C. L. Zitnick, and S. G. Narasimhan, "Exploring the spatial hierarchy of mixture models for human pose estimation," in *European Conference on Computer Vision.* Springer, 2012, pp. 256–269.

[72] H. Jiang and D. R. Martin, “Global pose estimation using non-tree models,” in *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on.* IEEE, 2008, pp. 1–8.

[73] D. Tran and D. Forsyth, “Improved human parsing with a full relational model,” *European Conference on Computer Vision*, pp. 227–240, 2010.

[74] M. Eichner and V. Ferrari, “Appearance sharing for collective human pose estimation,” in *Asian Conference on Computer Vision.* Springer, 2012, pp. 138–151.

[75] L. Bourdev and J. Malik, “Poselets: Body part detectors trained using 3d human pose annotations,” in *Computer Vision, 2009 IEEE 12th International Conference on.* IEEE, 2009, pp. 1365–1372.

[76] L. Pishchulin, M. Andriluka, P. Gehler, and B. Schiele, “Poselet conditioned pictorial structures,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 588–595.

[77] F. Wang and Y. Li, “Beyond physical connections: Tree models in human pose estimation,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 596–603.

[78] A. Jain, J. Tompson, M. Andriluka, G. W. Taylor, and C. Bregler, “Learning human pose estimation features with convolutional networks,” in *Proceedings of the International Conference on Learning Representations*, 2014. [Online]. Available: http://arxiv.org/abs/1312.7302

[79] X. Chen and A. L. Yuille, “Parsing occluded people by flexible compositions,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 3945–3954.

[80] S. Li, Z.-Q. Liu, and A. B. Chan, “Heterogeneous multi-task learning for human pose estimation with deep convolutional neural network,” in *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, 2014, pp. 482–489.

[81] J. J. Tompson, A. Jain, Y. LeCun, and C. Bregler, “Joint training of a convolutional network and a graphical model for human pose estimation,” in *Advances in neural information processing systems*, 2014, pp. 1799–1807.

[82] R. Kindermann, “Markov random fields and their applications,” *American mathematical society*, 1980.

[83] 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*, 2014, pp. 1653–1660.

[84] 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*, 2015, pp. 1347–1355.

[85] J. Carreira, P. Agrawal, K. Fragkiadaki, and J. Malik, “Human pose estimation with iterative error feedback,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 4733–4742.

[86] C.-J. Chou, J.-T. Chien, and H.-T. Chen, “Self adversarial training for human pose estimation,” *arXiv preprint arXiv:1707.02439*, 2017.

[87] V. Belagiannis and A. Zisserman, “Recurrent human pose estimation,” in *2017 12th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2017)*. IEEE, 2017, pp. 468–475.

[88] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, “ImageNet Large Scale Visual Recognition Challenge,” *International Journal of Computer Vision (IJCV)*, vol. 115, no. 3, pp. 211–252, 2015.

[89] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1–9.

[90] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proceedings of the International Conference on Learning Representations*, 2015. [Online]. Available: http://arxiv.org/abs/1409.1556

[91] 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*, 2016, pp. 770–778.

[92] G. Huang, Z. Liu, K. Q. Weinberger, and L. van der Maaten, "Densely connected convolutional networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, vol. 1, no. 2, 2017, p. 3.

[93] A. Bearman and C. Dong, "Human pose estimation and activity classification using convolutional neural networks," *CS231n Course Project Reports*, 2015.

[94] T. Pfister, K. Simonyan, J. Charles, and A. Zisserman, "Deep convolutional neural networks for efficient pose estimation in gesture videos," in *Asian Conference on Computer Vision*. Springer, 2014, pp. 538–552.

[95] I. Lifshitz, E. Fetaya, and S. Ullman, "Human pose estimation using deep consensus voting," in *European Conference on Computer Vision*. Springer, 2016, pp. 246–260.

[96] N. Zhang, E. Shelhamer, Y. Gao, and T. Darrell, "Fine-grained pose prediction, normalization, and recognition," *arXiv preprint arXiv:1511.07063*, 2015.

[97] T. Pfister, J. Charles, and A. Zisserman, "Flowing convnets for human pose estimation in videos," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 1913–1921.

[98] M. Lin, Q. Chen, and S. Yan, "Network in network," *arXiv preprint arXiv:1312.4400*, 2013.

[99] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun, “Overfeat: Integrated recognition, localization and detection using convolutional networks,” *arXiv preprint arXiv:1312.6229*, 2013.

[100] P. Agrawal, R. Girshick, and J. Malik, “Analyzing the performance of multilayer neural networks for object recognition,” in *European conference on computer vision*. Springer, 2014, pp. 329–344.

[101] P. Refaeilzadeh, L. Tang, and H. Liu, “Cross-validation,” in *Encyclopedia of database systems*. Springer, 2009, pp. 532–538.

[102] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, “Caffe: Convolutional architecture for fast feature embedding,” in *Proceedings of the 22nd ACM international conference on Multimedia*. ACM, 2014, pp. 675–678.

[103] M. J. Choi, V. Y. Tan, A. Anandkumar, and A. S. Willsky, “Learning latent tree graphical models,” *Journal of Machine Learning Research*, vol. 12, no. May, pp. 1771–1812, 2011.

[104] Y. Yang and D. Ramanan, “Articulated pose estimation with flexible mixtures-of-parts,” in *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*. IEEE, 2011, pp. 1385–1392.

[105] S. Zheng, S. Jayasumana, B. Romera-Paredes, V. Vineet, Z. Su, D. Du, C. Huang, and P. H. Torr, “Conditional random fields as recurrent neural networks,” in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1529–1537.

[106] A. Krizhevsky, V. Nair, and G. Hinton, “Cifar-10 (canadian institute for advanced research).” [Online]. Available: http://www.cs.toronto.edu/~kriz/cifar.html

[107] V. Nair and G. E. Hinton, “Rectified linear units improve restricted boltzmann machines,” in *Proceedings of the 27th international conference on machine learning (ICML-10)*, 2010, pp. 807–814.

Appendix A

Background on CNNs

A.0.1 How do CNNs work

To understand how CNNs work, let examine a simple linear classifier to classify images in CIFAR-10 dataset [106]. The dataset contains 60000 32x32 color images in 10 classes including bird, cat, dog, ship, etc. Each image x_i is labeled as y_i, indicating the class of the image. The linear classification approach contains two main components: a score function that maps each input image to class scores and a loss function that calculates the differences or losses between class scores and ground-truth labels (Figure A.1).

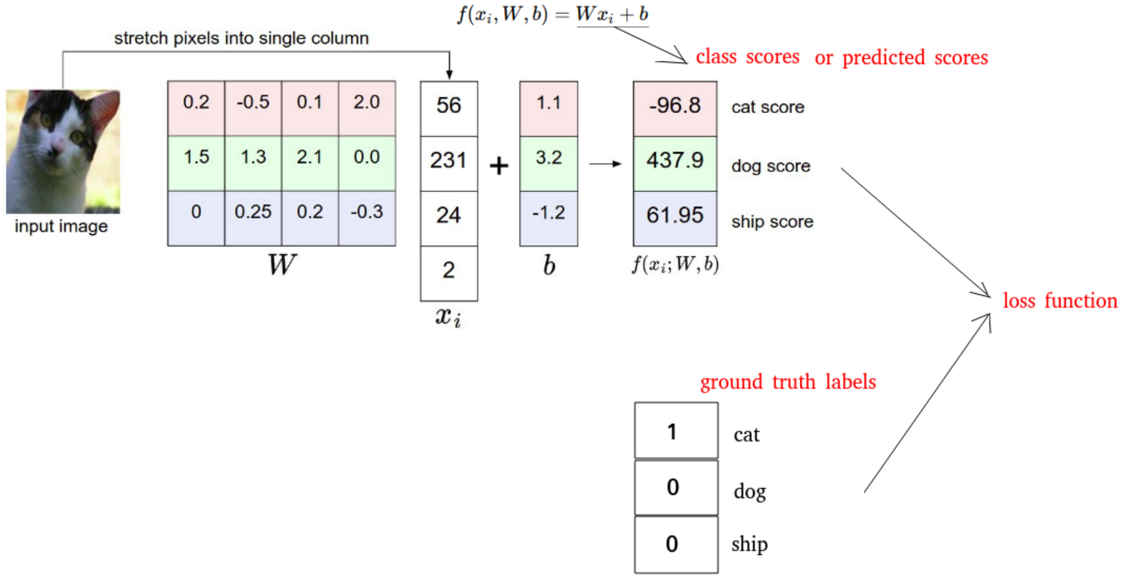


Fig. A.1 The loss and score function.

A score function is described by the formula $f(x_i, W, b) = W * x_i + b$ that maps each input image x_i to class score f where W and b denote weight and bias. Each input image x_i of size 32x32 is flattened into a single vector [3072x1]. The size of weight W and bias b are [10x3072] and [10x1] respectively. The output f contains 10 values equivalent to the scores of the 10 classes. It is noted that the weight matrix W contain 10 rows of [1,3072]. Each row is called a filter and is associated with an output class.

A loss function specifies the differences between class scores f and ground-truth labels y_i. Training a linear classifier model aims to find W and b to obtain a low loss.

Different from the above linear classifier that contains only one linear layer, a neural network consists of several linear layers followed by non-linear functions. CNN is a specific type of neural network designed for grid type inputs. Both CNNs and ordinary neural networks contain neurons, associated with a set of weights and biases. Each neuron in a layer performs a dot product of its associated weights with some neurons in the previous layers, followed by a non-linear function. The output score in the above linear classifier can be considered as a neuron without non-linearity function applied. In convolution layer, each neuron connects to a small set of neurons in the previous layer instead of all as in the fully connected layer. This characteristic of convolution layers dramatically reduces the number of parameters for the whole network compared to fully connected layers.

A.0.2 Layers in CNNs

There are four types of layers in CNNs: convolution layer, pooling layer, fully connected layer and activation layer, in which Rectified Linear Unit (ReLU) [107] was used widely in popular CNN structures such as AlexNet [3], VGGnet [90], ResNet [91] (Figure A.2). Each layer transforms one volume of activations to another. In contrast to convolution layer and fully connected layer, the ReLU and pooling layer do not contain weights.

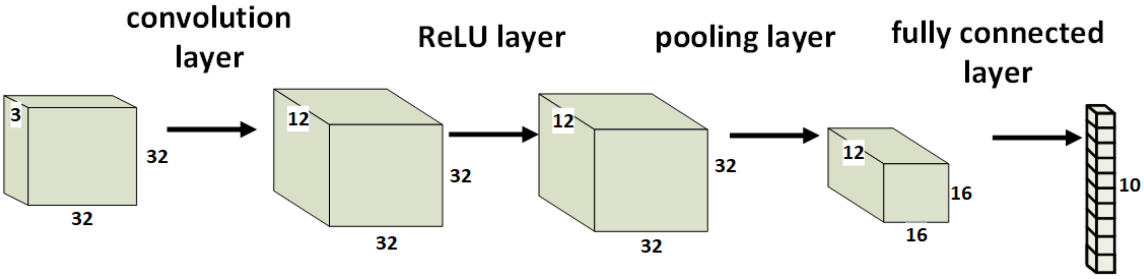


Fig. A.2 A simplified CNN structure.

Convolution Layer

In Figure A.3, the convolution layer transforms an input volume of size 32x32x3 to an output volume of size 32x32x12. The size of output volume is in the format of width, height and depth (or the number of channels). The output volume can be interpreted as the output feature or the 12 channels of 32x32 activation maps. Each value or element in this layer is called a neuron or an activation, obtained by convolving a filter of size 5x5x3 with a window of the same size in the input volume [32x32x3]. This window slides over the input volume and the sliding step is specified by the Stride (S) value. The 12 filters of the convolution layer generate 12 output channels of 32x32 activation maps.

The convolution layer is denoted by its weights [5x5x3x12], in which the receptive field (which is equal to the height or width of the weights, 5), the number of filters (12), and the Stride (S=1) are the three main parameters describing a convolution layer.

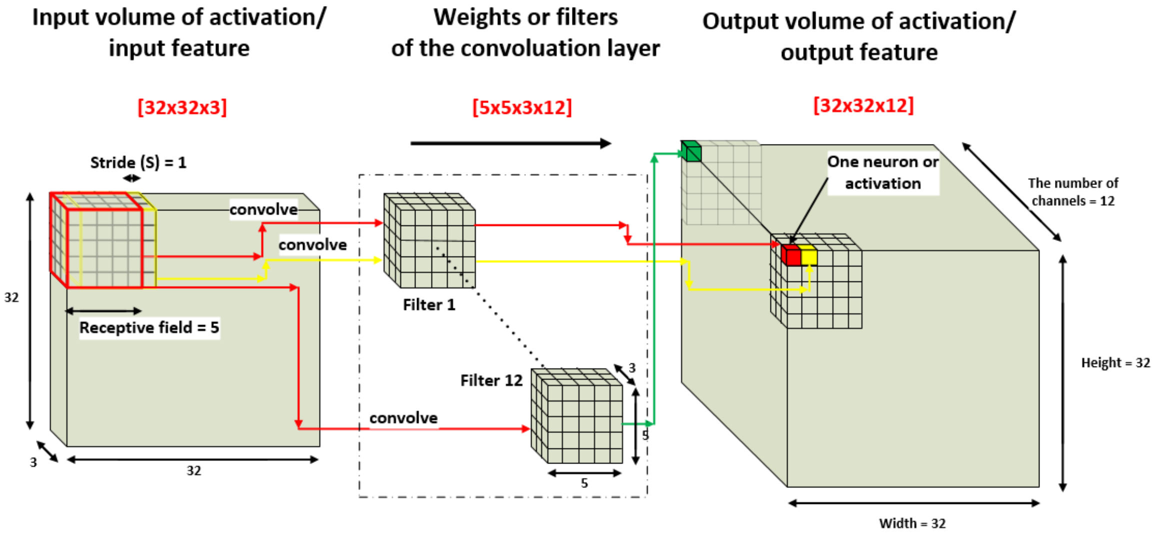


Fig. A.3 Background on convolution layer.

Pooling Layer

In Figure A.4, the pooling layer transform an input volume of size 32x32x12 to an output volume of size 16x16x12 using receptive field (F) of 2 and stride (S) of 2. Each activation of the output volume is obtained by sliding a window of size [FxF] or [2x2] with S = 2 and taking an operation given values inside the window. MAX pooling is a popular pooling operation where each output neuron of the MAX pooling layer is the maximum value of the values in the corresponding FxF window in the input activation map. The pooling layer aims to reduce the amount of parameters and computation in CNNs. Using the max pooling with a receptive field of 2 and a stride of 2, the feature size has reduced from 32x32x12 to 16x16x12. Therefore, it helps reduce the number of parameters required by the following fully connected layer.

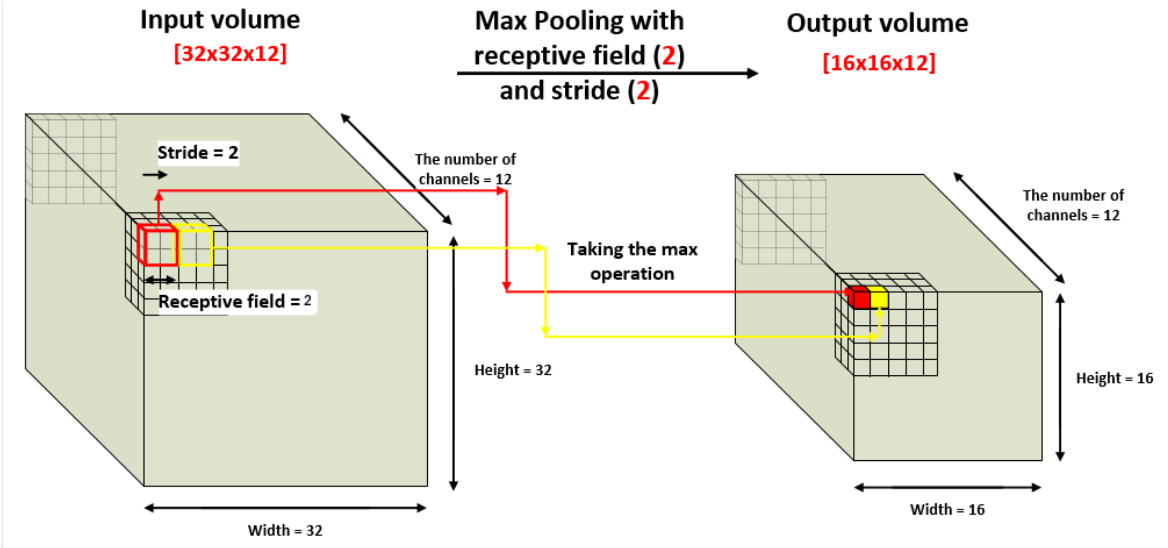


Fig. A.4 Background on pooling layer.

ReLU Layer

The Rectified Linear Unit (RELU) layer performs a non-linear function f(x) = max(0,x) on input layer x. This layer is a type of activation function.

Fully Connected Layer

In Figure A.5, the fully connected layer transforms an input volume of size 16x16x12 to an output volume of size 1x1x10. Each neuron in the output volume is connected to all neurons in the input volume, which is different from convolution layers where each output neuron just connects to a set of input neurons specified by the receptive field parameter.

Fully connected layer requires a large amount of parameters. A fully connected layer with 10 filters for an input volume of 16x16x12 contains 16*16*12*10 = 30720 parameters. However, given an input volume of 16x16x12, a convolution layer with a receptive field of 5, a stride of 1, and 10 filters contains 5*5*12*10 = 3000 parameters which is much less than that of the fully connected layer.

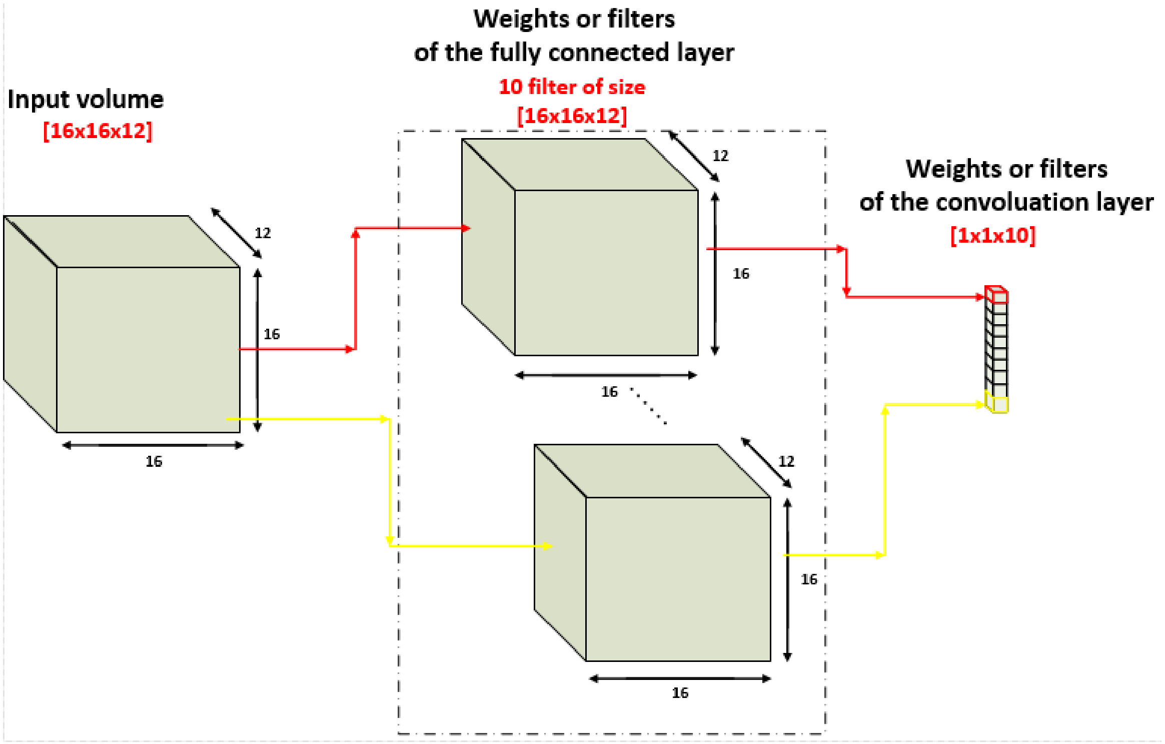


Fig. A.5 Background on fully connected layer.