Efficient Deep Learning methods for Human Pose Estimation

Von der Fakultät für Mathematik, Informatik und Naturwissenschaften der RWTH Aachen University zur Erlangung des akademischen Grades eines Doktors der Naturwissenschaften genehmigte Dissertation

vorgelegt von

MSc
Umer Rafi
aus Islamabad, Pakistan

Berichter: Prof. Dr. Bastian Leibe
Prof. Dr. Juergen Gall

Tag der mündlichen Prüfung: 19.09.2018

Diese Dissertation ist auf den Internetseiten der Hochschulbibliothek online verfügbar.

Abstract

Human pose estimation is a very active research area in the field of computer vision. The goal is to infer the body pose of highly articulated people in images and videos. It has applications in gaming, human computer interaction, analysis, image retrieval, robotics and autonomous driving. In the past few years, impressive success has been achieved on the task of human pose estimation in uncontrolled environments. The success can be mainly attributed to deep learning. However, the trend is to push for maximum performance by using computationally expensive deep learning methods. Despite good performance, the methods require high end graphical processing units for efficient performance. The recent increase in deployment of service robots and autonomous vehicles has raised the demand for human pose estimation methods that run efficiently and robustly on low end graphics processing units that are mostly available on robots and autonomous vehicles.

In this thesis, we propose efficient and robust deep learning methods for human pose estimation. We start with 3D human pose estimation from depth data. We build on a regression forest method to handle partial occlusion from objects without any increase in method's complexity. Our proposed method can be used in indoor scenarios to estimate body poses of people partially occluded by objects. We then move to 2D single person pose estimation from RGB images. We propose an efficient and robust deep learning method for 2D single person pose estimation. The proposed method achieves comparable performance to the recent state-of-the-art 2D single person pose estimation methods. Our proposed method runs efficiently on low-end graphics processing units. We then approach the task of multi-person pose estimation. We propose a bottom up multi-person method with a novel pairwise relative offsets based association. Our approach simultaneously predicts body parts and relative offsets maps respectively. The relative offsets are used to associate body joints with a greedy procedure. Compared to state-of-the-art bottom up approaches the proposed method does not require any expensive post-processing. Finally, we propose a correspondence matching method with an efficient

correlation layer. The correlation layer efficiently computes similarity heat-maps for all the query key points over the target image. The method is applied to the tasks of semantic key points matching, dense matching and instance key points matching. In the context of multi-person pose tracking, we apply instance key points matching for associating body joints over time.

Zusammenfassung

Die Schätzung der menschlichen Pose ist ein aktiver Forschungsbereich auf dem Gebiet der Computervision. Ziel ist es, in Bildern und Videos die Körperhaltung artikulierender Personen herzuleiten. Dies findet Anwendungen in Spielen, Mensch-Computer-Interaktion, Analyse, Bildsuche, Robotik und autonomes Fahren. In den letzten Jahren wurde ein beeindruckender Erfolg bei der Aufgabe der Schätzung von menschlichen Posen in unkontrollierten Umgebungen erreicht. Der Erfolg kann hauptschlich dem Deep-Learning zugeschrieben werden. Der Trend geht jedoch dahin, mit rechenintensiven Methoden auf maximale Leistung zu setzen. Trotz guter Leistung erfordern die Verfahren High-End-Grafikverarbeitungseinheiten fr eine effiziente Leistung. Die jüngste Zunahme von Servicerobotern und autonomen Fahrzeugen haben die Nachfrage nach Methoden zur Schätzung der menschlichen Pose erhht, die effizient und robust auf Low-End-Grafikverarbeitungseinheiten laufen, welche hauptsächlich fr Roboter und autonome Fahrzeuge verfügbar sind.

In dieser Arbeit schlagen wir effiziente und robuste Deep-Learning Methoden zur Schätzung der menschlichen Pose vor. Wir beginnen mit einer 3D-Methode zur Schätzung der menschlichen Pose anhand von Tiefendaten. Wir bauen auf einer Regressions-Forest-Methode auf, um partielle Verdeckungen von Objekten zu behandeln, ohne die Komplexität der Methode zu erhöhen. Unser vorgeschlagenes 3D-Schätzverfahren für menschliche Posen kann in Innenraum-Szenarien verwendet werden, um Körperposen von Personen abzuschätzen, die teilweise durch Objekte verdeckt sind. Wir gehen dann zur 2D-Posenschätzung von Einzelpersonen in RGB-Bildern ber. Wir schlagen eine effiziente und robuste Deep-Learning Methode für die 2D-Einzelposenschätzung vor. Das vorgeschlagene Verfahren erreicht eine vergleichbare Leistung wie die neuesten 2D-Einzelpersonen-Posenschätzmethoden. Unsere vorgeschlagene Methode läuft effizient auf Low-End-Grafikverarbeitungssystemen.

Wir widmen uns dann der Aufgabe der Mehrpersonen-Posenschätzung. Wir schlagen eine Bottom-Up-Mehrpersonen Methode vor mit einer neuartigen Assoziation, basierend

auf paarweisen relativen Offsets. Unser Ansatz sagt gleichzeitig Körperteile und relative Offsets voraus. Die relativen Offsets werden verwendet, um Körpergelenke mit einer greedy-Prozedur zu assoziieren. Im Vergleich zu modernen Bottom-up-Ansätzen erfordert das vorgeschlagene Verfahren keine kostspielige Nachbearbeitung. Schlielich schlagen wir eine effiziente und robuste Korrespondenz-Matching-Methode mit einer neuen Korrelationsebene vor. Die Korrelationsebene berechnet effizient Ähnlichkeits-Heat-Maps für alle Abfrage-Schlüsselpunkte über dem Zielbild. Die Methode wird auf den Vergleich von semantischen Schlüsselpunkten, dichte Anpassung und übereinstimmende Instanzschlüsselpunkte angewendet. Im Kontext von Mehrpersonen-Posentracking, kann der Vergleich von Instanz-Schlüsselpunkte verwendet werden, um Körpergelenke über die Zeit zu assoziieren.

Contents

1 Introduction

1.1 Motivation

Human Pose Estimation has been a long standing research area in the field of computer vision. The goal is to infer the body pose of highly articulated people in images and videos. Due to large potential applications, the problem has been actively researched. The applications can be roughly categorized into surveillance, control, analysis, robotics and autonomous driving. Surveillance applications are concerned with monitoring people in crowded areas such as airports, malls and subways. Applications may include detecting persons with abnormal behaviour, person identification etc. Body pose can be used as an important cue in these applications. Control applications include computer games and human computer interaction. In computer games, body pose is used to animate 3D game characters. In human computer interaction, body pose can be used for touch less interaction. Analysis applications include analysing athletes behaviour in various sports. Another application is body pose based retrieval of images from an image database or videos. In robotics, human pose estimation can be used to carry out user defined tasks. For e.g. a service robot in an elderly care scenario can use body pose to detect fallen elderly people and generate alarm for immediate assistance. On airports, a service robot can detect spokes person in a group of people by using body pose as an important cue. In autonomous driving, body pose can be used to anticipate behaviour of pedestrians, to detect sleeping drivers and for automatic control of air bags during accidents etc..

In the past few years, the success of deep learning in computer vision has led people to work on human pose estimation in uncontrolled environments. However, the trend is to push the state-of-the art performance for human pose estimation by using computationally expensive deep learning methods with complex post-processing on the top. Despite impressive performance, the proposed methods require high end graphics processing units for real time performance. The recent increase in deployment of service robots and autonomous vehicles has raised the demand for efficient and robust human pose

estimation methods. The methods should run in real time on robotics and autonomous vehicles where low end graphics processing systems are mostly available.

In this thesis, we propose efficient and robust methods for different variants of human pose estimation. Starting from 3D human pose estimation from depth data, we extend a existing regression forest method to handle partial occlusion from objects without any increase in method's complexity. Our proposed method can be used in indoor scenarios to estimate body poses of people partially occluded by objects. We then move to 2D single person pose estimation from RGB images. We propose an efficient deep learning method for 2D single person pose estimation. The proposed method achieves comparable performance to recent state-of-the-art 2D single person pose estimation systems. Our proposed method runs efficiently on low-end graphics processing systems. We then approach the task of multi-person pose estimation. We propose a bottom up multi-person method with a novel pairwise relative offsets based association. Our method simultaneously predicts body parts and relative offsets maps respectively. The relative offsets are used to associate body joints with a greedy procedure. Compared to state-of-the-art bottom up approaches the proposed method does not require any expensive post-processing. Finally, we propose a simple and robust correspondence matching method with an efficient correlation layer. The correlation layer efficiently computes similarity heat-maps for all the query key points over the target image. The method is applied to the tasks of semantic key points matching, dense matching and instance key points matching. In the context of multi-person pose tracking, we use instance key points matching for associating body joints over time.

1.2 Contributions

In this thesis we make the following contributions :

- We build on a regression forest method for 3D human pose estimation to handle partial occlusion by objects from depth data. The proposed method exploits the context of occluding objects as an additional cue to handle occluded body joints. We also propose a Pose Occlusion test set for evaluating approaches for 3D human pose estimation under partial occlusion by objects.

- We propose an efficient deep learning method for 2D single person pose estimation from RGB images. The method runs efficiently on low end graphics processing units. Despite the low computational requirements, it is on par with recent state-of-the-art 2D single person pose estimation methods.

- We propose a bottom up multi-person pose estimation method with a novel pairwise associative voting. Our method simultaneously predicts body parts and relative offsets maps respectively. The relative offsets are used in a greedy manner to associate different body joints. Our method achieves comparable performance to recent state-of-the-art on the MS COCO dataset.

- We propose a correspondence matching method with an efficient correlation layer. The correlation layer efficiently computes similarity heat-maps for all the query points over the target image. The method is applied to the tasks of semantic key points matching, dense matching and instance key points matching.

1.3 Structure of the Thesis

The thesis is structured as follows :

Chapter 2 is concerned with related work on human pose estimation. The purpose is to provide reader an overview on the progress of the state-of-the-art for different variants of the human pose estimation approached in this thesis.

Chapter 3 is concerned with fundamentals of human pose estimation. We start with an introduction to the basic human pose estimation problem. We then introduce the variants approached in this thesis. We also provide an overview of the popular human pose estimation datasets and the different evaluation measures used. We provide a brief introduction to aspects of convolutional neural networks relevant to this thesis.

Chapter 4 is concerned with 3D human pose estimation from depth data. We focus on handling partial occlusions from objects. We build on a regression forest method for 3D human pose estimation. Instead of discarding the occluders, the proposed method exploits the rich context of occluding objects as an additional cue to predict the positions of occluded joints. The proposed method outperforms the commercially available Kinect SDK for occluded joints. We also propose the Pose Occlusion test set for evaluating 3D human pose estimation under partial occlusion by objects.

Chapter 5 is concerned with 2D single person pose estimation from RGB images. We propose an efficient fully convolutional neural network architecture for 2D single person pose estimation. The proposed architecture runs efficiently on a mid-range graphics processing unit and therefore could be deployed in robotics scenarios for real time performance. Despite of low computational budget, the proposed architecture is on par with recent state-of-the-art architectures for 2D single person pose estimation.

Chapter 6 is concerned with multi-person pose estimation. We propose a bottom up multi-person pose estimation method with a novel pair-wise associative voting. Our approach simultaneously predicts body parts and relative offsets maps respectively. The relative offsets are used in a greedy manner to associate different body joints. Our approach achieves comparable performance to the recent state-of-the-art on the MS COCO dataset.

Chapter 7 is concerned with correspondence matching. We propose a correspondence matching method with an efficient correlation layer that efficiently computes similarity

heat-maps for all the query key points over the target image. The method is applied to the tasks of semantic key points matching, dense matching and instance key points matching. In the context of multi-person pose tracking, we use instance key points matching to associate different body joints of the same person over time in video sequence.

Chapter 8 concludes the thesis and provides future directions.

Note: *The thesis is based on the technical contributions of my respective first author publications (Rafi* et al. *(2015), Rafi* et al. *(2016)). Several images and text passages in the three major parts of this thesis are taken from these articles. However, additional and new content about further extensions has been added in order to provide deeper insights into our approaches.*

2

State of the Art

Human pose estimation is a challenging problem in the field of computer vision. It has been actively studied and has generated a large body of work. The problem has been approached with different lines of research. One line of research has focussed on applying generative models to the task. A variety of generative models have been explored in the literature. Other line of research has explored various discriminative models. Another line of research has focussed on applying both discriminative and generative models.

In this chapter, we overview related work for different variants of human pose estimation that are approached in this thesis. The purpose of this chapter is to provide reader an overview on state-of-the-art progress over time.

This chapter is organised as follows. In Section 2.1 we describe the related work for 3D human pose estimation from depth data. We describe state-of-the-art for 2D single person pose estimation in Section 2.2. In Section 2.3 we describe the related work on 2D multi-person pose estimation. Finally, in Section 2.4 we discuss the related work on 2D multi-person pose tracking and correspondence matching.

2.1 3D Human Pose Estimation.

3D human pose estimation has generated large body of related work. The section provides an overview on the state-of-the-art progress for 3D human pose estimation.

2.1.1 3D Human Pose Estimation from RGB images

Agarwal and Triggs (2006) trained discriminative regressors to predict the holistic 3D pose directly. The regressors performs sub-optimally due to high dimensional pose space and limited training data. Bo *et al.* (2008) proposed conditional mixture of experts for 3D human pose estimation. Each expert in the mixture is trained on subset of pose space. During inference, the output of each expert is weighted. The weights are conditioned on image data. Kostrikov and Gall (2014) proposed to predict each body joint separately along with the depth.

The above mentioned approaches relied on hand crafted features. Recently convolutional neural networks (CNNs) have been successfully used for feature learning in

computer vision and have achieved tremendous success. Li and Chen (2014) used CNN to directly predict the 3D pose from images. Chen and Remanan (2017) proposed to use 2D pose as intermediate representation for 3D pose estimation. They first predict the 2D pose for the image using a CNN. They then use the 2D pose to minimise the project error of the unknown 3D pose and unknown camera.

The main caveat in using RGB images for predicting the 3D human pose is the depth ambiguity.

2.1.2 3D Human Pose Estimation from depth images

In recent years, the availability of affordable depth sensors have spurred the progress in 3D human pose estimation from depth images. The work on 3D human pose estimation can be categorised into generative approaches and discriminative approaches.

Generative Approaches. The main idea in generative approaches is to fit a 3D model to the point cloud representation computed from the depth image. Different approaches vary in complexity of the 3D model used. Grest *et al.* (2005) fit a fixed mesh model to depth data to estimate the 3D pose. Ganapathi *et al.* (2012) used a cylindrical model and ray tracing to infer the 3D pose. Despite good performance, the generative approaches require good priors to initialise the state of the 3D model.

Discriminative Approaches. Discriminative approaches try to predict the 3D pose directly from image data and thus do not require any model initializations. Shotton *et al.* (2011) treat the 3D human pose estimation as a classification problem. They classify every pixel in the image to one of the body parts. Mean shift clustering is applied on the top to infer 3D joint positions. The approach works in real time and does not suffer from the the high pose space problem since prediction of each body joint is done separately. However, the approach cannot handle self occlusions.

Girshick *et al.* (2011a) used hough voting to infer the 3D human pose from the image. Each pixel in the image votes for the 3D position of each body joint along with a confidence. Mean shift clustering is applied to the voted joint positions to get final joints positions. Hough voting solves the problem of self occlusions.

The approaches (Shotton *et al.* (2011), Girshick *et al.* (2011a)) assume no limb length constraints on the joint predictions. Sun *et al.* (2012) condition the joint predictions on the person height/orientation in the image to enforce implicit limb length constraints on the joints. Taylor *et al.* (2012) use a discriminative regressor to get an initial 3D human pose. They use this 3D pose to initialise a 3D mesh model. The model is then fitted to the image data to get a final 3D pose. The model fitting procedure enforces limb length constraints explicitly.

Occlusion Handling. Occlusion is a natural phenomenon is images. Objects overlap each other and cause partial occlusions. Detecting and handling partial occlusions is an interesting problem in computer vision. Different approaches have been proposed to handle partial occlusions. Girshick *et al.* (2011b) used different occlusion templates to detect partially occluded people. The occlusion templates are learned from data.

Occlusion handling has also been explored in the context of facial landmark localization. Yu *et al.* (2014) propose to use a mixture of regressors to handle occluded faces. Each regressor is trained to handle occlusion for a particular region of face. During inference, the outputs of regressors are merged using a bayesian averaging procedure.

The work presented in chapter 4 deals with 3D human pose estimation from depth data. We foucs on handling partial occlusion by objects. The work is inspired from (Girshick *et al.* (2011a), Girshick *et al.* (2011b)). However the work from Girshick *et al.* (2011a) cannot handle occlusion by objects. In chapter 4, we extend the approach from Girshick *et al.* (2011a) to explicitly handle partial occlusion by objects. The work from Girshick *et al.* (2011b) learns patch level templates to handle occlusions. In contrast, the work in chapter 4 learns to handle occlusions at pixel levels.

2.2 2D Single Person Pose Estimation.

2D Human Pose Estimation has been well-explored in the computer vision literature. The research on 2D human pose estimation can be categorised into CNN and classical approaches.

Classical Approaches. Classical approaches on 2D Human Pose Estimation use the pictorial structures representation (Felzenswalb and Huttenlocher (2005)). The human body is divided into anatomical body parts. A tree-structured graphical model is used to encode spatial dependencies between neighbouring body parts.

Over the years, several extensions have been proposed to the pictorial structures model. Wang and Mori (2008) proposed a mixture of trees model to capture richer relationships between non-connected body parts in single tree model. During inference, the outputs of tree models are merged according to leaned weights. Johnson and Everingham (2010) rather proposed a mixture model at the body parts level. Their proposed model consisted of learned multiple templates for each body part. A a single tree structure graphical model is used on the top. Yang and Ramanan (2011) divided each body part into smaller body parts. The placement of smaller body parts is modelled using latent variables. In addition to that, they also proposed to model part types based on their orientation in the image. Sapp and Taskar (2013) proposed to learn image conditioned pair-wise templates for different body parts in a single tree model. During inference, a pair-wise template is selected according to image data.

Another line of research has focussed on modelling higher order relationships between different body parts. Inference in higher order models is intractable. Several approximation have been proposed. Pishchulin *et al.* (2013a) jointly detect body parts and a holistic pose from the image. The holistic pose is then used to model spatial dependencies between the detected body parts. Dantone *et al.* (2013) used a sequence of body part regressors to approximate higher order relationships between body parts. The output of each regressor is provided to the next regressor to implicitly enforce spatial dependencies. Sequential prediction machines are also proposed in Ramakrishna *et al.* (2014) to model complex dependencies with tractable inference.

Complex non-tree models (Sigal and Black (2006)) have also been used that model spatial dependencies between unconnected body parts. Loopy belief propagation is then used for approximate inference to predict the joint positions in the image. (Sun and Savarese (2011); Tian *et al.* (2012)) proposed hierarchical models. They idea is to first detect larger parts in the image. Then condition the smaller parts detection on the the detected larger parts.

CNN base Approaches. All the classical approaches relied on hand-crafted features. Recently, convolutional neural networks (CNNS) have been successfully applied to many computer vision tasks. (Toshev and Szegedy (2014)) first used CNNs to directly regress the positions of body joints. Their approach used a cascade of regressors. The first regressor predicted a holistic pose. The holistic pose was then refined by the next regressors in the cascade. Tompson *et al.* (2014) found that predicting the holistic pose is sub-optimal since the holistic space is high dimensional. They rather proposed to predict the position of each body part independently. They also proposed to predict a belief map for each body part. The belief map contains a per-pixel likelihood for each body part. Their approach also incorporated a simple graphical model on the top of predicted belief maps. The whole system was trained end-to-end. The body parts prediction suffered from the max-pooling used in the CNNs. Tompson *et al.* (2015), in their later work, compensated the negative effect of pooling by introducing a refinement module. Carreira *et al.* (2016) proposed to predict body parts positions in an iterative manner.

Wei *et al.* (2016) proposed convolutional pose machines. The idea is to use a stack of CNNs. Each CNN in the stack refines the prediction by the previous CNN. The stack based model makes the graphical model obsolete. Bulat and Tzimiropoulos (2016) proposed a stack of belief map predictors and regressors. Newell *et al.* (2016) proposed a stack of hourglass CNNs. The low resolution features are stage-wise up-sampled to high resolution belief maps for different body joints. Chu *et al.* (2017) extended the staked hourglass model with muilti-resolution attention. Yang *et al.* (2017) proposed a pyramid CNN where pyramids of features are learned for detecting each body part.

The current trend is to use expensive CNN architectures to achieve maximum performance. The expensive CNN architectures require high-end graphical processing units (GPUS) to run. This makes the approaches unrealistic for many real time computer vision applications for robotics where high-end GPUs are not available. In addition to that, stacked CNN architectures are slower to train.

In the context of this thesis, the work presented in chapter 5 proposes an efficient architecture for 2D single person pose estimation. The architecture runs efficiently on low-end GPUs and is faster to train with a simple and transparent training procedure. Despite low computational budget, the proposed architecture is on par with the recent CNN based approaches for 2D human pose estimation on the popular single person pose estimation benchmarks.

2.3 2D Multi-Person Pose Estimation.

In the past few years, tremendous progress has been made in 2D single person pose estimation from RGB images due to deep learning. This has led researcher in the computer vision community to focus on the more challenging problem of 2D multi-person pose estimation in the wild. The problem also involves associating detected body parts to persons in the image.

The current research on multi-person pose estimation can be categorised into top down and bottom up methods respectively. The bottom up methods first detect all the body parts in the image. Graph based methods are applied as a post-processing step for associating body parts to persons. The top down methods first detect bounding boxes around all the persons in the image. Single person pose estimation is applied to each bounding box to estimate body pose.

Bottom Up Multi-person Pose Estimation. Pishchulin *et al.* (2016) first proposed a bottom up approach for multi-person pose estimation. Their method first proposed a number of proposals for each body part in the image. A densely connected graph is constructed on the top of proposals. Linear programming optimization is applied to the graph to assign body parts proposals to persons. Despite suitable performance, the approach takes 72 hours to perform inference on a single image. Insafutdinov *et al.* (2016) improved the work from Pishchulin *et al.* (2016). They used expensive CNN architecture to improve detection of body parts. In addition to that, they used an incremental optimization procedure to speed up the inference procedure to 8 minutes. Iqbal and Gall (2016) further speeded up the inference by solving the inference locally for each person.

Cao *et al.* (2017) jointly predicted the body parts detection and pair-wise affinity heat maps. They used a greedy approach on the top for body-parts associations. They construct a graph that assumes a tree structure on the body parts connection and is faster to optimize. Their approach reduced the per-image inference time to 0.005s.

All the bottom up methods for multi-person pose estimation require graph based post-processing for associating body parts to persons. Newell *et al.* (2017) have recently proposed associative embeddings to solve the association problem. They proposed to jointly predict the body parts detection and tag maps. The tag maps consists of dense 1D tags for every pixel in the image. The idea is to predict similar tags for body parts belonging to the same person. During inference, mean shift clustering is applied to predicted tags to associate body parts to the person.

Top Down Multi-person Pose Estimation. Papandreou *et al.* (2017b) proposed a two stage pipeline for multi-person pose estimation. They first use state-of-the-art people detector Ren *et al.* (2015) to generate proposals for people detection in the image. They perform non-maximal suppression to get rid of noisy proposals. The final set of proposals are passed to a single person pose estimator. The approach also predicts relative offsets maps for each body part. The relative offsets maps densely predict relative offsets for every pixel to every body part belonging to the person.

He *et al.* (2017) extended the work from Ren *et al.* (2015) to predict body parts detection for every detected bounding box. In comparison to Papandreou *et al.* (2017b), their approach reuses the existing features for body parts predictions.

Chen *et al.* (2018) proposed a novel single person pose estimator employed in top down approaches. The proposed estimator consists of a global net and a refinement net. The global net predicts body parts detection at every stage of the CNN. The refinement net produces refined detections on the concatenated features from different stages of the CNN.

In chapter 6, we propose a bottom up multi-person pose estimation method with a novel pair-wise associative voting. Our approach simultaneously predicts body parts and relative offsets maps respectively. The relative offsets are used in a greedy manner to associate different body joints. Our approach achieves comparable performance to the recent state-of-the-art on the MS COCO dataset.

2.4 2D Multi-Person Pose Tracking and Correspondence Matching.

2D Multi-person pose tracking. 2D multi-person pose tracking from videos is an extremely challenging problem. The task involves detecting and associating body parts of a person over time. In recent years, researchers have proposed new datasets and methods for multi-person pose tracking.

Iqbal and Gall (2017) have recently proposed a new dataset for Multiperson pose tracking. The have also proposed a base-line formulation for joint multi-person pose estimation and tracking. Their approach first detects body-parts proposals for all the frames in a video clip. Then a spatio-temporal graph is constructed on the top of body-parts proposals. The weights for spatial edges in the graph are computed using a logistic regression classifier. The weights for temporal edges are computed using optical flow. Linear programming optimization is then applied for body parts association in and across frames. A similar idea has been proposed in Insafutdinov *et al.* (2017).

Girdhar *et al.* (2017) have recently extended the work from He *et al.* (2017) for multi-person pose estimation from videos. They replaced the 2D CNN back-end with a 3D CNN back-end for feature generation. The 3D CNN back-end slowly fuses the spatial and temporal information from frames in the clip. Their approach generates tubelets proposals. Each tublet contains people detections over frames. Each tublet is processed by pose head to generate poses of people over time. For pose association, they use similarity of existing features leaned by the 3D CNN.

Correspondence Matching. Correspondence Matching is a well-explored problem in computer vision. Classical approaches used hand-crafted features such as SIFT (Lowe (2004)), SURF (Bay *et al.* (2008)), or DAISY (Tola *et al.* (2010)).

Recently CNNs based Siamese Networks have been used for correspondence matching tasks. (Zagoruyko and Komodakis (2015)) proposed a Siamese network is used to measure patch similarity. Other Siamese networks have been used in (Schroff *et al.* (2015))

to learn face embeddings for super-human face identification performance. Zbontar and LeCun (2015) used them for stereo matching. Long *et al.* (2014) analyzed a CNN pre-trained on ImageNet for the task of semantic correspondence search. In (Wang *et al.* (2014)) a CNN is trained with triplet loss for fine-grained image ranking. Luo *et al.* (2016) trained a Siamese network with a dot product layer and multi-class classification loss for efficient disparity estimation.

Recently approaches that perform image-image semantic keypoints matching have been proposed . Choy *et al.* (2016) introduced a "Universal Correspondence Network" with spatial transformer layers (Jaderberg *et al.* (2015)). Their network is trained efficiently using metric learning and requires a single forward pass at inference time for matching multiple key points. However, due to metric learning it requires an extra hard negative mining step that introduces the extra distance measure and k-nearest neighbor hyper-parameters. In contrast, we present an end-end learning framework that does not require the extra hard negative mining step. Kim *et al.* (2017) proposed a fully convolutional self-simalrity descriptor for dense semantic keypoints matching. However, their approach still requires a matching framework e.g (Ham *et al.* (2016)) on the top to establish correspondences. Han *et al.* (2017) recently proposed a system that matches region proposals across a pair of images using appearances and geometry.

In chapter 7 of this thesis, we propose a correspondence matching framework with an efficient correlation layer. The correlation layer efficiently generates peaked similarity heat-maps for query points. The framework is applied to the task of instance key points matching, dense matching and instance key points matching. In the context of multi-person pose tracking, instance key points matching is used to associate joints detection over time.

3 Fundamentals

This chapter is concerned with the fundamentals of human pose estimation and its variants that are approached in later chapters. We introduce the problem of human pose estimation in Section 3.1. In Section 3.2 we introduce different variants of the human pose estimation. We overview popular human pose estimation benchmarks in Section 3.3. In Section 3.4, we describe the different evaluation measures used for human pose estimation. Finally, in Section 3.5 we briefly introduce fundamentals of Convolutional Neural Networks (CNNs). We focus only on part of the CNNs that is relevant to this thesis.

3.1 Human Pose Estimation

In this section, we introduce the human pose estimation problem and the various challenges involved in it. The human pose estimation problem in the basic setting can be defined as :

Definition. Given an image containing a single person, the task is to infer the (x, y) positions of the different body parts of the person in the image.

The problem is illustrated in Figure 3.1. The body parts are shown in blue circles. In this basic setting, the problem is still challenging due to various nuisance factors described below.

Shapes and Sizes. Human come in various shapes and sizes. The shapes range from very thin people to very thick people. Similarly sizes vary from short to tall people. In addition to that, scale of the people is not available in RGB images.

Appearance and illumination changes. People come in different clothings. Appearance of people also vary with illumination changes.

Figure 3.1: Basic human pose estimation problem. The goal is to infer (x, y) positions of different body parts as shown in blue circles.

Figure 3.2: Different nuisance factors in human pose estimation. (Row 1) Appearance variations. (Row 2) Size and shape variations across people. (Row 3) Pose and activity variations. (Row 4) Self occlusions.

Variation in Activities and Pose. People appear in different activities, e.g., cooking, running and walking etc.. Poses of people vary in each activity.

Self Occlusions. In many cases, some body parts of people are self occluded and are harder to localize. Examples of factors are shown in Figure 3.2.

3.2 Variants of Human Pose Estimation

Given the introduction in Section 3.1, we describe different variants of the problem that are approached in the later chapters in this thesis.

3D Human Pose Estimation from depth data. In 3D human pose estimation from depth data the goal is to infer the (x, y, z) positions of different body parts. It is assumed that the person that has already been segmented. An example is shown in Figure 3.3 (left). The depth information makes person segmentation easier. The depth based representations are invariant to lightening and appearance changes. However, the problem is still challenging due to shape and pose variations.

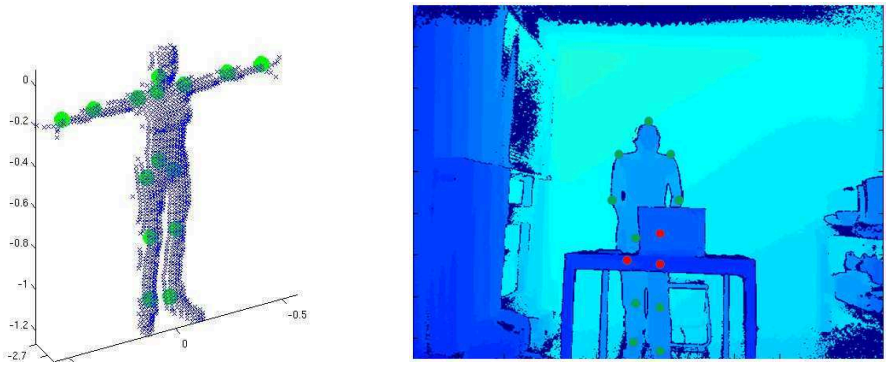


Figure 3.3: (left) : 3D human pose estimation. The goal is to infer (x, y, z) positions of different body parts as shown in green circles. (right) 3D human pose estimation under partial occlusion by objects. The visible body parts are shown in green circles and the occluded body parts are shown in red circles respectively.

In this thesis, we focus on estimating 3D poses of people partially occluded by objects. An example in shown in Figure 3.3 (right). The person's left hand and hips are partially occluded by the laptop and the table respectively.

2D Single person Pose Estimation. 2D single person pose estimation has already been introduced in Section 3.1. Person segmentation is difficult since the depth information is not available. It is assumed that a bounding box is available around the person.

2D Multi-person pose estimation. In 2D multi-person pose estimation the goal is to infer and and associate body parts for every person in the image. No prior information is provided about the locations and number of people present in the image. Persons may overlap each other or may be truncated. This makes the problem challenging. Examples are shown in Figure 3.4.

2D Multi-person pose tracking. 2D Multi-person pose tracking, from videos, is an extremely challenging problem. The problem also requires people association over the video. The scale and appearance of people change over time. People may overlap each

Figure 3.4: 2D Multi-person pose estimation. The goal is to infer and associate body parts for every person in the image. In the right image, the persons overlap and are truncated.

other or may be truncated. This makes people association harder. An example is shown for two different frames from a video sequence in Figure 3.5.

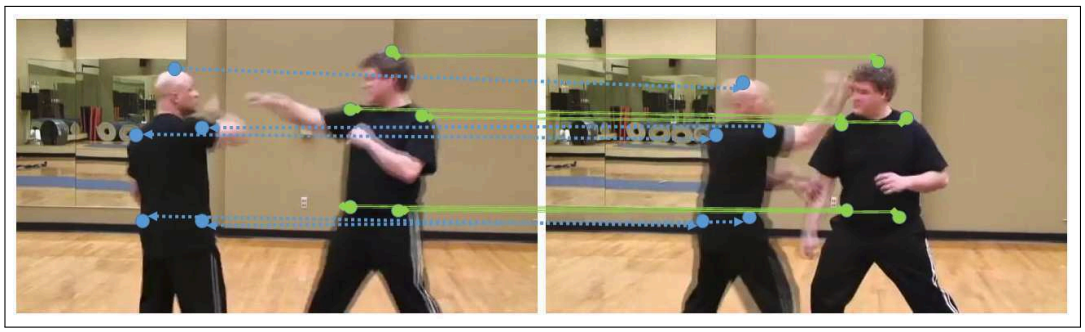


Figure 3.5: 2D Multi-person pose tracking. The goal is to infer and associate body parts of every person in and across frames in a video sequence.

3.3 Human Pose Estimation Datasets

In recent years, various datasets have been introduced for different variants of the human pose estimation problem. In this section we will describe popular datasets for evaluating different human pose estimation approaches.

3.3.1 Pose Occlusion Dataset

In this thesis, we introduce the Pose Occlusion test dataset (Rafi *et al.* (2015)). The dataset consists of 1000 depth images of partially occluded people. The dataset is recorded in offices and living rooms. The dataset consists of seven different subjects in various sitting and standing poses recorded from different view points. Each image in the dataset consists of a single person partially occluded by objects like table, laptop and monitor. Each image is annotated with 2D ground-truth positions of 15 body joints and their visibility. Noisy Foreground masks for each image is also provided. Example images and foreground masks from the dataset are shown in Figure 3.6.

Figure 3.6: Example depth images from the Pose-Occlusion test set (Rafi *et al.* (2015)) with ground-truth skeleton annotations. Row 3 shows noisy foreground masks.

3.3.2 Frames Labeled in Cinema

Frames Labeled in Cinema (FLIC) dataset (Sapp and Taskar (2013)) consists of 5003 images extracted from 30 Hollywood Movies. A people detector was run on every 10th frame in the movies. Images with high people detection scores were selected. The selected images were annotated with skeletons by using Amazon Mechanical Turk. Many Images in the dataset contain multiple persons. In order to use the dataset for single person pose estimation a bounding box is provided for every person in the image. The dataset has 3987 training and 1016 test images respectively. The skeleton annotations consists of 7 upper body parts. Example images from the dataset are shown in 3.7.

3.3.3 Leeds Sports and Extended Leeds Sports Datasets

The Leeds sports dataset (LSP) (Johnson and Everingham (2010, 2011)) consists of 2000 Images. The images were collected from Flickr. The images consists of people performing different sport activities. Most of the images contain single person only. The dataset has 1000 training and 1000 test images respectively.

The Extended Leeds sports (ELSP) (Johnson and Everingham (2010, 2011)) dataset provides 10000 more training images. The images were collected from Flickr. The images consists of person performing aerobics, gymnastics and parkour.

Figure 3.7: Example images with ground-truth skeleton annotations from the Frames Labeled in Cinema (FLIC) dataset (Sapp and Taskar (2013)).

Both LSP and ELSP datasets have skeleton annotations consisting of 14 different body parts. All the images in LSP and ELSP datasets have been scaled so that average person height is rougly 150 pixels. Example images are shown in Figure 3.8.

3.3.4 MPII Human Pose Dataset

MPII human pose dataset (Andriluka *et al.* (2014)) is a challenging dataset. The dataset consists of 25k images. The total number of persons in the dataset are roughly 40k. Comapred to FLIC, LSP and ELSP datasets, the MPII human pose dataset contains people performing various every day activities. The images in the dataset were collected from YouTube videos. The dataset contains skeleton annotations of 16 body parts. Scale and bounding box are provided for every person in the dataset. The dataset has 19185 training images and 7246 test images respectively. The ground-truth annotations for the test set are held out. Examples are shown in Figure 3.9.

3.3.5 MPII Multiperson Dataset

The MPII multi-person dataset (Pishchulin *et al.* (2016)) consists of 3844 training and 1758 testing images. The images were collected from YouTube videos. Each image contains groups of multiple persons in different poses. To make the problem easier, group scales and bounding boxes are provided for test images. Skeleton annotations

Figure 3.8: Example images with ground-truth skeleton annotations from the Leeds Sports (LSP) and Extended Leeds Sports (ELSP) datasets (Johnson and Everingham (2010, 2011)).

consists of 16 body parts. The test annotations are held out. Examples are shown in Figure 3.10.

3.3.6 Microsoft Coco Dataset

Microsoft Coco (Lin *et al.* (2014)) is the largest dataset for multi-person pose estimation. The dataset has 115k training and 5k validation images. The images represent various everyday environments in totally uncontrolled settings. The dataset contains richer annotations including mask, bounding box and skeleton annotations for every person in the image. The skeleton annotations contains location of 17 body parts. Precise visibility flags are also provided. The dataset is challenging as person's scale and location is not available at test time. For testing, test-dev and test-std sets are provided. The ground-truth annotations for both sets are held out. Examples are shown in Figure 3.11.

3.3.7 PoseTrack Dataset

PoseTrack (Iqbal *et al.* (2017)) is the first dataset for the problem of multi-person pose estimation and tracking. The dataset consists of 514 video sequences. It has a total of 66,374 frames with 150k pose annotations. The sequences are split into split into 300, 50 and 208 videos for training, validation and testing, respectively. The validation and test

Figure 3.9: Example images with ground-truth skeleton annotations from the MPII Human Pose dataset (Andriluka *et al.* (2014)).

videos have skeleton annotations for every fourth frame. The ground-truth annotations for test set are held out. Examples are shown in Figure 3.12.

3.4 Evaluation Measures

In this section we describe the different evaluation measures used for evaluating different variants of human pose estimation on different datasets.

Precision per Joint accuracy. The evaluation measure is introduced in (Shotton *et al.* (2011)). It is used for evaluating 3D pose estimation from depth data. According to the measure :

" *A 3D body part prediction, $p \in R^3$, is correct if the euclidean distance D between the prediction and ground-truth is less than or equal to a threshold T* ". Usual choice for T is 20cm.

Percentage of Correct Key Points (PCK). PCK is introduced in (Sapp and Taskar (2013)). It is used for 2D pose estimation. According to the measure :

" *A 2D body part prediction, $p \in R^2$, is correct if the euclidean distance D between the prediction and ground-truth is less than or equal to a threshold T* ".

The distance D is normalised by the size of the torso to incorporate different person sizes. Operating range for T is between 10 and 20.

Percentage of Correct Key Points with head-segment (PCKh). PCKh is introduced in (Andriluka *et al.* (2014)). It is another evaluation measure for 2D pose

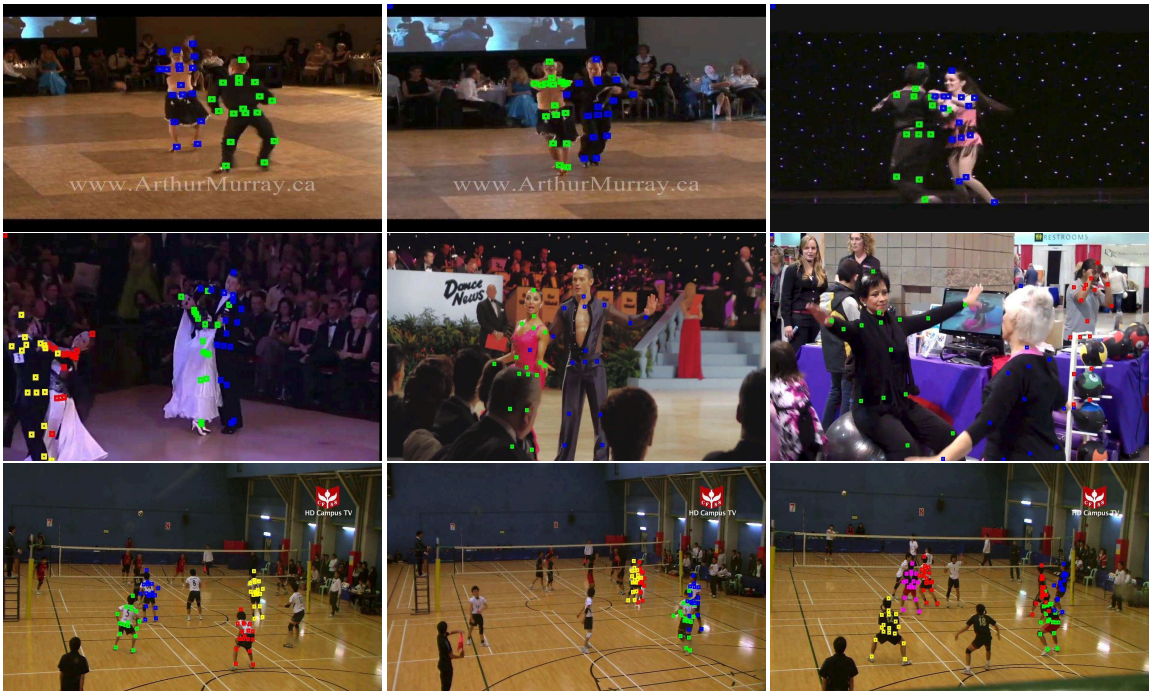


Figure 3.10: Example images with ground-truth skeleton annotations from the MPII Multiperson dataset (Pishchulin *et al.* (2016)).

estimation. It is similar to PCK except the distance D between the prediction and ground-truth is normalised by the head size of the person. The threshold T is set to 50% of the head size.

Object key point similarity (OKS). OKS (Lin *et al.* (2014)) is used for multi-person pose estimation. Given predicted key points $\{p\}_k$ and ground truth key points $\{g\}_k$ for a person, OKS is computed as :

$$OKS = \frac{\sum_k \exp \frac{-d_k^2}{2s_k^2} * \delta(v_k > 0)}{\sum_k \delta(v_k > 0)} \tag{3.1}$$

d_k is the distance between the ground truth and the predicted key point. s_k is the scale of the kth key point. v_k is the visibility flag of the kth key point.

Multi Object Tracking Accuracy and Precision (MOTA and MOTP). For multi-person pose tracking multi-object tracking accuracy (MOTA) (Iqbal and Gall (2017)) and multi-object tracking precision (MOTP) (Iqbal and Gall (2017)) evaluation measures are used. MOTA is based on linear combination of false positives, id switches and missed targets. MOTP is used to measure the localization of each body part belonging to every person in the video sequence.

Figure 3.11: Example images with ground-truth skeleton and mask annotations from the Microsoft Coco dataset (Lin *et al.* (2014)).

3.5 Convolutional Neural Networks (CNNs)

In this section we briefly describe fundamentals of a convolutional neural network (CNN) relevant to this thesis. Interested reader is referred to Goodfellow *et al.* (2016) for details.

3.5.1 Building Blocks of a CNN

A CNN architecture consists of series of three main layers, namely, convolutional layers, pooling layers and fully connected layers. The layers are explained below.

Convolutional Layer. A modern convolutional layer is shown Figure 3.13 (top). It consists of three sub layers.

The filter layer consists of a number of filters. Each filter has a set of weights and is connected to a local region in the input. The output of each filter is generated by sliding the filter over the input. Each filter produces a response map of a width and height depending on the size of the input and shape of the filter. Example is shown in Figure 3.13 (bottom).

Batch normalisation (Ioffe and Szegedy (2015)) is a recent technique to improve convergence speed of CNNs and solves the internal covariance shift problem encountered in CNNs. The idea is to normalize and scale the output of each filter in a filter layer by the batch mean and variance respectively. By doing so, each layer can focus on learning

Figure 3.12: Example images with ground-truth skeleton annotations from the Posetrack dataset (Iqbal *et al.* (2017)).

its features and is not affected by the output of the previous layers. Given a layer with $\{f\}_F$ filters, during training each filter's output is normalised as :

$$\hat{f} = \frac{f - \mathbf{E}(f)}{\mathbf{Var}(f)} \tag{3.2}$$

$\mathbf{E}(f)$ and $\mathbf{Var}(f)$ are batch mean and variance respectively. The above normalization may change the representation learned by the filter f. To make the above normalization an identity transform, a scale γ and offset β is applied to $\hat{f}$ as :

$$\bar{f} = \gamma\hat{f} + \beta \tag{3.3}$$

The parameters γ and β are learned during training. During inference, the output of each layer is normalized by the population mean and variance. Population mean and variance are the moving average of batch means and variances during training.

The output of batch normalisation layer is passed through a ReLU layer. The ReLU applies point-wise activation on the batch normalised response maps. The output of ReLU layer is $\max(0, x)$ where x is the input to a ReLU layer. The derivative of ReLU is 1 and therefore does not affects the incoming gradients and makes training faster.

Pooling Layer. Pooling layer is another essential component of a convolutional neural network. Pooling layer comes in two variants, namely, average pooling and max pooling.

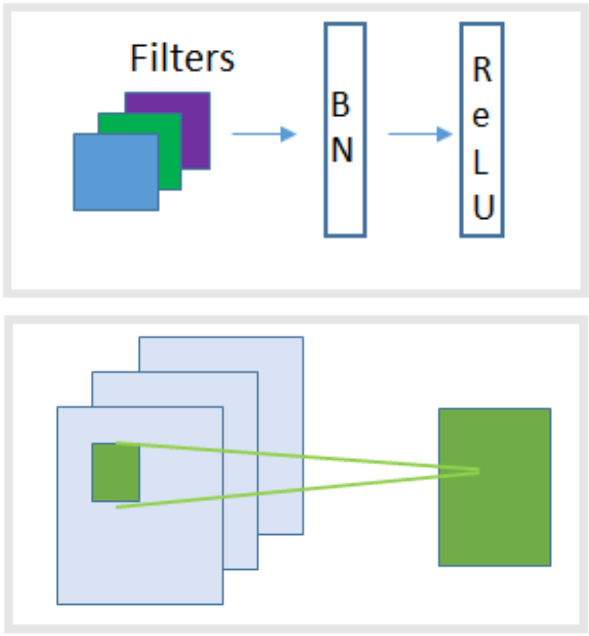


Figure 3.13: (top). A modern convolutional layer. It consists of a filter layer, a batch-normalised layer and a ReLU layer. (bottom) Filter response map generated by sliding the filter over the input.

Example is shown for both variants in Figure 3.14. Pooling layers are used to introduce translational invariance in CNNs.

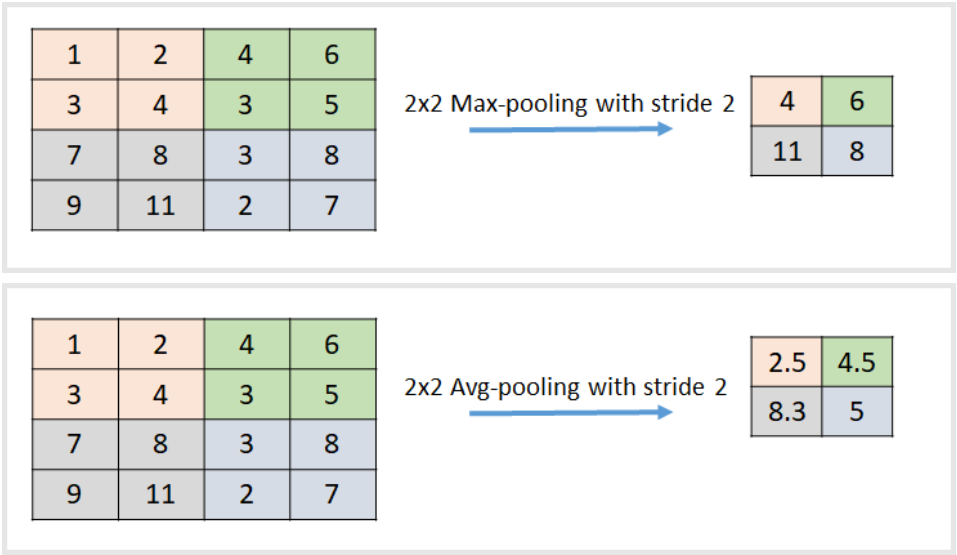


Figure 3.14: Pooling Operation. (top) Max pooling with filter size and stride of 2 (bottom) Average pooling with filter size and stride of 2.

Fully Connected Layer. In fully connected layers each neuron is connected to all the inputs. The is in contrast to convolutional layers where neurons have local connectivity. Fully connected layers are used for classification. The number of output neurons in a fully connected layer is equal to the number of output classes. Example is shown in Figure 3.15.

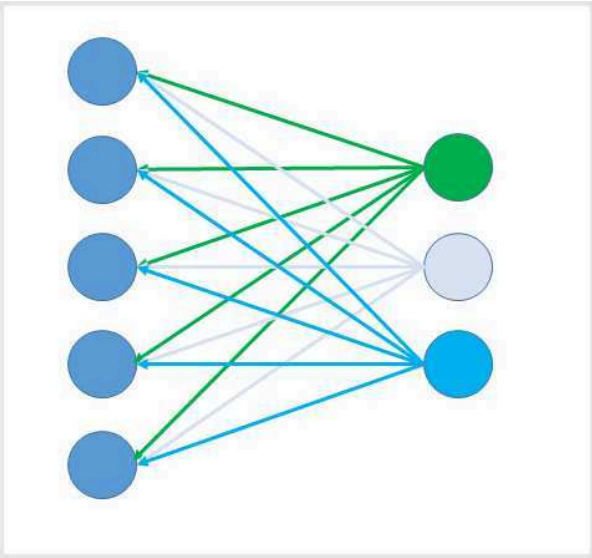


Figure 3.15: Fully connected layer. Each neuron is connected to all the inputs. Fully connected layers are used for classification.

3.5.2 Training CNNs

Convolutional neural networks are trained using an optimization procedure and the back-propagation algorithm. The main idea is to use the back-propagation algorithm to compute the derivatives for all the parameters in the CNN. The parameters are then updated with an optimization procedure, e.g., stochastic gradient descent.

Back-propagation Algorithm. The back-propagation algorithm uses chain rule to compute derivatives recursively from the last layer towards the first layer. For any layer with input $\mathbf{z}^l$, output $\mathbf{z}^{l+1}$ and the network output E, the derivatives $\frac{\partial E}{\partial \mathbf{z}^l}$ are computed as :

$$\frac{\partial E}{\partial \mathbf{z}^l} = \frac{\partial E}{\partial \mathbf{z}^{l+1}} \frac{\partial \mathbf{z}^{l+1}}{\partial \mathbf{z}^l} \tag{3.4}$$

Similarly for any layer with parameters θ^l, output $\mathbf{z}^{l+1}$ and the network output E, the derivatives $\frac{\partial E}{\partial \theta^l}$ are computed as :

$$\frac{\partial E}{\partial \theta^l} = \frac{\partial E}{\partial \mathbf{z}^{l+1}} \frac{\partial \mathbf{z}^{l+1}}{\partial \theta^l} \tag{3.5}$$

Both $\frac{\partial \mathbf{z}^{l+1}}{\partial \theta^l}$ and $\frac{\partial \mathbf{z}^{l+1}}{\partial \mathbf{z}^l}$ are Jacobian matrices.

Stochastic Gradient Descent Optimizer. Given the derivatives computed by the back-propagation algorithm for parameters θ^l of a layer l, the parameters are updated by as :

$$\theta^l_{new} = \theta^l_{old} - \alpha \frac{\partial E}{\partial \theta^l} \tag{3.6}$$

where α is the learning rate. α is a hyper parameter and is optimized on a validation set.

4

3D Human Pose Estimation from depth data with Semantic Occlusion Priors

This chapter is concerned with 3D human pose estimation from depth data. 3D human pose estimation from depth data has made significant progress in recent years and current commercial depth sensors estimate human poses in real-time. However, sensors work well for gaming scenarios, where a person is standing in front of the sensor and can be well-segmented. In many real life scenarios person may be partially occluded by objects and, therefore, the sensors fail to predict the occluded body joints accurately. In such situations it is still desirable by many computer vision applications to estimate the occluded body joints correctly.

In this chapter we propose a 3D human pose estimation framework from depth data where a person is partially occluded by objects[1]. We introduce a semantic occlusion model that is incorporated into the existing regression forest framework for human pose estimation from depth data. Instead of discarding the occluder information, the model exploits the rich context information of occluding objects like a table or a desktop monitor to predict the locations of occluded joints. We evaluate the proposed model on synthetic and real depth data containing partially occluded person and show that the proposed occlusion model improves the joint estimation accuracy and outperforms the commercial Kinect 2 SDK for occluded joints. In this work, we release the Pose Occlusion test set, a data set of real depth images containing person that are partially occluded by objects. We hope that the dataset will facilitate further research in this area.

[1]The work described in this chapter is published in CVPR 2015 Chalearn Looking at People Workshop. -Rafi *et al.* (2015)

4.1 Introduction

Human pose estimation from depth data has made significant progress in recent years. One success story is the commercially available Kinect system (Microsoft Corporation Redmond WA (2010)), which is based on Shotton *et al.* (2011) and provides high-quality body joint predictions in real time. However, the system works under the assumption that the humans can be well segmented and works for scenarios where there is no object between the sensor and the person as shown in rows 1 and 2 in Figure 4.1. This assumption is valid for gaming application for which the device was developed. Many computer vision applications, however, require human pose estimation in more general environments where objects often occlude some body parts as shown in rows 3 and 4 in Figure 4.1. In this case, the current SDK for Kinect 2(Kinect For Windows SDK 2.0 (2012)) fails to estimate the partially occluded body parts. An example is shown in Figure 4.2(a) where the joints of the left leg of the person are wrongly estimated.

In this work, we address the problem of estimating human pose in the context of occlusions. Since for most applications it is more practical to have always the entire pose and not only the visible joints, we aim to predict the locations of all joints even when some of the joints are occluded by objects. To this end, we build on the work from Girshick *et al.* (2011a), which estimates human pose from depth data using a regression forest framework. Similar to the SDK, (Girshick *et al.* (2011a)) works well for visible body parts but it fails to estimate the joint locations of partially occluded parts since it does not model occlusions. We therefore, extend the approach by an occlusion model. Objects, however, not only occlude body parts but they also provide some information about the expected pose. For instance, when a person is sitting at a table as in Figure 4.2, some joints are occluded but humans can estimate the locations of the occluded joints. The same is true if the hands are occluded by a laptop or monitor. In this case, humans can infer whether the person is using the occluded keyboard and they can predict the locations of the occluded hands by using the context of the occluder.

We therefore, introduce a semantic occlusion model that exploits the semantic context of occluding objects to improve human pose estimation from depth data. The model is trained by using synthetic as well as real occluder data. For synthetic data, we use motion capture data and 3D models of furniture. For evaluation, we recorded a test dataset of poses with occlusions. The dataset was recorded from 7 subjects in 7 different rooms using the Kinect 2 sensor. In our experiments, we evaluate the impact of synthetic and real training data and show that our approach outperforms Girshick *et al.* (2011a). We also compare our approach with the commercial pose estimation system that is part of Kinect 2. Although the commercial system achieves a higher detection rate for visible joints since it uses much more training data and highly engineered post-processing, the detection accuracy for occluded joints of our approach is twice as high as the accuracy of the Kinect 2 SDK.

4.2 Contributions

In this chapter we make the following contributions :

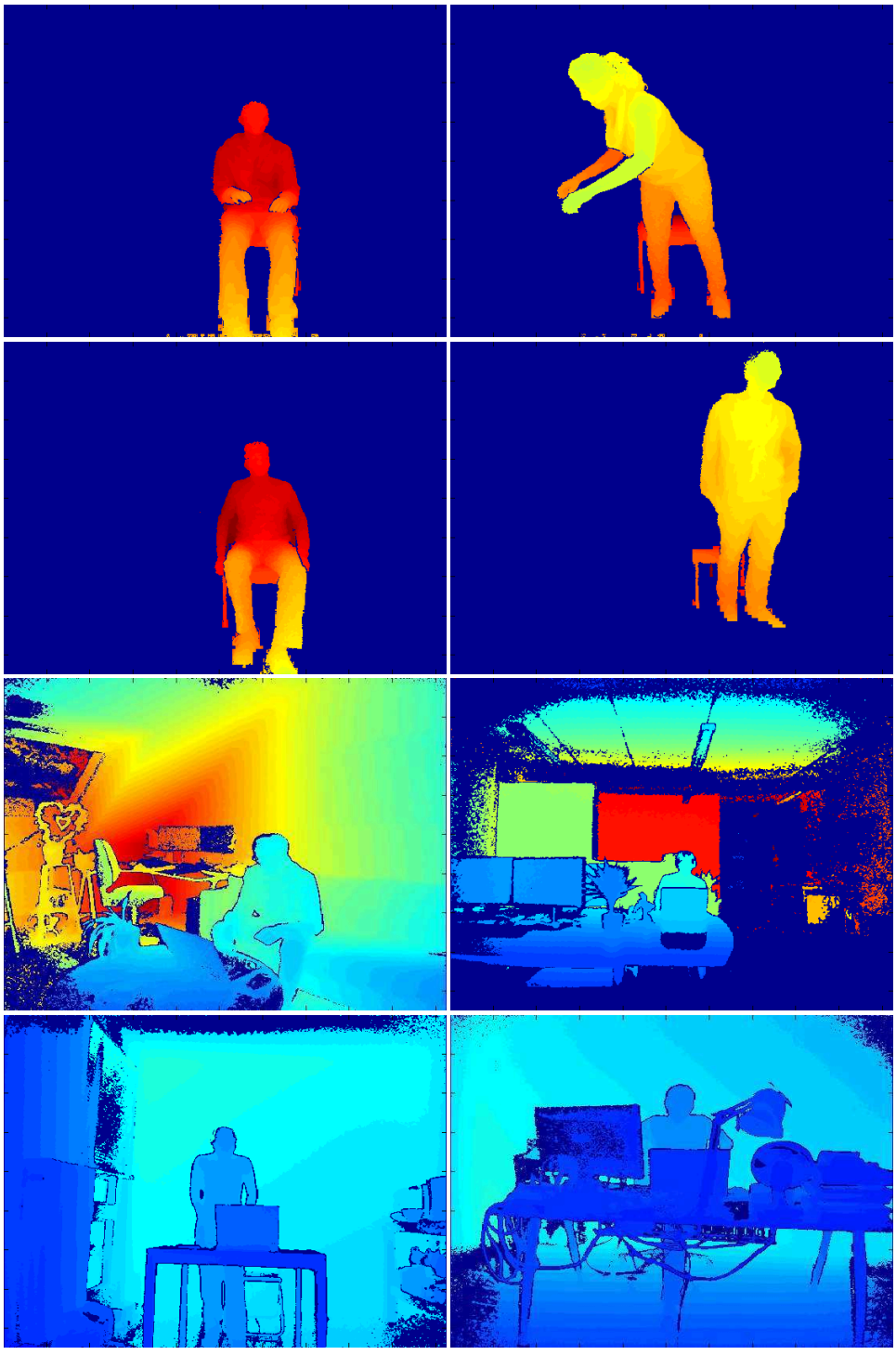
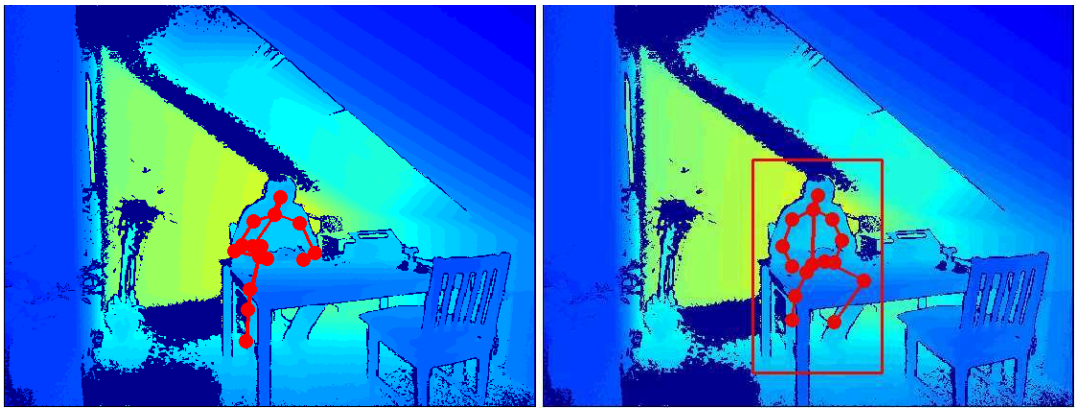


Figure 4.1: Rows 1 and 2 show images where the person can be well-segmented. Rows 3 and 4 show images where person is partially occluded by objects.



(a) (b)

Figure 4.2: Comparison of Kinect 2 SDK and our approach. The SDK estimates the upper body correctly but the left leg is wrongly estimated due to the occluding table (a). Given the bounding box, our approach exploits the context of the table and predicts the left leg correctly (b).

- We propose a 3D human pose estimation framework from depth data where a person is partially occluded by objects.
- We evaluate the proposed model on synthetic and real depth data containing partially occluded person and show that the proposed occlusion model improves the joint estimation accuracy and outperforms the commercial Kinect 2 SDK for occluded joints.
- We release the Pose Occlusion test set[2], a data set of real depth images containing person that are partially occluded by objects. We hope that the dataset will facilitate further research in this area.

4.3 Related Work

3D Human Pose Estimation. Human pose estimation is a challenging problem in computer vision. It has applications in gaming, human computer interaction and security scenarios. It has generated a lot of research surveyed in Poppe (2007). There are a variety of approaches that predict 3D human pose from monocular RGB images (Agarwal and Triggs (2006); Bo *et al.* (2008); Kostrikov and Gall (2014)). However, a major caveat of using RGB data for inferring 3D pose is that the depth information is not available.

In recent years the availability of affordable depth sensors has significantly reduced the depth information problem and further spurred the progress. Researchers have proposed tracking based approaches (Ganapathi *et al.* (2010, 2012); Grest *et al.* (2005); Plagemann

[2]The dataset is available at http://www.vision.rwth-aachen.de/data.

et al. (2010)). These approaches work in real time but operate by tracking 3D pose from frame to frame. They cannot re-initialize quickly and are prone to tracking failures. Recently random forest based approaches (Girshick *et al.* (2011a); Shotton *et al.* (2011); Sun *et al.* (2012); Taylor *et al.* (2012)) have been proposed. They predict the 3D pose in super real time from a single depth image captured by Kinect. These approaches work on monocular depth images and are therefore more robust to tracking failures. Some model based approaches (Baak *et al.* (2011); Ye *et al.* (2011)) have also been proposed that fit a 3D mesh to image data to predict the 3D pose. However, all these approaches require a well-segmented person to predict the 3D pose and will fail for occluded joints when the person is partially occluded by objects. The closest to our method is the approach from Girshick *et al.* (2011a) that uses regression forest to vote for the 3D pose and can handle self-occlusions. Our approach in contrast can also handle occlusions from other objects.

Occlusion Handling. Explicit occluder handling has been used in recent years to solve different problems in computer vision. Girshick *et al.* (2011b) use grammar models with explicit occluder templates to reason about occluded people. The occlusion patterns needs to be specially designed in the grammar. Ghiasi *et al.* (2014) automatically learn the different part-level occlusion patterns from data to reason about occlusion in people-people interactions by using flexible mixture of parts (Yang and Ramanan (2011)). Wang *et al.* (2013) use patch based Hough forests to learn object-object occlusions patterns. In facial landmarks localization, recently some regression based approaches have been proposed that also incorporate occlusion handling for localizing occluded facial landmarks (Burgos-Artizzu *et al.* (2013); Yu *et al.* (2014)). However, these approaches only use the information from non-occluded landmarks in contrast to our approach that also uses the information from occluding objects. Similar to (Ghiasi *et al.* (2014); Wang *et al.* (2013)) our approach also learns the occlusions from data, however, our approach learns occlusions for human-object interactions in contrast to Ghiasi *et al.* (2014) that learns occlusions for human-human interactions and the occluding objects in our approach are classified purely on appearance at test time and does not incorporate any knowledge about distance from the object they are occluding as in Wang *et al.* (2013).

4.4 Semantic Occlusion Model

In this work, we propose to integrate additional knowledge about occluding objects into a 3D pose estimation framework from depth data to improve the pose estimation accuracy for occluded joints. To this end, we build on a regression framework for pose estimation that is based on regression forests (Girshick *et al.* (2011a)). We briefly discuss regression forests for pose estimation in Section 4.4.1. In Section 4.4.2 we extend the approach for explicitly handling occlusions. In particular, we predict the semantic label of an occluding object at test time and then use it as context to predict the position of an invisible joint.

4.4.1 Regression Forests for Human Pose Estimation

Random Forests have been used in recent years for various regression tasks, e.g., for regressing the 3D human pose from a single depth image (Girshick *et al.* (2011a); Shotton *et al.* (2011); Sun *et al.* (2012); Taylor *et al.* (2012)), estimating the 2D human pose from a single image (Dantone *et al.* (2013)), or for predicting facial features (Dantone *et al.* (2012)). In this section, we briefly describe the approach from Girshick *et al.* (2011a), which will be our baseline.

Regression forests belong to the family of random forests and are ensembles of T regression trees. In the context of human pose estimation from a depth image, they take an input pixel q in a depth image D and predict the probability distribution over the locations of all joints in the image. Each tree t in the forest consists of split and leaf nodes. At each split node a binary split function $\phi_\gamma(q, D) \mapsto \{0, 1\}$ is stored, which is parametrized by γ and evaluates a pixel location q in a depth image D. In this work, we use the depth comparison features from Shotton *et al.* (2011):

$$\phi_\gamma(q, D) = \begin{cases} 1 & \text{if } D\left(q + \frac{u}{D(q)}\right) - D\left(q + \frac{v}{D(q)}\right) > \tau \\ 0 & \text{otherwise.} \end{cases} \tag{4.1}$$

where the parameters $\gamma = (u, v, \tau)$ denote the offsets u, v from pixel q, which are scaled by the depth at pixel q to make the features robust to depth changes. The threshold converts the depth difference into a binary value. Depending on the value of $\phi_\gamma(q, D)$, (q, D) is send to the left or right child of the node. During training each such split function is selected from a randomly generated pool of splitting functions. This set is evaluated on the set of training samples $Q = \{(q, D, c, \{V_j\})\}$, each consisting of a sampled pixel q in a sampled training image D, a class label c for the limb the pixel belongs to, and for each joint j the 3D offset vectors $V_j = q_j - q$ between pixel q and the joint position q_j in the image D.

Each sampled splitting function ϕ, partitions the training data at the current node into the two subsets $Q_0(\phi)$ and $Q_1(\phi)$. The quality of a splitting function is then measured by the information gain:

$$\phi^* = \arg\max_\phi g(\phi), \tag{4.2}$$

$$g(\phi) = H(Q) - \sum_{s \in \{0,1\}} \frac{|Q_s(\phi)|}{|Q|} H(Q_s(\phi)), \tag{4.3}$$

$$H(Q) = -\sum_c p(c|Q) \log(p(c|Q)), \tag{4.4}$$

where $H(Q)$ is the Shannon entropy and $p(c|Q)$ the empirical distribution of the class probabilities computed from the set Q. The training procedure continues recursively until the maximum allowed depth for a tree is reached.

At each leaf node l, the probabilities over 3D offset vectors V_j to each joint j, i.e., $p_j(V|l)$ are computed from the training samples Q arriving at l. To this end, the vectors V_j are clustered by mean-shift with a Gaussian kernel with bandwidth b and for each

joint only the two largest clusters are kept for efficiency. If V_{ljk} is the mode of the k^{th} cluster for joint j at leaf node l, then the probability $p_j(V|l)$ is approximated by

$$p_j(V|l) \propto \sum_{k \in K} w_{ljk} \cdot \exp\left(-\left\|\frac{V - V_{ljk}}{b}\right\|_2^2\right) \tag{4.5}$$

where the weight of a cluster w_{ljk} is determined by the number of offset vectors that ended in the cluster. Cluster centers with $\|V_{ljk}\| > \lambda_j$ are removed since these correspond often to noise Girshick *et al.* (2011a).

For pose estimation, pixels q from a depth image D are sampled and pushed through each tree in the forest until they reach a leaf node l. For each pixel q, votes for the absolute location of a joint j are computed by $x_j = q + V_{ljk}$. In addition a confidence value that takes the depth of the pixel q into account is computed by $w_j = w_{ljk} \cdot D^2(q)$. The weighted votes for a joint j are collected for all pixels q and form the set $\mathcal{X}_j = \{(x_j, w_j)\}$. The probability of the location of a joint j in image D is then approximated by

$$p_j(x|D) \propto \sum_{(x_j, w_j) \in \mathcal{X}_j} w_j \cdot \exp\left(-\left\|\frac{x - x_j}{b_j}\right\|_2^2\right) \tag{4.6}$$

where b_j is the bandwidth of the Gaussian kernel learned separately for each joint j. As for training, the votes are clustered and only the clusters with the highest summed weights w_j are used to predict the joint location.

4.4.2 Occlusion Aware Regression Forests (OARF)

In order to handle occlusions, we propose Occlusion Aware Regression Forests (OARF) that build on the regression forest framework described in Section 4.4.1. They predict additionally the class label of an occluding object at test time and then use this semantic knowledge about the occluding object as context to improve the pose estimation of occluded joints. To this end, we use an extended set of training samples $Q_{ext} = Q \cup Q_{occ}$, where $Q_{occ} = \{(q_{occ}, D, c_{occ}, \{V_{jocc}\})\}$, is a set of occluding object pixels where each pixel q_{occ} is sampled from a training image D, has a class label c_{occ} and a set of offset vectors $\{V_{jocc}\}$ to each joint of interest j.

During training we use the depth comparison features described in Section 4.4.1 for selecting a binary split function $\phi_\gamma(q, D)$ at each split node in each tree. To select binary split functions that can distinguish between occluding objects and body parts we minimize the Shannon entropy $H(Q)$ over the extended set of class labels $c_{ext} = c \cup c_{occ}$:

$$H(Q) \quad = \quad -\sum_{c_{ext}} p(c_{ext}|Q) \log(p(c_{ext}|Q)), \tag{4.7}$$

To use the semantic knowledge about occluding object as an additional cue for the prediction of occluded joints, we also store at a leaf node l the probabilities over 3D offset vectors V_{jocc} to each joint j, i.e., $p_j(V_{occ}|l)$, that are computed from the training

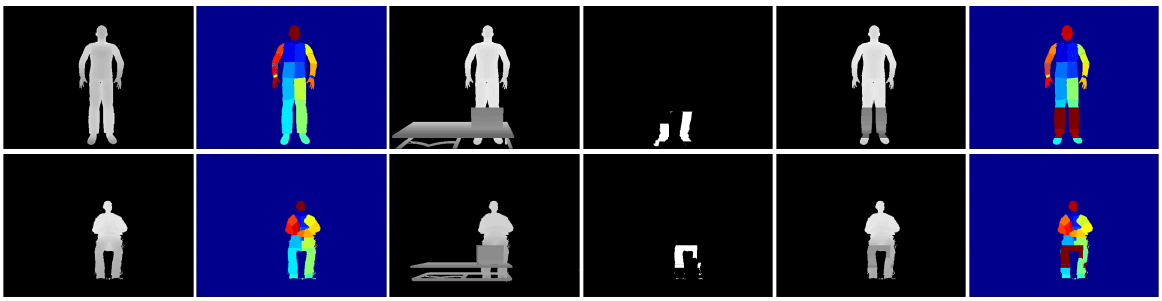


Figure 4.3: Procedure for generating depth images with pixel level ground truth body parts labels and occluding objects masks with class labels. Synthetic data (top row) and real data (bottom row): (a) Rendered or captured and segmented depth image. (b) Body part labels at pixel level. (c) Depth data with object. (d) Occluded body parts. (e) Segmented depth map with occluding objects. (f) Combined labels of body parts and occluding objects.

samples Q_{occ} arriving at l by using the mean shift procedure described in Section 4.4.1. The probability $p_j(V_{occ}|l)$ is approximated by

$$p_j(V_{occ}|l) \propto \sum_{k \in K} w_{ljk} \cdot \exp\left(- \left\| \frac{V_{occ} - V_{ljocck}}{b} \right\|_2^2 \right) \tag{4.8}$$

The inference procedure is similar to standard regression forest inference. At test time pixels q_{occ} from occluding objects in a test image D are also pushed through each tree in the forest until they reach a leaf node l and cast a vote $x_{jocc} = q_{occ} + V_{ljocck}$ with confidence w_{jocc} for each joint j. The weighted votes for each joint j form the set $\mathcal{X}_j = \{(x_j, w_j) \cup (x_{jocc}, w_{jocc})\}$. The final joint position is then predicted by using the mean shift procedure described in Section 4.4.1.

4.5 Training and Test Data

Gathering real labeled training data for 3D human pose estimation from depth images is expensive. To overcome this, Shotton *et al.* (2011) generated a large synthetic database of depth images of people covering a wide variety of poses together with pixel annotations of body parts. For generating such a database, a large motion capture corpus of general human activities has been recorded. The body poses of the corpus were then retargeted to textured body meshes of varying sizes. Since the dataset is not publicly available, we captured our own dataset.

Synthetic Training and Test Data. We follow the procedure from Shotton *et al.* (2011) to generate synthetic training and test data. To this end, we use the motion capture data for sitting and standing poses from the publicly available CMU motion capture database (CMU Mocap (2008)). We retarget the poses to 6 textured body meshes using Poser (Poser (2010)), a commercially available animation package, and generate a synthetic dataset of 1110 depth images of humans in different sitting and

Figure 4.4: Synthetically generated depth-maps with annotations for various 3D models in different poses and view points.

standing poses from different viewpoints. For each depth image, we also have a pixelwise body part labeling and the 3D joint positions. The depth maps and body part labels are shown in Figure 4.3(a-b). We split the dataset into training and test sets respectively. Examples of synthetically generated depth-maps with annotations are shown in Figure 4.4.

For the occluding objects, we use the publicly available 3D models of tables and laptops from the SweetHome 3D Furniture Library (Sweet Home 3D (2014)). We render the objects together with the humans as shown in Figure 4.3(c). The compositions are randomized under the constraints that the tables and feet are on the ground plane, the laptops are on the tables and the distance between the humans and the objects is between 3-5 cm. For each composition, we compute the occluded body parts (Figure 4.3(d)) and the depth and class labels with occluding object classes as shown in Figure 4.3(e-f).

Real Training Data. We recorded a dataset using the Kinect 2 sensor. The dataset contains depth images of humans in different sitting and standing poses without occlusions as shown in Figure 4.3(a). The 3D poses are obtained by the Kinect SDK and we discarded images were the SDK failed. This resulted in 2552 images. The pixelwise segmentation of the body parts is obtained by the closest geodesic distance of a surface point to the skeleton as shown in Figure 4.3(b). The composition with synthetic 3D objects is done as for the synthetic data. Examples of real depth maps with occluder masks for different actors are shown in Figure 4.5.

Real Test data. We recorded a test dataset using the Kinect 2 sensor. The dataset consists of 1000 depth images of partially occluded people. The dataset is recorded using in different indoor environments, offices and living rooms, and consists of seven different subjects in different sitting and standing poses from different viewpoints. The dataset is further split into test and validation set with 800 and 200 images, respectively. Each image in the dataset consists of a single person partially occluded by objects like table, laptop and monitor. Each image is annotated with 2D ground-truth positions of 15 body joints and their visibility. Noisy Foreground mask for each image is also provided. Example images and foreground masks from the data set are shown in Figure 4.6. The occluded joints are shown in red.

4.6 Experiments and Results

In this section we describe in detail the experimental settings used to evaluate our method and report quantitative and qualitative results. For comparison, we consider three approaches:

1. The approach from Girshick *et al.* (2011a) described in Section 4.4.1 is our baseline. It is trained on the training data without occluding objects shown in Figure 4.3(a-b).

2. The occlusion aware regression forest (OARF W semantics) described in Section 4.4.2 is trained with the semantic labels of occluding objects shown in Figure 4.3(e-f).

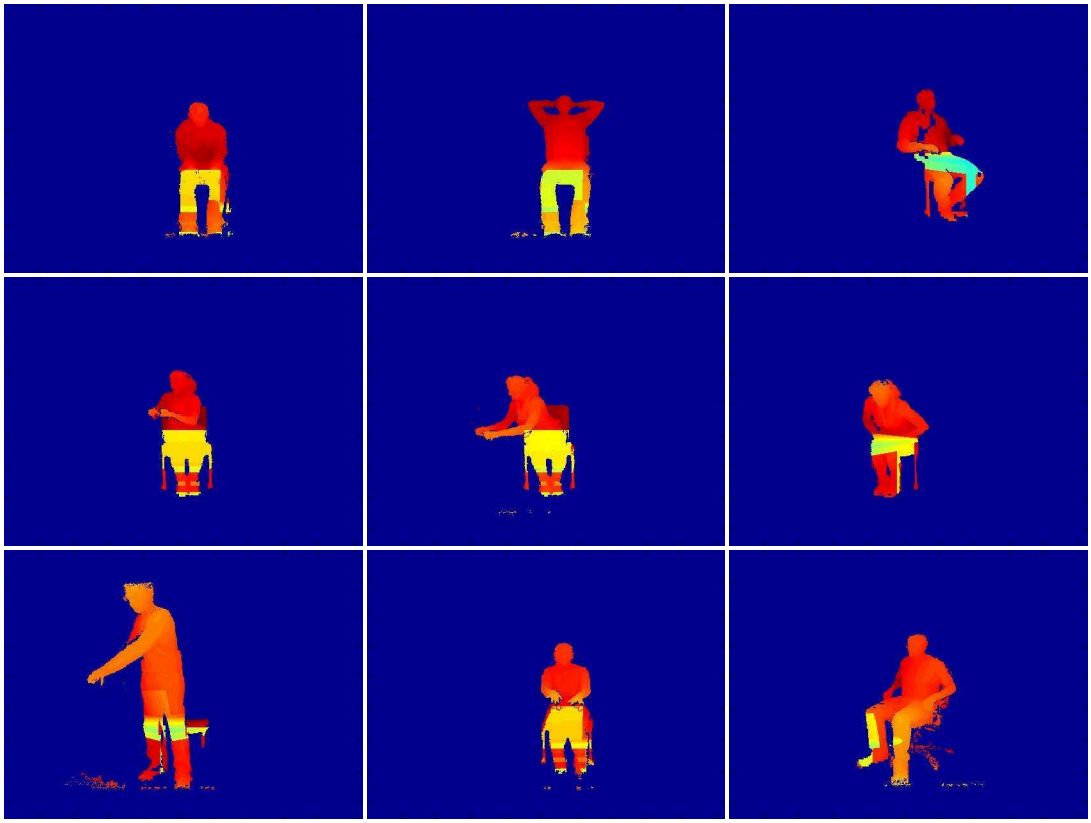


Figure 4.5: Real depth images with synthetic occluder masks for various real subjects.

3. To show the impact of the semantic information of the occluding objects, we also train an OARF by assigning all occluding objects a single label and not the labels of the object classes (OARF W/O semantics).

Training Procedure. We train all forests with the same parameters. For each training depth image, we sample 1000 pixel locations. The other parameters are used as in Girshick *et al.* (2011a), i.e., the regression forests consist of 3 trees with maximum depth 20. For each splitting node, we sample 2000 depth comparison features and use $b = 0.05m$.

Testing Procedure. For testing our approach, we use synthetic and real data. The real dataset is recorded in different indoor environments, offices and living rooms, and consists of seven sequences of different subjects in different sitting and standing poses. From each sequence, we select a set of unique poses thus providing us with a total of 1000 images. We split the 1000 images into test and validation set with 800 and 200 images, respectively. While b_j were set as proposed in Girshick *et al.* (2011a), we observed that the values for λ_j proposed in Girshick *et al.* (2011a) are not optimal for our baseline. We therefore estimate them on the validation set. The synthetic test set consists of 440 images of 2 subjects. The subjects, occluding objects and the poses in both test sets are

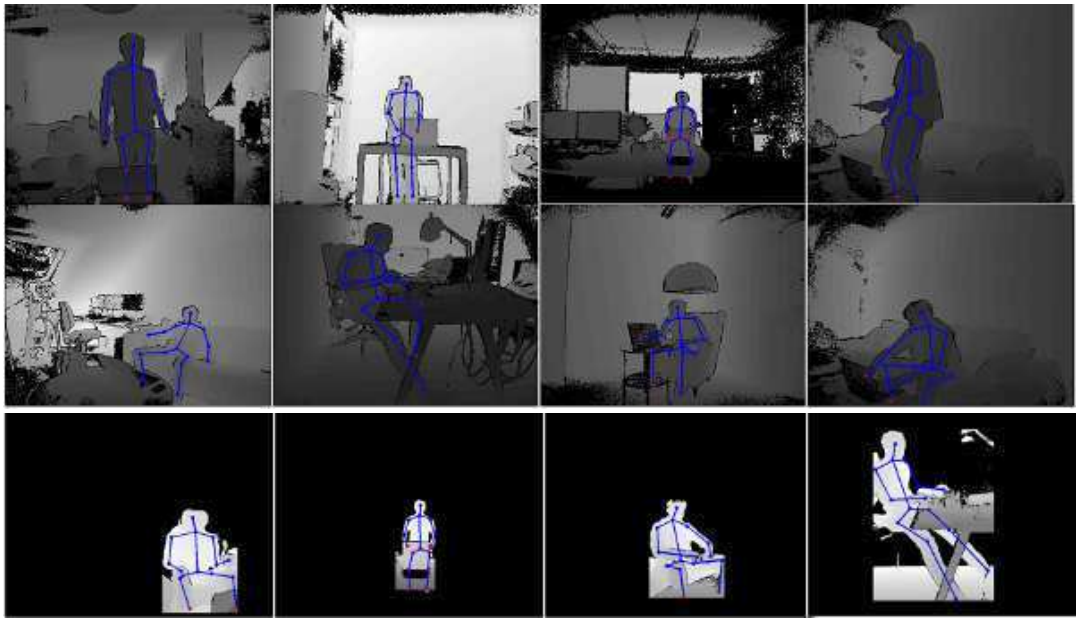


Figure 4.6: Example depth images from the Pose-Occlusion test set with ground-truth skeletons. Row 3 shows noisy foreground masks.

different from the ones in the training set. We report quantitative results on real and synthetic test sets for the 15 body joint positions of our skeleton.

Performance Measure for Synthetic Test Data. For quantitative evaluation on synthetic data, we use the performance measure from Shotton *et al.* (2011). According to the measure a joint is predicted correctly if the distance between the predicted and ground-truth 3D position of the joint is with-in a certain distance threshold. In this work, we use distance threshold of 10 cm as done in Shotton *et al.* (2011).

We report the mean average precision over the 3D body joint predictions in Table 4.1. Integrating the additional knowledge about occluding objects without object class labels alone provides 4% improvement over the baseline. When we also integrate semantic knowledge about occluding objects then this provides a significant improvement of 10% over the baseline. This shows that occlusion handling is beneficial for pose estimation, but also that the semantic context of occluding objects contains important information about joint locations. Qualitative results for body-parts and occluder labels predicted by the model are shown in Figure 4.7.

Performance Measure for Real Test Data. For real test data, it is difficult to get 3D ground truth body joint positions. For the quantitative evaluation, we therefore manually labeled the 2D body joint positions. Since manual annotations of the 2D positions of body joints, especially occluded joints, in depth images are sometimes noisy, we used the mean annotations of three different annotators as ground truth. For the quantitative evaluation on the test data, we use the evaluation measure from Dantone

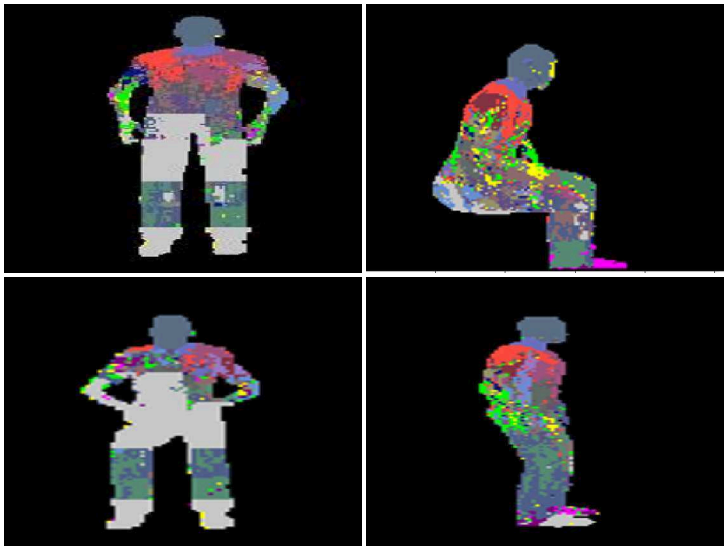


Figure 4.7: Example test images from synthetic test data with body parts and occluder labels predicted by the Semantic Occlusion Model.

Method	Mean Average Precision
Baseline	44%
OARF W/O semantics	48%
OARF W semantics	54%

Table 4.1: Mean average precision of the 3D body joint predictions on the synthetic test set by using the evaluation measure from Shotton *et al.* (2011) with a distance threshold of 10 cm. The results show that integrating semantic knowledge about occluding objects provides a significant improvement over the baseline.

et al. (2013). A joint is considered correctly predicted if the distance between the predicted 2D position and ground-truth for that joint, normalised by the size of the torso of that person, lies with an error threshold of 0.1 of upper body size.

We report average detection accuracy over 2D body joint predictions. In Table 4.2, the results for the baseline, OARF with and without semantics are reported. We also evaluate the impact of the synthetic and real training data. The results show that the synthetic training data is not as good as the real training data. However, if we combine real and synthetic data we get another boost of performance. For all three training settings, the numbers support the results of the synthetic test set. The baseline is improved by occlusion handling and semantic occlusion handling achieves the best result of the three methods. The semantic occlusion model mainly improves the accuracy of occluded joints, but there is also a slight improvement of non-occluded joints. Without occlusion handling, objects close to joints introduce some noisy votes that can result

Setting	Average Detection Accuracy(%)		
	Occluded Joints	Non Occluded Joints	All Joints
Synthetic Data			
Baseline	22.17	50.51	44.54
OARF W/O semantics	24.36	52.57	46.62
OARF W semantics	25.42	52.43	46.73
Real Data			
Baseline	25.42	46.21	41.82
OARF W/O semantics	28.26	49.72	45.19
OARF W semantics	31.12	51.69	47.94
Real + Synthetic Data			
Baseline	28.62	55.02	49.45
OARF W/O semantics	32.60	55.50	50.66
OARF W semantics	35.77	56.01	51.72
Kinect SDK	18.13	66.36	56.94

Table 4.2: Average detection accuracy of 2D body joint predictions on the real test set measured by the evaluation measure from Dantone *et al.* (2013) with an error threshold of 0.1 of upper body size.

in wrong estimates. The occlusion handling reduces this effect. We also compared our results to the Kinect SDK. The Kinect SDK achieves a higher overall accuracy since it is trained on much more training data and the SDK includes additional post-processing, which is not part of our baseline. However, our approach achieves a much better accuracy for the occluded joints.

Table 4.3 presents detailed quantitative results for 11 body joints that are occluded in the real test set. The results are reported for OARF trained on real and synthetic data and for the Kinect SDK. The results show that for most joints our model achieves a better accuracy than the Kinect SDK and adding semantic knowledge provides further improvements. There are only three joints, namely the right elbow and the ankles, with a low accuracy. This can be explained by the training data that contains only few examples where these joints are occluded.

Qualitative Results. We show some qualitative results on our real test set in Figure 4.8 and a few failure cases in Figure 4.9. The top row shows the body joints predicted by OARF with the semantic occlusion model, the middle row shows the body joints predicted by OARF without the semantic occlusion model and the bottom row shows the body joints predicted by the Kinect SDK.

Joint	Per Joint Detection Accuracy(%)		
	OARF W semantics	OARF W/O semantics	Kinect SDK
Spine	77.51	73.9	42.57
Left Elbow	76.47	73.53	47.06
Right Elbow	17.5	10.53	71.93
Left Hand	47.59	36.55	28.28
Right Hand	59.38	39.58	18.23
Left Hip	50.53	48.76	10.60
Right Hip	75.42	71.25	22.08
Left Knee	31.64	31.64	22.60
Right Knee	27.27	29.09	12.73
Left Ankle	3.87	4.9	5.15
Right Ankle	5.44	5.07	7.88
Average	35.77	32.60	18.13

Table 4.3: Per joint detection accuracy of 11 occluded body joints in our real test set by using the evaluation measure from Dantone *et al.* (2013) with an error threshold of 0.1 of upper body size. The results are reported for OARF with and without semantics and for the Kinect SDK.

4.7 Conclusion and Future directions

In this chapter, we have proposed a semantic occlusion model for 3D human pose estimation that integrates the rich contextual knowledge about occluding objects into the existing 3D pose estimation framework from depth data. We have shown that occluding objects not only need to be detected to avoid noisy estimates, but also that the semantic information of occluding objects is a valuable source for predicting occluded joints. Although our experiments already indicate the potential of the approach and outperform a commercial SDK already for occluded joints, the overall performance can still be boosted by increasing the variety of objects and poses in the training data.

The proposed framework uses hand-crafted features in combination with the regression forest framework described in 4.4.1. Recently, End-to-End Convolutional Neural Networks (CNNs) based frameworks have become a popular choice for many challenging computer vision problems and have achieved great success as compared to regression forest frameworks. Exploring an End-to-End framework for predicting the poses of partially occluded people is an interesting future direction.

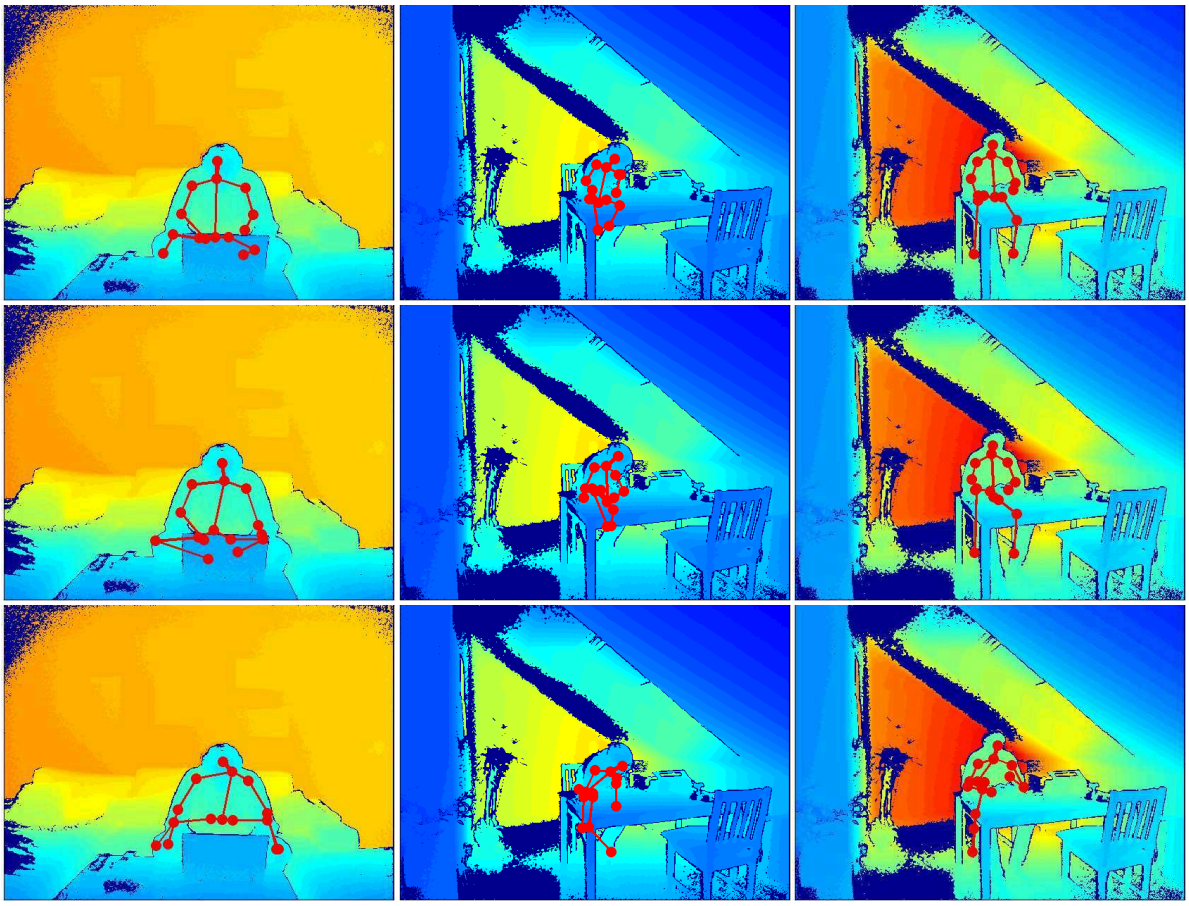


Figure 4.8: Qualitative results for some sample images taken from our real test set. The top row shows the body joints predicted by OARF with semantics. The middle row shows the body joint predictions from OARF without semantics. The bottom row shows the body joints predicted by the Kinect SDK. The results show that integrating knowledge about occluding objects helps in improving the body joint prediction performance for the occluded joints.

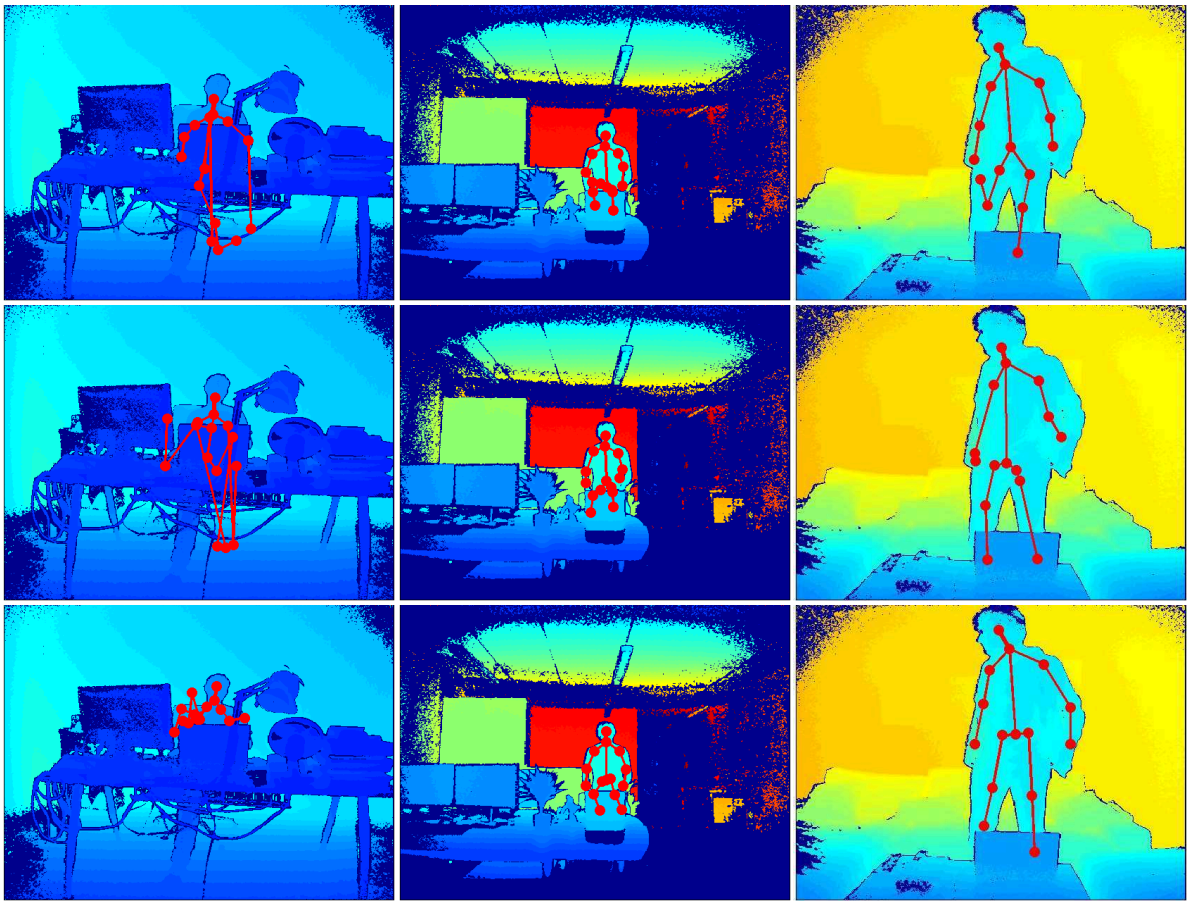


Figure 4.9: Example failure cases for occluded joints from our real test set. OARF with semantics (top row), OARF without semantics (middle row) and Kinect SDK (bottom row).

5

An Efficient Deep Learning method for 2D Single Person Pose Estimation

In this chapter, we focus on 2D single person pose estimation from RGB images. 2D single person pose estimation from RGB data is a challenging task since the scale of a person is not known. In recent years, 2D single pose estimation has greatly benefited from deep learning and huge gains in performance have been achieved. However, the trend is to push for maximum performance by employing pre-trained convolutional neural networks (CNNs) with large computational budget and fine-tune them on multiple datasets.

Despite impressive performance, CNNs with large computational budget require high-end graphical processing units (GPUs) to run. This is unrealistic for many real time computer vision applications in robotics and hand-held devices where GPUs with low computational budget are available. In addition to that, it is hard to reproduce state-of-the-art performance.

In this chapter, we propose an efficient convolutional neural network architecture that runs efficiently on mid-range GPUs.[1]. We propose a transparent training procedure that is simple to replicate. Despite the low computational requirements of our CNN, it is on par with much more complex models on popular benchmarks for human pose estimation. The proposed architecture may serve as a simple baseline for future work. We release the code for training and evaluating the CNN in order to ensure reproducibility of work.

[1]Part of the work described in this chapter is published in BMVC 2016. -Rafi *et al.* (2016)

5.1 Introduction

Convolutional networks have raised the bar substantially for many computer vision benchmarks. Human pose estimation is such an example where methods based on convolutional networks dominate the leader boards (Andriluka *et al.* (2014); Johnson and Everingham (2010)). Despite the recent success in human pose estimation, a direct comparison between the architectures remains difficult. For architectures that do not provide the source code for training and testing, the reproducibility of the results can be very difficult due to small details that might be essential for the performance, such as the used image resolution or the exact form of data augmentation. Very often, pre-trained models are used that are fine-tuned on the benchmark datasets, making it difficult to compare them with methods that are trained from scratch on benchmarks and therefore on less training data. Another issue is the increasing complexity of the models for human pose estimation. Despite the impressive accuracy they achieve, computationally expensive network architectures with a large memory footprint can be impractical for many applications since they require high-end graphics cards.

In this work, we propose an efficient network architecture for human pose estimation that exploits the best current design choices for network architectures with a low memory footprint and we train it using best-practice ingredients for efficient learning. An important design choice is to learn features in different layers at multiple scales. This has been recently exploited in the context of classification by Szegedy *et al.* (2015) through the introduction of inception layers. Surprisingly, very little attention has been paid on exploiting this concept for human pose estimation. Similarly, learning features at multiple image resolutions has been shown to be very effective for human pose estimation Tompson *et al.* (2014), as it helps the network to use a larger context for difficult body joints like wrists and ankles. In our network, we combine both ideas to achieve maximum performance. Another important aspect is the use of features from the middle layers in addition to features from the last layer. While coarse features from the last layer are very good for classification but poor for localisation due to pooling, features from the middle layers are better for localisation. Long *et al.* (2015) exploited this for semantic segmentation and Hariharan *et al.* (2015) used it for object localisation. Similarly, using context around features from the last layer has shown to be very effective in the context of semantic segmentation Lin *et al.* (2016). Our network also exploits context around features from the last layer along with features from a middle layer. Other recent advances in deep learning, e.g., Adam optimiser Kingma and Ba (2014), exponential learning rate decay, batch normalisation Ioffe and Szegedy (2015) and extensive data augmentation, also have shown to provide further benefits for the overall performance. We therefore exploit the above ingredients to add additional gains to the performance.

Based on the design choices, we propose a network architecture for human pose estimation that is efficient to train and has a low memory footprint such that a mid-range GPU is sufficient. Yet, our network architecture achieves state-of-the-art accuracy on

the most popular benchmarks for human pose estimation, indicating that very complex architectures might not be needed for the task. For best comparison and reproducibility, we evaluate the network using the protocols of state-of-the-art benchmarks without any pre-training or post-processing. The learned models for all benchmarks and the source code for training and testing are publicly available and serve as an up-to-date baseline for more complex models.

5.2 Contributions

In this chapter we make the following contributions :

- We propose a convolutional neural network (CNN) for human pose estimation that is efficient to train and has a low memory footprint such that a mid-range GPU is sufficient
- We evaluate the proposed CNN on the the MPII Human Pose (MPII) dataset (Andriluka *et al.* (2014)), the Leeds Sports Pose (LSP) dataset(Johnson and Everingham (2010, 2011)) and the Frames Labelled In Cinema (FLIC) dataset (Sapp and Taskar (2013)) and achieve competitive performance.
- We release the code[2] to ensure reproducibility of work and facilitate further research in the field.

5.3 Related Work

Human pose estimation has been intensively studied in the last decades. The classical approaches are based on the pictorial structure model (Andriluka *et al.* (2009, 2010); Dantone *et al.* (2013); Felzenswalb and Huttenlocher (2005); Johnson and Everingham (2010); Pishchulin *et al.* (2013a,b); Yang and Ramanan (2011)) that uses a tree-structured graphical model to encode spatial dependencies between neighbouring joints. These approaches have shown to be successful for many applications but they can suffer from double counting, for instance, in case of occlusions. Another line of research is based on hierarchical models (Sun and Savarese (2011); Tian *et al.* (2012)) that first detect larger body parts in the image and then condition the detection of smaller body parts on the detected larger body parts. Complex non-tree models (Sigal and Black (2006)) have also been used that model spatial dependencies between unconnected body parts. Loopy belief propagation is then used for approximate inference to predict the joint positions in the image. Recently, sequential prediction machines (Ramakrishna *et al.* (2014)) have been proposed that combine the benefit of modelling complex spatial dependencies between joints with an efficient inference procedure.

Approaches based on CNNs became popular in the last two years. (Toshev and Szegedy (2014)) first used CNNs to directly regress the positions of body joints. Tompson *et al.* (2014) have shown that predicting belief maps as opposed to point estimates of

[2] https://web-info8.informatik.rwth-aachen.de/software/pose-cnn

Architecture	Err %	Forward-Pass time (ms)
VGGA	29.6	372
VGGD	26.8	687
ResNet-34	26.7	231
BN-GoogleNet	25.2	192
ResNet-50	24.0	403
ResNet-101	22.4	649
Inception-v3	21.2	494

Table 5.1: Comparison of speed/accuracy for popular CNN architectures used in Image Classification.

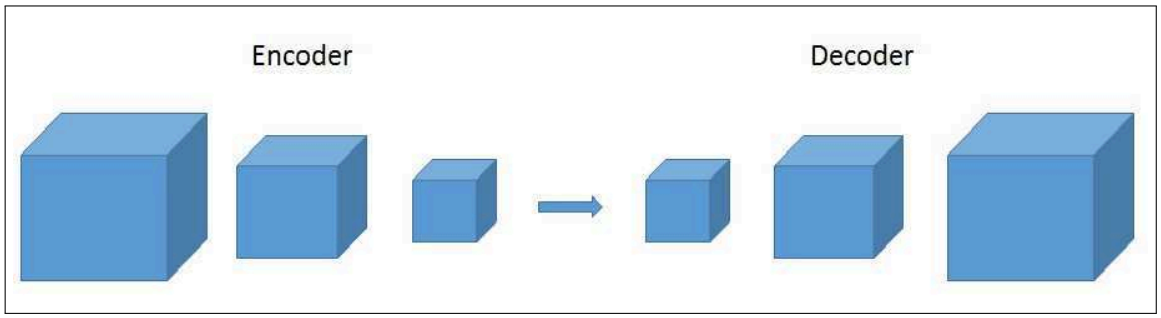


Figure 5.1: Deep Encoder-Decoder architecture used in recent human pose estimation frameworks.

joints improves accuracy. In their later work, Tompson *et al.* (2015) further improved performance by introducing a cascade architecture that compensates for the negative effect of pooling. In Carreira *et al.* (2016) the joint positions are not directly predicted, but an iterative procedure is used to refine the pose step by step until it converges to a pose configuration. While the majority of approaches estimates the pose of a single person, multi-person pose estimation is addressed in Pishchulin *et al.* (2016). Very recently, convolutional pose machines (Wei *et al.* (2016)) have been proposed that stack multiple CNNs where each CNN refines the pose. The method achieves very accurate pose estimates, but it is very expensive to train and requires 6GB of GPU memory. In contrast, our network is fast to train and requires only 3GB which makes our network also suitable for mid-range GPUs like GTX980.

5.4 Fully Convolutional Deep Network for Human Pose Estimation

For 2D human pose estimation, the positions of all body joints in an image need to be predicted. Recent approaches (Tompson *et al.* (2014); Toshev and Szegedy (2014)) have shown that regressing point estimates for body joints may be sub-optimal and a better

strategy is to use fully convolutional neural network (CNNs) architectures to predict dense belief maps for each body joint. However, the max-pooling used in the CNNs results in low resolution belief maps and effects the over-all joint prediction accuracy. Popular choice is to use encoder-decoder architectures as shown in Figure 5.1. The decoder up samples the low resolution features generated by the encoder to a desired higher resolution belief maps in a feed forward manner.

In addition to that, the fully convolutional networks can be very inefficient in terms of memory usage and training time, when not designed carefully. Table 5.1 lists the performance/speed comparison of popular CNNs architectures for image classification. BN-GoogleNet seems an attractive choice with top speed and competitive performance compared to other architectures.

In this works, we therefore, adapt the batch-normalised GoogleNet from Ioffe and Szegedy (2015) into a fully convolutional encoder. The encoder produces low resolution features that are upsampled by a pose decoder into belief maps for body joints. In this work, we refer the fully convolutional encoder as Fully Convolutional Google Net(FCGN)

We propose two frameworks for Human Pose Estimation. The frameworks build on the top of FCGN. FCGN and both frameworks are described in next section.

5.4.1 Human Pose Estimation Framework

Fully Convolutional Google Net . FCGN is illustrated in Figure 5.2(a). We use the first 17 layers of (Ioffe and Szegedy (2015)) and remove the average pooling, drop-out, linear and soft-max layers from the last stages of the network. We add a skip connection to combine feature maps from layer 13 with feature maps from layer 17. We upsample the feature maps from layer 17 to the resolution of the feature maps from layer 13 by a deconvolution filter with size 2×2 and stride 2. The output of the FCGN1 consists of feature maps from layer 13 and 17 that have 16 times lower resolution than the input image due to pooling.

PoseNet1 PoseNet1 [3] is illustrated in Figure 5.2(b). It uses two FCGNs with shared weights, where each FCGN1 takes the same image at a different resolution and produces feature maps as previously described. The feature maps obtained from the half resolution image are upsampled to the resolution of the feature maps extracted from the full resolution image by a deconvolution filter with size 2×2 and stride 2. The feature maps of the half resolution and full resolution FCGN1 are then directly upsampled to obtain belief maps for different body joints by a pose decoder. In pose decoder, we use a larger deconvolution filter of size 32×32 and stride 16. Due to the large deconvolution filter, we implicitly exploit the context of neighbouring pixels in the feature maps for predicting belief maps for joints. The belief maps are then normalised by using a sigmoid function. We also use spatial drop out from Tompson *et al.* (2015) before upsampling to further regularise our network. All non-linearities in PoseNet1 are ELU.

[3] PoseNet1 is published in Rafi *et al.* (2016).

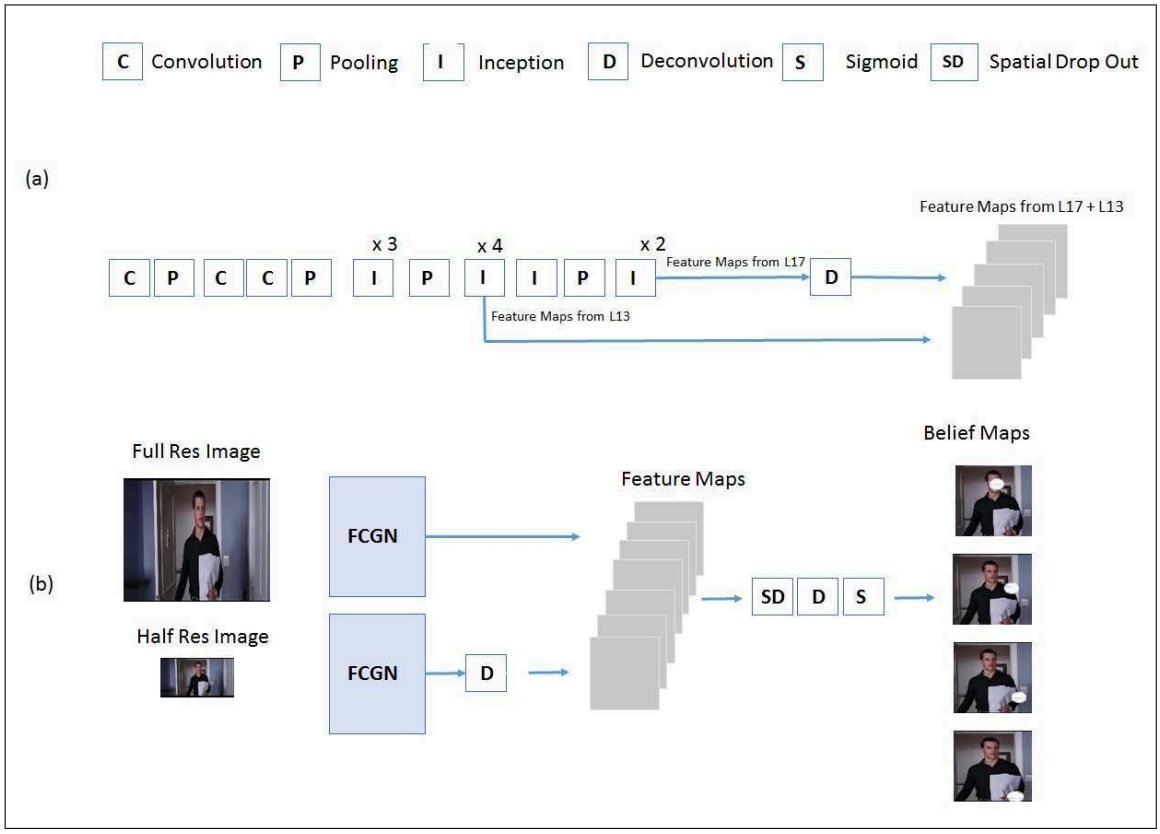


Figure 5.2: (a) Proposed fully convolutional encoder (FCGN), which is an adaptation of the batch-normalised inception network Ioffe and Szegedy (2015). (b) The proposed multi-resolution PoseNet1 combines two FCGNs. One takes the full resolution image as input and one takes a half resolution image as input.

PoseNet2. PoseNet2 is shown in Figure 5.3. PoseNet2 consists of a single FCGN that takes the Full resolution image only. Training at single resolution not only reduces memory footprint but also training time. All non-linearities in PoseNet2 are ReLU. Using ReLU inplace of ELU slight improves the performance as shown in Section 5.5. The pose decoder in PoseNet2 also consists of deconvolution filter of size 32×32 and stride 16. We do not use spatial drop out in PoseNet2. Comparison with PoseNet1 is shown in Table 5.2.

5.4.2 Data Augmentation

Data augmentation is an essential ingredient for deep networks and has a significant impact on performance. In contrast to image classification, we do not only transform

Features	PoseNet1	PoseNet2
Multi-Res	yes	no
Non-linearity	ELU	RELU
2D droput	yes	no
Feature concat	yes	yes

Table 5.2: Comparison of PoseNet1 and PoseNet2 architectures.

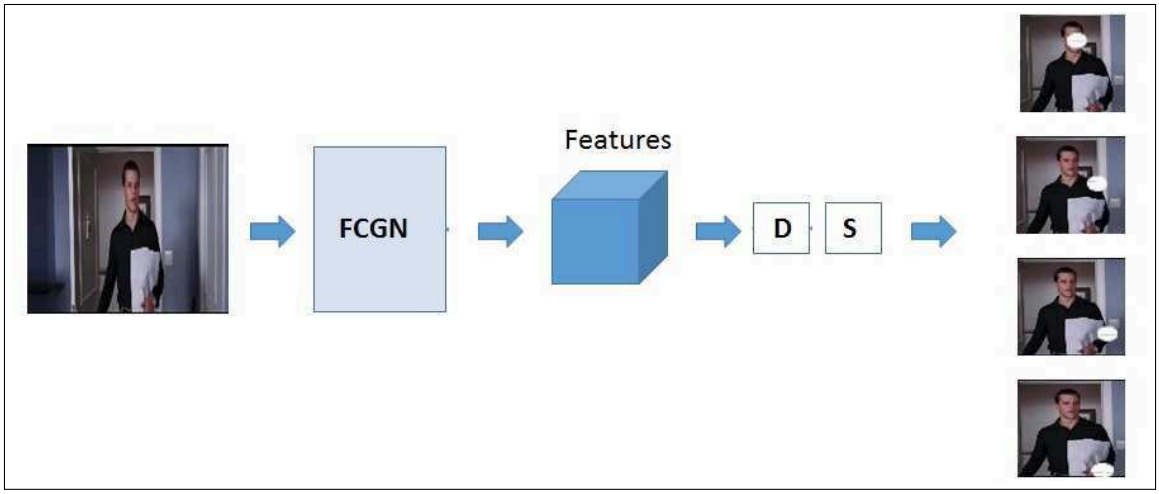


Figure 5.3: PoseNet2. It builds on the top of a single FCGN. The FCGN operates on a FullRes image only.

the image but also the joint annotations. We apply the following transformation to the training images:

$$W = \begin{pmatrix} 1 & 0 & c_x \\ 0 & 1 & c_y \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} rscos\theta & -ssin\theta & 0 \\ ssin\theta & scos\theta & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & -x_p + t_x \\ 0 & 1 & -y_p + t_y \\ 0 & 0 & 1 \end{pmatrix}, \tag{5.1}$$

The first transformation on the right hand side moves the person centred at (x_p, y_p) to the origin and applies a random translation (t_x, t_y) to it. The second transformation applies random scaling s, rotation θ and reflection r and the last transformation shifts the warped person back to the centre (c_x, c_y) of the cropped image. Figure 5.5 illustrates the procedure. For each training image, we generate and apply random transformations W. If one or more joints of the person are outside the image, we discard the transformation and replace it with another random transformation. Random augmentations are shown in Figure 5.4.

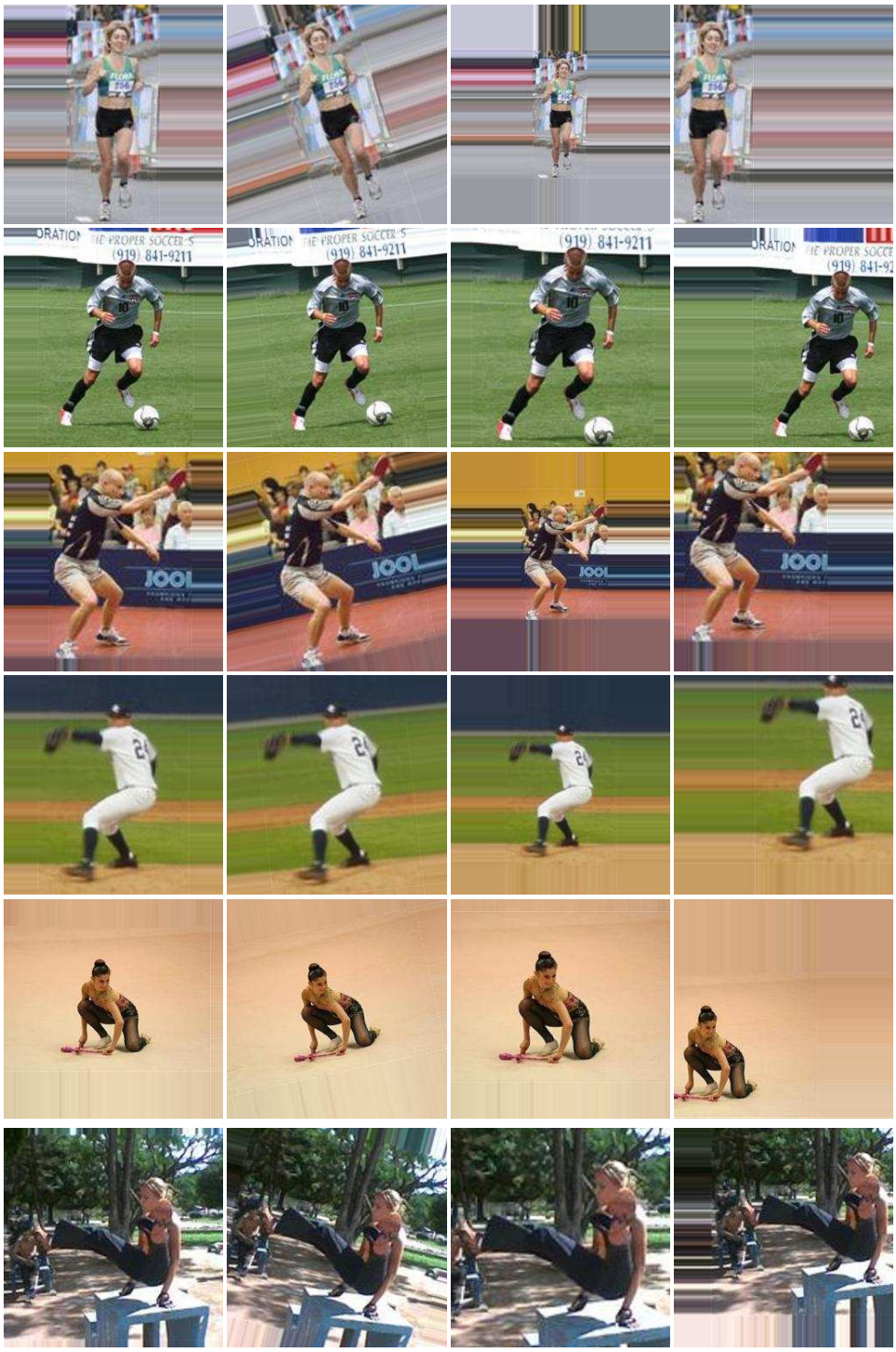


Figure 5.4: Data augmentation examples. (a) Orignal image. (b) Random rotation. (c) Random scaling. (d) Random translation.

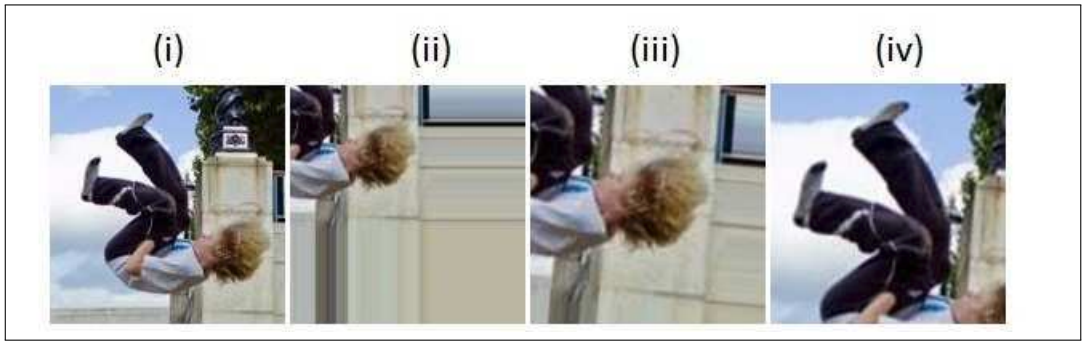


Figure 5.5: Illustration of data augmentation procedure : (i) Original Image (ii) Person shifted to upper left corner (iii) Person warped (iv) Person shifted back to center of cropped image.

5.4.3 Training

We denote a training example as $(I, \{b_j(x,y)\})$, where $b_j(x,y) \in \mathbb{R}^{w \times h}$ is the ground-truth 2D belief map for joint j and contains the per-pixel likelihood of joint j for each pixel (x,y) in image I of width w and height h. The belief map represents a binary image containing a disk of 1's with radius 'r' centred at the ground-truth (x,y) location of joint j and zero elsewhere, as shown in Figure 5.6. Given training samples $N = \{(I, \{b_j(x,y)\})\}$, we minimize the binary cross entropy between the ground-truth and predicted belief maps for k joints in each training image I as follows:

$$\arg\min_{w} \quad -\sum_{j=1}^{k} \sum_{x,y} b_j(x,y) \log(\hat{b}_j(x,y)) + (1 - b_j(x,y)) \log(1 - \hat{b}_j(x,y)), \quad (5.2)$$

where w and $\hat{b}_j(x,y)$ are the parameters of our network and the predicted belief maps, respectively. The weights of the network are then learned using back propagation and the Adam optimizer (Kingma and Ba (2014)).

5.4.4 Inference

At inference, the networks predicts a belief map $\hat{b}_j(x,y)$ for the joint j for every x, y in the image. The joint position is computed as:

$$x_p, y_p = \arg\max_{x,y}(\hat{b}_j(x,y)) \quad (5.3)$$

where x_p, y_p is the location of the peak in the predicted belief map $\hat{b}_j(x,y)$ for the joint j .

5.4.5 Test time Augmentation.

Test time data augmentation has been used successfully in image classification (Szegedy *et al.* (2015)) to slightly improve the perfromance. In this work, we apply test time

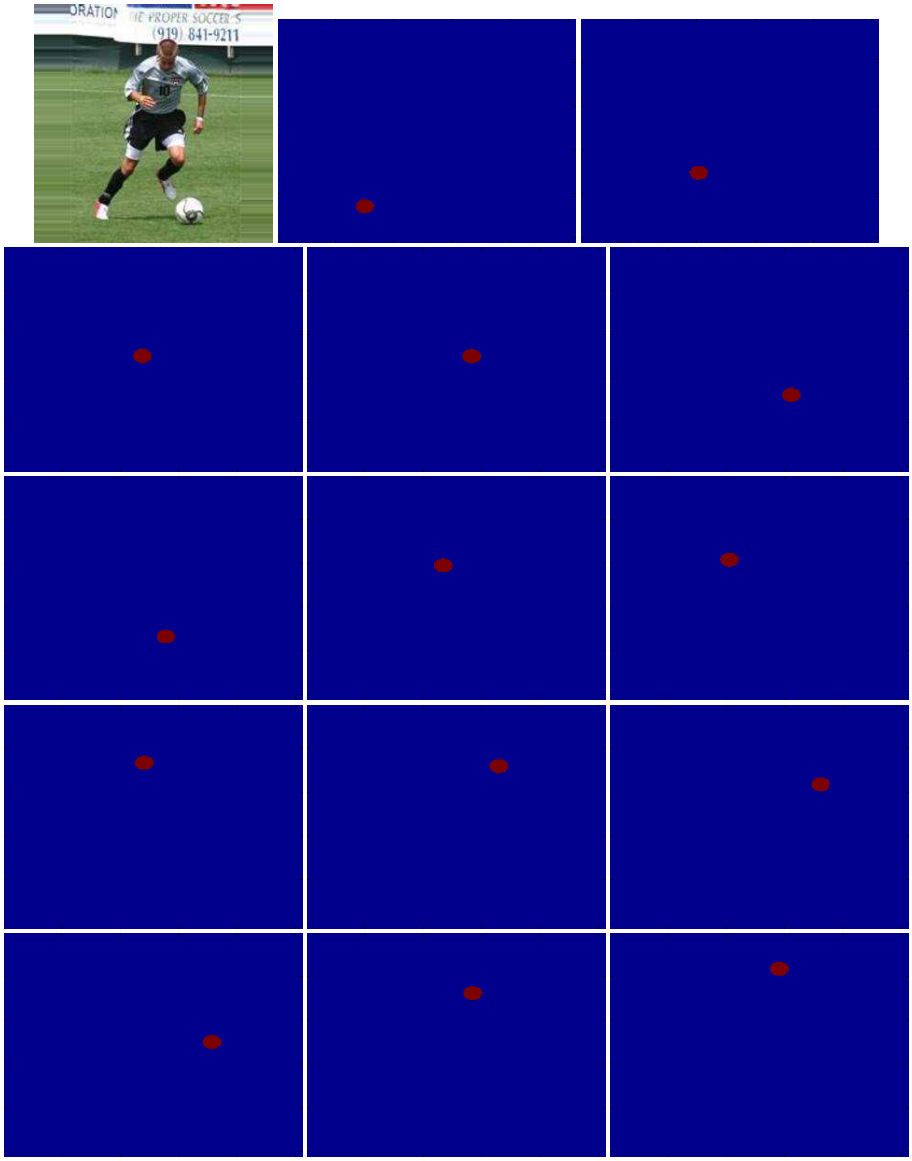


Figure 5.6: Image with ground-truth body joint belief-maps.

augmentation on the top of PoseNet2. We re-scale each test image to 192, 256, 288, 320, 384 resolutions respectively. We flip the rescaled images horizontally. An example is shown in Figure 5.7. We process these images with PoseNet2 and get multiple heatmaps at different resolutions for each body joint. We locate the peak in each heatmap and take the location of the max-scoring peak as the final prediction for each joint. We refer to test time augmentation as TTA.

Figure 5.7: An example of test time data augmentation. Row 1 shows a image rescaled to multiple resolutions. Row 2 shows each of the rescaled image flipped horizontally.

5.5 Experiments

We evaluate PoseNet1 and PoseNet2[4] on standard benchmarks for human pose estimation, namely the MPII Human Pose (MPII) dataset (Andriluka *et al.* (2014)), the Leeds Sports Pose (LSP) dataset (Johnson and Everingham (2010, 2011)) and the Frames Labelled In Cinema (FLIC) dataset (Sapp and Taskar (2013)). Some qualitative results are shown in Figures 5.9 and 7.13.

Our experimental settings are as follows: We crop the images in all datasets to a resolution of 256×256 pixels. For the training images, we crop around the person's centre, computed as the midpoint between maximum and minimum ground-truth joint positions in x, y directions. For the test images, we crop around the provided rough location when available and around the centre of the image otherwise. We train the network from scratch without any pre-training with a learning rate of 0.00092 using a stair case decay of 0.95 applied after 73 epochs, shown in Figure 5.8. We use a batch size of 8. For Adam, we use $\beta_1 = 0.9$ and $\epsilon = 0.1$. The ground-truth belief maps are created by setting all pixels within an 8 pixel distance to the annotated joint to 1. We train the network for 130 epochs for each dataset. For data augmentation, we transform each training image 130 times. The scaling parameter $s \in [0.5, 1.5]$, the translation

[4]PoseNet2 is evaluated on LSP and MPII benchmarks only.

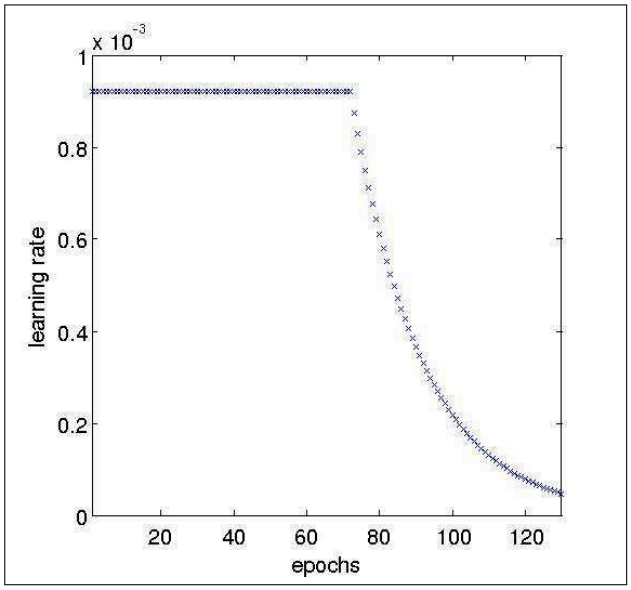


Figure 5.8: Learning Rate used during training.

parameters $t_{x,y} \in [-20, 20]$ and the rotation parameter $\theta \in [-20, 20]$ are randomly selected with uniform probability. Horizontal flipping r is applied with probability 0.5.

We use the Torch 7 (Collobert *et al.* (2011)) framework for training and evaluating PoseNet1 and PyTorch (Paszke *et al.* (2017)) for training and evaluating PoseNet2 in our experiments. Unless otherwise stated, we report results for the above-mentioned settings. For evaluation, we use the PCK measure (Sapp and Taskar (2013)) for the LSP and FLIC datasets and PCKh (Andriluka *et al.* (2014)) for the MPII dataset.

5.5.1 FLIC dataset

The FLIC (Sapp and Taskar (2013)) dataset consists of 3,987 training images and 1,016 test images. PoseNet1 takes 10 hours to train on the FLIC dataset using a GTX 980 GPU. Following the standard practice, we report PCK @ 0.2 only for wrists and elbows. Quantitative results are given in Table 5.3. PoseNet1 outperforms the other methods for elbows and wrists. PoseNet1 does not use any image dependent explicit prior model as compared to Chen and Yuille (2014).

5.5.2 Leeds Sports Pose (LSP) dataset

The LSP dataset (Johnson and Everingham (2010)) consists of 1,000 training and 1,000 test images. The extended LSP dataset (Johnson and Everingham (2011)) consists of an additional 10,000 training images. For our experiments, we use the 11,000 training images from LSP and extended LSP together. PoseNet1 takes 2 days for training using a GTX 980 GPU. PoseNet2 takes 10 hours for training. Quantitative results for PCK @ 0.2 are shown in Table 5.4.

Method	Elbows	Wrists
Tompson *et al.* (2015)	93.1	89.0
Toshev and Szegedy (2014)	92.3	82.0
Chen and Yuille (2014)	95.3	92.4
Wei *et al.* (2016)	97.6	95.0
PoseNet1	98.5	96.5

Table 5.3: Comparison over Quantitative Results for FLIC over Elbows and Wrists with state-of-the-art using PCK @ 0.2.

Method	Head	Shoulder	Elbow	Wrist	Hip	Knee	Ankle	PCK
Tompson *et al.* (2014)	90.6	79.2	67.9	63.4	69.5	71.0	64.2	72.0
Fan *et al.* (2015)	92.4	75.2	65.3	64.0	75.7	68.3	70.4	73.0
Carreira *et al.* (2016)	90.5	81.8	65.8	59.8	81.6	70.6	62.0	73.1
Chen and Yuille (2014)	91.8	78.2	71.8	65.5	73.3	70.2	63.4	73.4
Yang *et al.* (2016)	90.6	78.1	73.8	68.8	74.8	69.9	58.9	73.6
Wei *et al.* (2016)	.	.	.	.	.	.	.	84.3
Pishchulin *et al.* (2016) + MPII	97.0	91.0	83.8	78.1	91.0	86.7	82.0	87.1
Wei *et al.* (2016) + MPII	97.8	92.5	87.0	83.9	91.5	90.8	89.9	90.5
Bulat and Tzimiropoulos (2016)* + MPII	97.2	92.1	88.1	85.2	92.2	91.4	88.7	90.7
Chu *et al.* (2017)* + MPII	98.1	93.7	89.3	86.9	93.4	94.0	92.5	92.6
Yang *et al.* (2017)* + MPII	98.3	94.5	92.2	88.9	94.7	95.0	93.7	93.9
PoseNet1	95.8	86.2	79.3	75.0	86.6	83.8	79.8	83.8
PoseNet2 with TTA	96.9	86.8	81.2	76.5	87.4	85.5	82.6	86.3

Table 5.4: Comparison over Quantitative Results for LSP with state-of-the-art using PCK @ 0.2.

PoseNet1 achieves a higher accuracy than most of the other methods and gets very close to Wei *et al.* (2016)[5] when the same training data is used, which is consistent with the results on the FLIC dataset. PoseNet2, despite being faster, with TTA outperforms PoseNet1 and many of the other methods.

Performance is not directly comparable to methods listed with "*" since they pre-train on MPII Human Pose (Andriluka *et al.* (2014)) dataset and use cascades of CNNs for performance improvement. Wei *et al.* (2016) showed pre-training on MPI improves PCK by 6 percent.

[5]Wei *et al.* (2016) was the top performing method when PoseNet1 was published.

Method	Head	Shoulder	Elbow	Wrist	Hip	Knee	Ankle	PCKh
Hu and Ramanan (2016)	95.0	91.6	83	76.6	81.9	74.5	69.5	82.4
Carreira *et al.* (2016)	95.7	91.7	81.7	72.4	82.8	73.2	66.4	81.3
Tompson *et al.* (2015)	96.1	91.9	83.9	77.8	80.9	72.3	64.8	82.0
Pishchulin *et al.* (2016)	94.1	90.2	83.4	77.3	82.6	75.7	68.6	82.4
Wei *et al.* (2016) + LSP	97.8	95.0	88.7	84.0	88.4	82.8	79.4	88.5
Bulat and Tzimiropoulos (2016)*	97.9	95.1	89.9	85.3	89.4	85.7	81.7	89.7
Newell *et al.* (2016)*	98.2	96.3	91.2	87.1	90.1	87.4	83.6	90.9
Chu *et al.* (2017)*	98.2	96.8	92.2	88.0	91.3	89.1	84.9	91.8
Yang *et al.* (2017)*	98.5	96.7	92.5	88.7	91.1	88.6	86.0	92.0
PoseNet1	97.2	93.9	86.4	81.3	86.8	80.6	73.4	86.3
PoseNet2 with TTA	98.2	95.9	86.8	82.4	88.8	80.6	76.4	88.0

Table 5.5: Comparison over Quantitative Results for MPI with recent state-of-the-art using PCKh @ 0.5.

5.5.3 MPII Human Pose Dataset

The MPII Human Pose (Andriluka *et al.* (2014)) dataset is a challenging dataset and consists of around 40,000 images of people. We use 25,925 images for training and use 2,958 images for validation according to the train/validation split from Tompson *et al.* (2015). We evaluate on the 7,247 single person test images with withheld annotations. The dataset provides a rough scale and person location for both training and test images. We crop test images around the given rough person location. We normalise both training and test images to the same scale by using the provided rough scale information. PoseNet1 takes 3 days to train on the MPII dataset using a GTX 980 GPU. PoseNet2 takes 1 day for training.

Quantitative results are shown in Table 5.5. Both PoseNet1 and PoseNet2 outperform most of the recent state-of-the-art methods and are competitive with the top performing methods. Methods with "*" were published after PoseNet1. Performance is not directly comparable as they employ heavy architectures with extra CNN stages on the top.

5.5.4 Ablation Studies.

We study the effect of various design choices used in PoseNet1 and PoseNet2 on the LSP (Johnson and Everingham (2010)) and FLIC (Sapp and Taskar (2013)) datasets respectively.

Impact of scale normalisation. For the FLIC dataset, rough torso detections are available for the training and testing images. We thus normalise all training and test images to the same scale by re-normalising the height of the detected torso in each image to 200 pixels. The results reported in Table 5.6 show that using scale information, when

Figure 5.9: Images with heatmaps for all body joints over-laid. Rows 1-3 shows examples from the Leeds Sports Pose (LSP) dataset (Johnson and Everingham (2010, 2011)). Rows 4-6 shows examples from the MPII Human Pose (MPII) dataset (Andriluka *et al.* (2014)).

Method	Elbows	Wrists
PoseNet1	96.1	89.7
PoseNet1 with scale normalization	98.5	96.5
PoseNet1 with reduced augmentation	92.4	81.9
PoseNet1	94.5	86.5

Table 5.6: Comparison over various design choices for FLIC over Elbows and Wrists with state-of-the-art using PCK @ 0.2.

Method	Head	Shoulder	Elbow	Wrist	Hip	Knee	Ankle	PCK
PoseNet1 with single resolution only	96.0	85.8	79.6	73.7	86.3	80.8	77.7	82.8
PoseNet1 multi-resolution	95.8	86.2	79.3	75.0	86.6	83.8	79.8	83.8
PoseNet1 with feature maps from last layer only	95.4	83.7	74.8	70.4	84.4	78.2	74.2	80.2
PoseNet2	96.5	86.8	80.1	76.5	86.7	85.3	81.3	84.3
PoseNet2 with TTA	96.9	86.8	81.2	76.5	87.4	85.5	82.6	86.3

Table 5.7: Comparison over various design choices for LSP with state-of-the-art using PCK @ 0.2.

available, can provide significant gains in accuracy, especially for wrists from 89.7 to 96.5.

Impact of data augmentation. We evaluate the impact of data augmentation by reducing the ranges for scaling $s \in [0.7, 1.3]$, translation $t_{x,y} \in \{-5, 5\}$ and rotation $\theta \in \{-5, 5\}$ on the FLIC dataset. The results reported in Table 5.6 show that the accuracy drops from 96.1 to 92.4 for elbows and from 89.7 to 81.9 for wrists. This shows that extensive data augmentation is indeed important for achieving high accuracy and that the exact details of data augmentation are important for reproducibility.

Effect of exponential learning rate decay. We also compare an exponential decay of the learning rate with a constant learning rate using the same number of epochs. Without the learning rate decay, accuracy drops from 96.1 to 94.5 for elbows and from 89.7 to 86.5 for wrists as reported in Table 5.6.

Impact of middle layer features. We evaluate the impact of using feature maps from the middle layer on the LSP (Johnson and Everingham (2010)) datset. We remove the skip layer connection from layer 13 in PoseNet1. The accuracy drops for all joints and in average from 83.8 to 80.2 as shown in Table 5.7.

Impact of multi-resolution architecture. We compare our model that combines two FCGNs, one applied to the full resolution image and the second one to the half resolution image. If we use only one FCGN with the full resolution image for training and testing, the average accuracy decreases slightly from 83.8 to 82.8 as shown in Table 5.7. The slight decrease can be explained by the fact that the half resolution FCGN introduces more context for the prediction. As a result, the additional context only

Method	Forward pass time(ms)	Memory(Mb)
Yang *et al.* (2017)	92±5	1434±10
Newell *et al.* (2016)	36±5	1066±10
PoseNet1	17±5	826±10
PoseNet2	11±5	593±10

Table 5.8: Speed/Memory comparison of PoseNet1 and PoseNet2 with state-of-the-art methods, Newell *et al.* (2016) and Yang *et al.* (2017).

improves the prediction of the joints that are far away from the head, namely wrist, knee and ankle.

ReLU vs ELU. Table 5.7 shows PoseNet2 with ReLU non-linearities performs slighty better than multiRes and singleRes PoseNet1 . This shows selecting the right non-linearity is also important for performance.

Test time Augmentation. We study the impact of test time augmentation on the LSP (Johnson and Everingham (2010)) dataset. We augmented each test image as explained in Section 5.4.5. Comparison is shown in Table 5.7. PoseNet2 with TTA achieves 2 percent better performance than PoseNet2 without TTA.

5.5.5 Benchmarking

We benchmark PoseNet1 and PoseNet2 against the top performing methods, Newell *et al.* (2016) and Yang *et al.* (2017), in Table 5.8. We compare the speed/memory taken by PoseNet1 and PoseNet2 against the architectures from Newell *et al.* (2016) and Yang *et al.* (2017). We use batch size of 1 and image of resolution 256×256 respectively. The benchmarking is done on GTX Titan. PoseNet2 is the fastest with a low memory footprint yet it performs very competitive.

5.6 Application: Frame-by-Frame Single Person Pose Estimation in Indoor Scenarios.

Figure 5.10: (Left) Person is using copy machine. (Right) Person is washing the cup.

PoseNet2 has been successfully deployed for single person body pose estimation in the European projects STRANDS (Hawes (2016)). The aim of the projects was to integrate state-of-the-art robotics and perception methods on a robot in order to make it fully autonomous. The robot was deployed in indoor environments for periods of days to months and was programmed to perform user defined tasks. One task was higher level activity recognition as shown in Figure 5.10[6]. The task required the accurate positions of 2d body joints of the person and objects in the scene . The robot was programmed to use this information to infer the high level activity.

PoseNet2 was deployed on the robot to estimate the 2d body joints positions. We used the tracker from Linder *et al.* (2016) to get the person bounding box as shown in 5.11 (left). We cropped the image around the bounding box and processed it with PoseNet2 to get 2d postions of body joints. Example is shown in Figure 5.11 (right). Qualitative results are shown in Figure 5.12. PoseNet2 even predicts the occluded joints correctly.

PoseNet2 ran at 6 frames/second on GTX1050, a mid range GPU, that was deployed on the robot.

Figure 5.11: (left) Bounding box predicted by the tracker from Linder *et al.* (2016). (right) Body pose estimated by PoseNet2 on the crop.

[6]Faces are blurred to preserve privacy.

Figure 5.12: Qualitative results for human pose estimation in indoor scenarios. PoseNet2 predicts the body joint positions correctly even for occluded joints.

Figure 5.13: Images with skeletons are shown for examples from the MPII Human Pose (MPII) dataset (Andriluka *et al.* (2014)).

5.7 Conclusion

In this work, we have proposed a convolutional neural network with a low memory footprint for human pose estimation. The network can be trained efficiently on a mid-range GPU. It achieves competitive performance on popular benchmarks for human pose estimation, which is impressive since the model does not require any pre-training on large datasets as other models and can be trained from scratch also on small datasets like FLIC. The proposed network, which is publicly available, can serve as a baseline for more complex models in the future.

The proposed architecture assumes person's location and scale are available before hand. Exploring an end-to-end method that jointly estimates location, scale and body joints is an interesting direction for future work.

6

Efficient Bottom Up Multi-person Pose Estimation with Bi-directional Relative Offsets Fields

This chapter is concerned with bottom up multi-person pose estimation. Recently, significant success in single person pose estimation has been achieved with deep learning. This has led researchers to focus on the challenging task of unconstrained multi-person pose estimation. Current work on multi-person pose estimation can be categorized into top down approaches and bottom up approaches. Top down approaches first detect persons in the image and then estimate their body pose. Bottom up approaches first detect person agnostic body joints and then perform joints-to-person association.

Existing bottom up approaches require graph based post-processing to perform joints-to-person association. Graph based post-processing is expensive and does not benefit from end-to-end learning. Towards this goal, we propose a bottom up multi-person pose estimation framework that does not require any graph-based post-processing on the top. Our framework outputs novel bi-directional relative offset fields for neighbouring body parts along with part detection heat-maps. The relative offsets fields densely predict relative offset vectors to neighbouring body parts. The relative offsets encode spatial dependencies between neighbouring body parts and provide a strong cue for associating body parts. The predicted relative offset vectors are combined with predicted body parts with a simple greedy procedure to associate predicted body parts to persons in the image. Our approach performs comparably to the recent state-of-the-art method for multi-person pose estimation on the MS COCO dataset with a superior run-time on mid-range graphics processing unit.

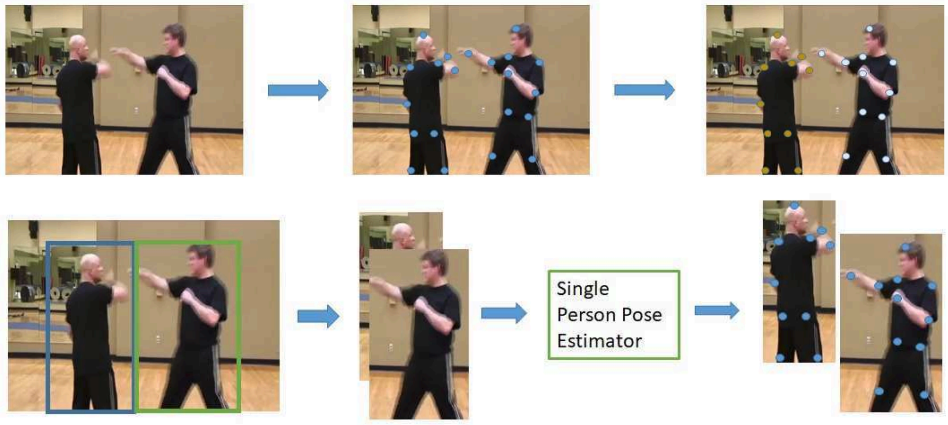


Figure 6.1: Multi-person Pose Estimation. (Top) Bottom-up multi-person pose estimation. (Bottom) Top-down multi-person pose estimation.

6.1 Introduction

Deep learning has achieved remarkable success for single person pose estimation. However, recent deep learning approaches assume there is only single person in the image and the location of the person is known as a priori. The assumptions are too strong for real life scenarios where images contain multiple people and existing deep learning single person pose estimation methods cannot be applied there. These limitations have pushed researchers to work on the challenging task of unconstrained multi-person pose estimation (Pishchulin *et al.* (2016), Insafutdinov *et al.* (2016), Cao *et al.* (2017), Newell *et al.* (2017), Papandreou *et al.* (2017b), He *et al.* (2017)).

The current research on multi-person pose estimation follows two paradigms shown in Figure 6.1. Top down multi-person pose estimation follows a two stage pipeline. In the first stage, a person detector is used to detect persons in the image. In the second stage, single person pose estimation is applied to the detected persons. Top down approaches may suffer from severe people overlap or wrongly localised bounding boxes. This has led researchers to propose bottom up multi-person pose estimation where the goal is to first detects all body parts in the image and then associate them to persons in the image.

In this chapter, we focus on bottom up multi-person pose estimation. Despite impressive performance, bottom up approaches require additional graph based post-processing to associate body parts to persons in the image and do not benefit from end-to-end learning. The form of association used also effects the over all performance. Cao *et al.* (2017) densely predict unit vectors to neighbouring body parts on a limb and then use them to assign weights to edges on the graph. Insafutdinov *et al.* (2016) separately learn logistic regression classifiers for associating pair of body parts in the image and then use the score predicted by the classifier as edge weights in the graph. Since the used associations are weak, expensive graph based processing is needed for optimal association.

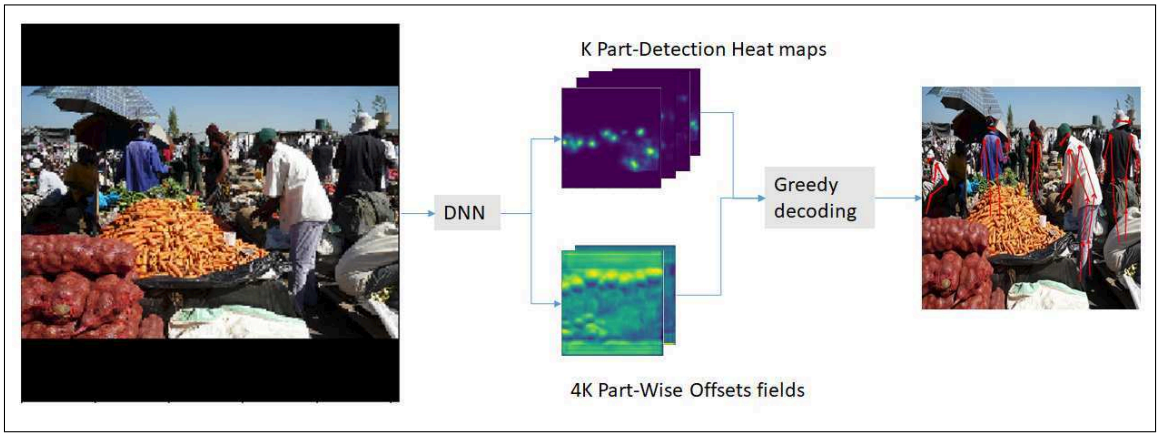


Figure 6.2: Our framework for bottom up multi-person pose estimation. The framework produces two outputs. (i) Part detection heat-maps. (ii) Bi-directional relative offsets fields. The detected body parts are used with predicted relative offsets fields to associate body parts to persons in the image by a simple greedy procedure.

We propose a bottom up multi-person framework that densely predicts bi-directional relative offsets for pairs of neighbouring body parts on the human body along with body part detection heat-maps as shown in Figure 6.2. The relative offsets encode spatial dependencies between neighbouring body parts and provide a strong association signal for connecting neighbouring parts. For efficient association, we assume a tree structured skeleton hierarchy on the body parts connectivity as shown in Figure 6.3. During inference, we combine the predicted relative offsets with the predicted part detection heat-maps with a simple and efficient greedy procedure for associating body parts to persons in the image thus making the graph-based post-processing obsolete. We evaluate the proposed method on the challenging MS COCO dataset and achieve comparable performance to state-of-the-art bottom up approaches. Our methods runs twice as fast than the graph based bottom up approaches on mid-range graphics processing units.

6.2 Contributions

In this chapter we make the following contribution :

- We propose a bottom up multi-person pose estimation framework with novel bi-directional relative offsets fields. The relative offsets fields densely predict the relative offsets vectors to neighbouring body parts. The relative offsets vectors encode spatial dependencies between neighbouring body parts and provide a strong cue for associating detected body parts.
- The predicted relative offsets vectors are combined with the predicted body part detection heat-maps by a simple greedy procedure to associate detected body parts to persons in the image.

Figure 6.3: (Left) Tree structure skeleton hierarchy used to for associating neighbouring body parts. (Right) Bidirectional offsets vectors

- The proposed approach is evaluated on the MS-COCO dataset and performs comparably to the recent state-of-the-art bottom up methods with a superior run time on mid-range graphics processing units.

6.3 Related Work

The current research on multi-person pose estimation can be categorised into top down and bottom up methods respectively. The bottom up methods first detect all the body parts in the image. Graph based methods are applied as a post-processing step for associating body parts to persons. The top down methods first detect bounding boxes around all the persons in the image. Single person pose estimation is applied to each bounding box to estimate body pose.

Bottom Up Multi-person Pose Estimation. Pishchulin *et al.* (2016) first proposed a bottom up approach for multi-person pose estimation. Their method first proposed a number of proposals for each body part in the image. A densely connected graph is constructed on the top of proposals. Linear programming optimization is applied to the graph to assign body parts proposals to persons. Despite suitable performance, the approach takes 72 hours to perform inference on a single image. Insafutdinov *et al.* (2016) improved the work from Pishchulin *et al.* (2016). They use expensive CNN architecture to improve detection of body parts. In addition to that, they used an incremental optimization procedure to speed up the inference procedure to 8 minutes. Iqbal and Gall (2016) further speeded up the inference by solving the inference locally for each person.

Cao *et al.* (2017) jointly predict the body parts detection and pair-wise affinity heat maps. They use a greedy approach on the top for body-parts associations. They construct a graph that assumes a tree structure on the body parts connection and is faster to optimize. Their approach reduces the per-image inference time to 0.005s.

All the bottom up methods for multi-person pose estimation require graph based post-processing for associating body parts to persons. Newell *et al.* (2017) proposed associative embeddings to solve the association problem. The proposed to jointly predict the body parts detection and tag maps. The tag maps consists of dense 1D tags for every pixel in the image. The idea is to predict similar tags for body parts belonging to the same person. During inference, mean shift clustering is applied to predicted tags to associate body parts to the person.

Top Down Multi-person Pose Estimation. Papandreou *et al.* (2017b) proposed a two stage pipeline for multi-person pose estimation. They first use state-of-the-art people detector Ren *et al.* (2015) to generate proposals for people detection in the image. They perform non-maximal suppression to get rid of noisy proposals. The final set of proposals are passed to a single person pose estimator. The approach also predicts relative offsets maps for each body part. The relative offsets maps densely predict relative offsets for every pixel to every body part belonging to the person.

He *et al.* (2017) extended the work from Ren *et al.* (2015) to predict body parts detection for every detected bounding box. In comparison to Papandreou *et al.* (2017b), their approach reuses the existing features for body parts predictions.

Chen *et al.* (2018) proposed a novel single person pose estimator employed in top down approaches. The proposed estimator consists of a global net and a refinement net. The global net predicts body parts detection at every stage of a convolutional neural network. The refinement net produces refined detections on the concatenated features from different stages of the CNN.

Similar to (Pishchulin *et al.* (2016), Insafutdinov *et al.* (2016), Cao *et al.* (2017), Iqbal and Gall (2016), Newell *et al.* (2017)), our approach also predicts person agnostic body joints proposals and then associates them. However, we additionally predict pairwise relative offsets and use a simple greedy procedure to associate different body joints.

6.4 Our Framework

Our framework is shown in Figure 6.2. It consists of a single resolution FCGN as the feature generator. Given an input image I the feature generator produces low resolution features $F \in R^{c \times h \times w}$ with channels c, height h and width w respectively. The features F are processed by a body parts detector and a offsets fields generator to produce part detection heat-maps and relative offsets fields respectively.

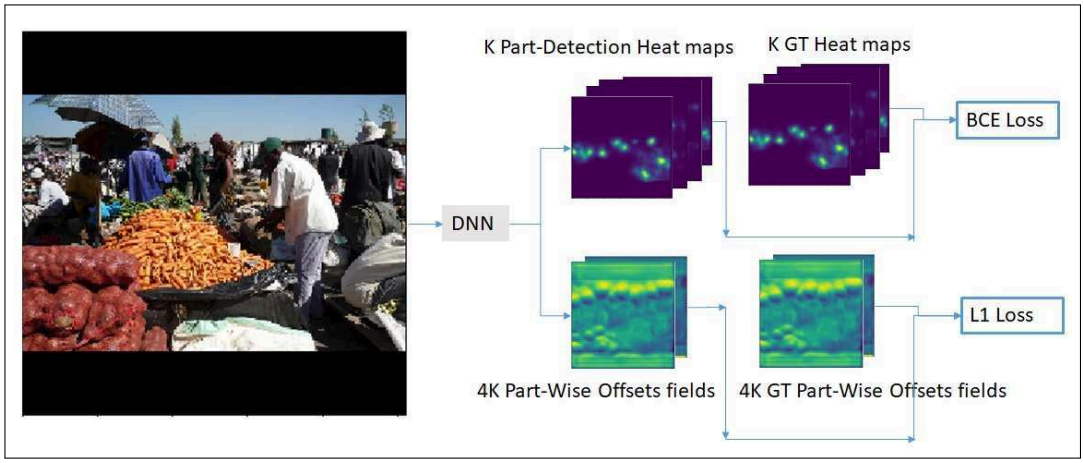


Figure 6.4: Training Procedure. For part detection heat-maps we optimize the BCE loss. For relative offsets fields we optimize L1 loss.

6.4.1 Body Parts Detector

The body parts detector outputs k part detection heat- maps $\hat{B} \in R^{k \times h \times w}$ where k is the number of body parts. Given the ground truth body part positions x_{kp} for a body part k and a person p in the image, the value at a location $l(x, y)$ in the ground truth heat map B_{kp} is computed as :

$$B_{kp}(l) = \exp(-\frac{\|l - x_{kp}\|_2^2}{\sigma^2}) \tag{6.1}$$

The ground truth map for a body part k for all instances is then computed as :

$$B_k(l) = \max_p B_{kp}(l) \tag{6.2}$$

We take the maximum instead of sum where the two instances overlap.

6.4.2 Relative Offsets Fields Generator

The relative offsets fields generator generates bi-directional relative offsets fields $\hat{O}_x \in R^{k \times h \times w}$ and $\hat{O}_y \in R^{k \times h \times w}$ respectively as shown in Figure 6.3 (Right). $\hat{O}_{xj}$ contains x offsets and $\hat{O}_{yj}$ contains y relative offsets to the body part j_p of person p from the pixels that are within a radius of R pixels of the neighbouring body part $n(j_p)$ of j_p in the skeleton hierarchy as shown in Figure 6.3. The ground truth offsets maps O_{xj} and O_{yj} for a location $l(x, y)$ are computed as :

$$O_x p(l) = \begin{cases} x_{jp} - l_x, & |l_x - x_{l(x_{jp})}| \leq R \\ 0, & \text{otherwise} \end{cases} . \tag{6.3}$$

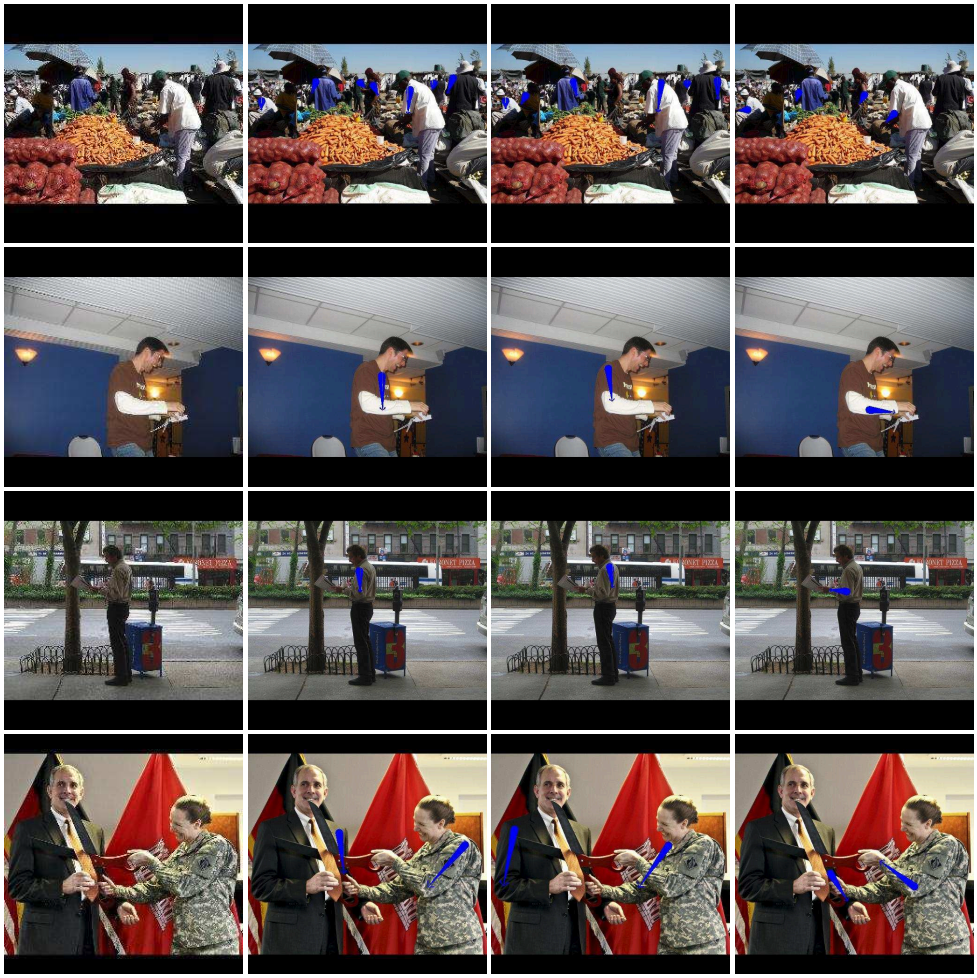


Figure 6.5: Relative Offsets for different limbs.

$$O_y j(l) = \begin{cases} y_{jp} - l_y, & |l_y - x_{l(y_{jp})}| \leq R \\ 0, & \text{otherwise} \end{cases} . \tag{6.4}$$

Examples are shown in Figure 6.5.

6.4.3 Training

The training procedure is shown in Figure 6.4. For part detection heat-maps we use binary cross entropy loss since the predicted heat-maps have scores between 0 and 1. We use L1 regression loss for relative offsets fields as it is a continuous regression problem. Given ground truth body parts and relative offsets maps respectively, we optimize the following loss :

$$L = L_J + L_O \tag{6.5}$$

where

$$L_J = \sum_j \sum_p \|\hat{B}_j(p) - B_j(p)\|_2^2 \tag{6.6}$$

$$L_O = \sum_{c=\{x,y\}} \sum_j \sum_p \|\hat{O}_{cj}(p) - O_{cj}(p)\| \tag{6.7}$$

The loss is optimised using back propagation and the adam optimiser.

6.4.4 Inference

Given part detection heat-maps $\{\hat{B}\}_j$ and relative offsets maps $\{\hat{O}_x, \hat{O}_y\}_j$ we first get a set of m top scoring detections $\{\{D\}_m\}_j$ from every part detection heat-map j. We then use a greedy decoding procedure to associate top scoring detections to persons. The procedure is described in Algorithm 1.

Algorithm 1 Greedy Decoding Procedure

procedure DECODING
  Inputs : $\{\{D\}_m\}_j$, $\{\hat{O}_x, \hat{O}_y\}_j$
  Output : $S_{all} = \{\}$
  do
    $S = \{\}$
    Get the max scoring part detection d_m from $\{\{D\}_m\}_j$
    $S \cup d_m$
    Remove d_m from $\{\{D\}_m\}_j$
    do
      Read offsets O from d_m to the neighbouring body part in the $\{\hat{O}_x, \hat{O}_y\}_j$
      Next body part $n = d_m + O$
      if $dist(n, \{\{D\}_m\}_j) \leq R$ **then**
        Remove n from $\{\{D\}_m\}_j$
      end if
      $S \cup n$
      $d_m \leftarrow n$
    while All parts in the skeleton hierarchy are found
    $S_{all} \cup S$
  while $\{\{D\}_m\}_j \neq \{\}$
  return S_{all}
end procedure

The procedure finds the maximum scoring peak in all the part detection heat-maps. Then it finds the remaining body-parts peaks for this detected peak in the tree structured skeleton hierarchy by using the relative offsets fields. All the associated body parts

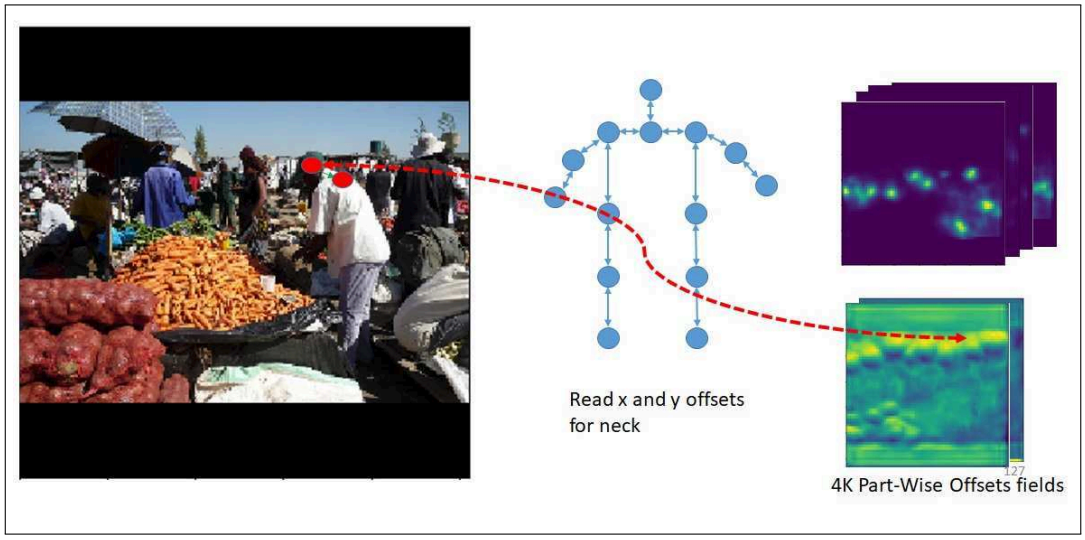


Figure 6.6: Training Procedure. For part detection heat-maps we optimize the BCE loss. For relative offsets fields we optimize L1 loss.

peaks are then removed from $\{\{D\}_m\}_j$.The procedure is repeated until all the peaks in $\{\{D\}_m\}_j$ are associated. An example is shown for head to neck association in Figure 6.6.

6.5 Experiments

In this chapter we perform experiments on the MS COCO dataset (Lin *et al.* (2014)). MS COCO is the largest dataset for multi-person pose estimation. The dataset has 115k training and 5k validation images. The images represent various everyday environments in totally uncontrolled settings. The dataset contains richer annotations including mask, bounding box and skeleton annotations for every person in the image. The skeleton annotations contains location of 17 body parts. Precise visibility flags are also provided. The dataset is challenging as person's scale and location is not available at test time. For testing, test-dev and test-std sets are provided. The ground-truth annotations for both sets are held out.

Training Protocol. We train for a total of 200 epochs. We use a batch size of 16. We use a learning rate of $2e-4$ dropped to $1e-5$ after 150 epochs. For data augmentation, we transform each training image 130 times. The scaling parameter $s \in [0.5, 1.5]$, the translation parameters $t_{x,y} \in [-20, 20]$ and the rotation parameter $\theta \in [-20, 20]$ are randomly selected with uniform probability.

We report results on the MS COCO test-dev set. We train on the 60000 training images for the Person Category. We compare our method with the state-of-the-art approaches in Table 6.1. Our method is trained from scratch. The results are evaluated using the OKS (Lin *et al.* (2014)) evaluation measure. Our approach performs compa-

	OKS	Speed (ms)	ImageRes	Backbone	PeopleScale
Bottom UP Methods					
Cao *et al.* (2017)	61	100	380×600	CPM	Variable
Newell *et al.* (2017)	66.3	150	512×512	HG	Variable
Papandreou *et al.* (2018)*	68.7		1400×1400	Deeplab	Variable
Ours(FCGN)	60	50	512×512	FCGN	Variable
Ours(GT)	90	50			
Top Down Methods					
Papandreou *et al.* (2017a)	65		368×368	ResNet	Fixed
Chen *et al.* (2018)*	73		368×368	ResNet	Fixed
Xiao *et al.* (2018)*	73.7		368×368	ResNet	Fixed

Table 6.1: OKS and Run time Comparison over MS-COCO (Lin *et al.* (2014)) test-dev set with recent state-of-the-art methods for multi-person pose estimation.

rably to state-of-the-art methods at the time with a faster run time. Methods with * have been published recently.

Qualitative Results are shown in Figure 6.8. Part detection heat-maps and relative offsets fields are shown in Figures 6.7 and 6.9 respectively.

6.6 Discussion

In this section, we discuss the effect of various factors on the performance for multi-person pose estimation.

Backbone architecture. The CNN used for generating the features is termed as the backbone. In our case FCGN from Chapter 5 is the backbone. Having a stronger backbone improves significantly the performance as shown in Table 6.1. To push the state-of-the-art, the other bottom up approaches (Cao *et al.* (2017), Newell *et al.* (2017), Papandreou *et al.* (2018)) use stronger backbones, CPM, HG, and Deep Lab respectively. However, these backbones require high-end graphics processing units and have a slower run-time compared to our method.

Image Resolution. Size of images also effect the performance of bottom up approaches, in particular for small people in the background. Training on larger image resolution boosts the performance since it magnifies small people in the image and results in better detection of small body parts like hands and feet. Interested reader is referred to Papandreou *et al.* (2018)) for a detailed analysis on the effect of image resolution on bottom up multi-person pose estimation.

People Scale. Currently top down approaches dominate bottom up approaches as they are trained and evaluated on a fixed scale. The scale of the cropped person is normalised to a fixed scale. The scale for the cropped person is provided by the bounding box from

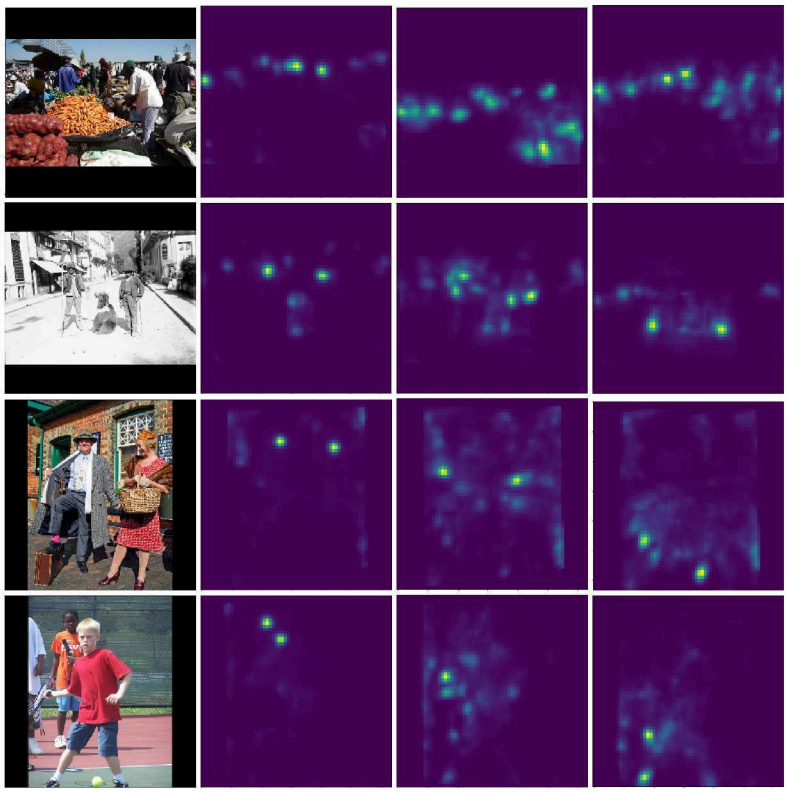


Figure 6.7: Examples of predicted Part Detection heat-maps on the MS COCO (Lin *et al.* (2014) test-dev set.

the person detector. In bottom up approaches, the model is trained to detect body parts at various scales which is a difficult task as compared to detecting body parts at a fixed scale.

6.7 Conclusion

In this chapter, we have proposed an efficient approach for bottom up multi-person. Our approach outputs novel relative offsets fields in addition to part detection heat-maps. The relative offsets fields predicts relative offsets to neighbouring body parts. The predict relative offsets encode spatial dependencies between neighbouring body parts and provide strong cue for associating body parts. The predicted relative offsets are combined with the predicted part detection heat-maps by a simple greedy procedure. The predicted relative offsets make the expensive graph-based post-processing obsolete. Our approach performs comparably to existing state-of-the-art bottom up approaches and runs faster on mid-range graphics processing units.

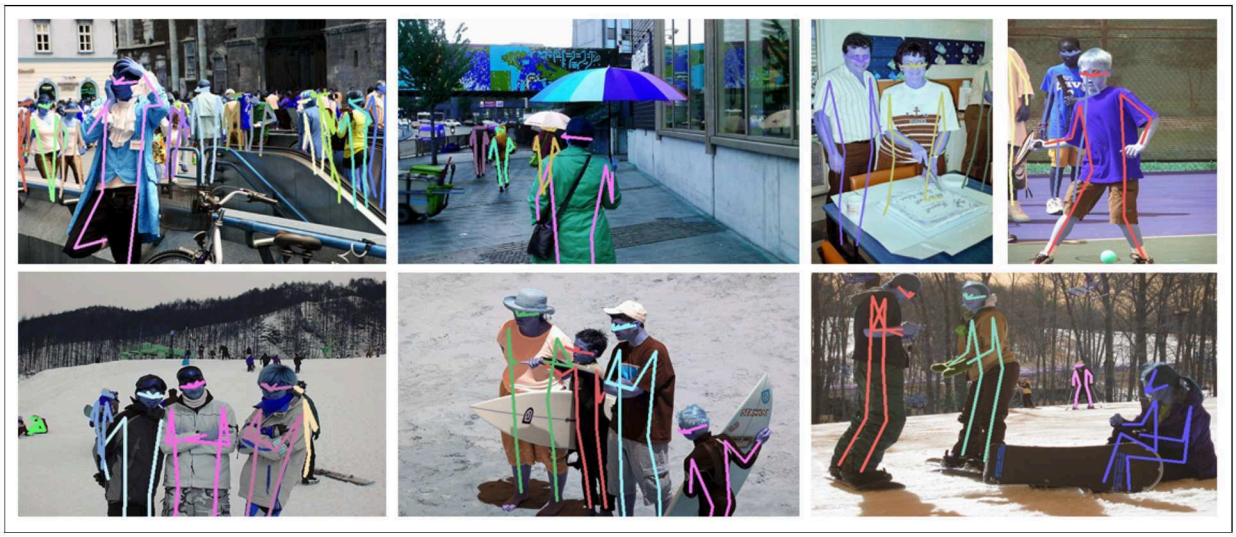


Figure 6.8: Qualitative Results on the MS COCO (Lin *et al.* (2014)) test-dev set. Our method associates the body parts correctly for occluded people.

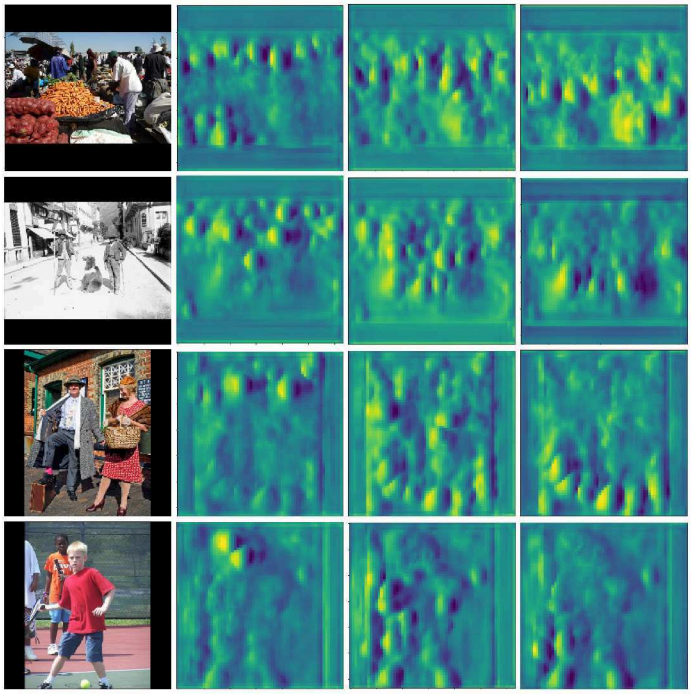


Figure 6.9: Examples of predicted relative offsets fields on the MS COCO (Lin *et al.* (2014) test-dev set.

7

An Efficient Framework for End-to-End Correspondence Matching

In this chapter, we focus on the task of correspondence matching. Correspondence matching is a fundamental problem in computer vision and has applications for many vision tasks such as stereo reconstruction, optical flow estimation, image retrieval, object tracking, and multi-person tracking. Variants of the task can range from finding exact matches, e.g., in stereo matching, to finding semantic correspondences, e.g., associating corresponding body parts of a person in different images.

We propose a fully supervised end-to-end learning method for the task of correspondence search. Our method consists of a novel matching network that efficiently generates peaked similarity heatmaps over the target image for all the query keypoints in a single pass. The method is end-to-end trainable with a multi-class classification loss. Since the loss is computed over the heatmaps, any wrong correspondence among two images contributes to the loss without the need of sampling a subset of hard-negative wrong matches.

We apply the proposed method to three variants of the correspondence matching, namely, semantic keypoints matching, dense matching and instance keypoints matching and achieve state-of-the-art performance on standard benchmarks for each variant. We release the code for the method to ensure reproducibility of work.

7.1 Introduction

Correspondence matching is a fundamental problem in computer vision and has applications for many vision tasks such as stereo reconstruction, optical flow estimation, image retrieval, object tracking, and articulated tracking. Variants of the task can range from finding exact matches, e.g., in stereo matching, to finding semantic correspondences, e.g., matching corresponding body parts of different species of birds. In this work, we focus on three variants, namely, semantic keypoints matching, dense matching and instance keypoints matching

Earlier work on correspondence matching relied on hand-crafted features such as SIFT (Lowe (2004)) or SURF (Bay *et al.* (2008)). Recently, convolutional neural networks (CNNs) have replaced hand-crafted features. In the context of correspondence search, Siamese networks have been proposed which achieve impressive results (Choy *et al.* (2016); Han *et al.* (2017); Kim *et al.* (2017); Schroff *et al.* (2015); Zagoruyko and Komodakis (2015)).

However, in recent work for the task of image-to-image semantic keypoints matching either a matching framework (Ham *et al.* (2016), Liu *et al.* (2011), Kim *et al.* (2013)) is applied over the descriptors obtained from a CNN (Kim *et al.* (2017)) or (Choy *et al.* (2016)) metric learning is applied on top of features learned by the deep network. The network is not trained for the end goal of precise localization of semantic key points.

Towards this end, we propose a novel matching network that efficiently generates a similarity heatmap for a every semantic query point in the source image over the target image in a single forward pass. The matching network shown in Figure 7.1 has no trainable parameters and can be appended to any standard deep network architecture to make it end-to-end trainable for the task of semantic keypoints matching. The peaks in the heatmaps then define the corresponding locations of the semantic keypoints in the target image.

Despite impressive performance achieved by recent deep learning based approaches (Choy *et al.* (2016); Han *et al.* (2017); Kim *et al.* (2013)) for image-to-image semantic keypoints matching the impact of the features from various layers of a deep network has not been well studied. For example, UCN (Choy *et al.* (2016)) uses inception layer 4a of the GoogleNet architecture as feature generator and uses a feature transformer layer on top of it for matching, while (Kim *et al.* (2017)) uses various layers of the VGGNet as input for self-similarity layers. To date, it is unclear that improving the features or adding follow-up layers, for example spatial transformer layer, is crucial for performance.

In this work, as a second contribution we analyse the effect of the features from various layers of a deep network on the task of image-to-image semantic keypoints matching. Towards this end, we use different variants of the batch-normalized GoogleNet architecture (Ioffe and Szegedy (2015)) in combination with the novel matching network to make each variant end-to-end trainable. Each variant differs in the pooling and number of layers used and therefore provide various trade-offs between resolution and captured semantics.

We evaluate the performance of explored variants on the PF-Pascal (Ham *et al.* (2016)) dataset and show that even small modifications to a varaint e.g adding another layer to learn better semantics can have a substantial impact on the accuracy for image-to-image semantic keypoint search. Interestingly, the combination of best performing variant with the novel matching network achieves state-of-the-art performance on all datasets.

7.2 Contributions

In this chapter we make the following contributions :

(i) We propose a novel matching network that efficiently generates a similarity heatmap for every semantic query point in the source image over the target image in a single forward pass. The matching can be appended to any backbone network for feature generation.

(ii) We analyse the effect of the features from various layers of a deep network on the task of image-to-image semantic keypoints matching by using different variants of the batch-normalized GoogleNet architecture in combination with the novel matching network.

(iii) We evaluate the proposed approach on standard benchmarks for semantic keypoints matching, dense matching and instance keypoints matching and achieve state-of-the art performance.

7.3 Related Work

Correspondence search is one of the fundamental problems in computer vision and has generated a lot of literature. Early work on correspondence search focused on using hand-crafted features such as SIFT (Lowe (2004)), SURF (Bay *et al.* (2008)), or DAISY (Tola *et al.* (2010)).

In recent years Siamese based CNNs have been used extensively for similarity related tasks. They were first used by (Bromley *et al.* (1994)) for signature verification. In (Zagoruyko and Komodakis (2015)) a Siamese network is used to measure patch similarity. Other Siamese networks have been used in (Schroff *et al.* (2015)) to learn face embeddings for super-human face identification performance. Zbontar and LeCun (2015) used them for stereo matching. Long *et al.* (2014) analyzed a CNN pre-trained on ImageNet for the task of semantic correspondence search. In (Wang *et al.* (2014)) a CNN is trained with triplet loss for fine-grained image ranking. Luo *et al.* (2016) trained a Siamese network with a dot product layer and multi-class classification loss for efficient disparity estimation.

Despite impressive performance, all of the above-mentioned approaches estimate patch-patch or patch-image similarity and require multiple forward passes during training and inference for matching similarity of multiple key-points.

Recently approaches that perform image-image semantic keypoints matching have been proposed . Choy *et al.* (2016) introduced a "Universal Correspondence Network" with spatial transformer layers (Jaderberg *et al.* (2015)). Their network is trained efficiently using metric learning and requires a single forward pass at inference time for

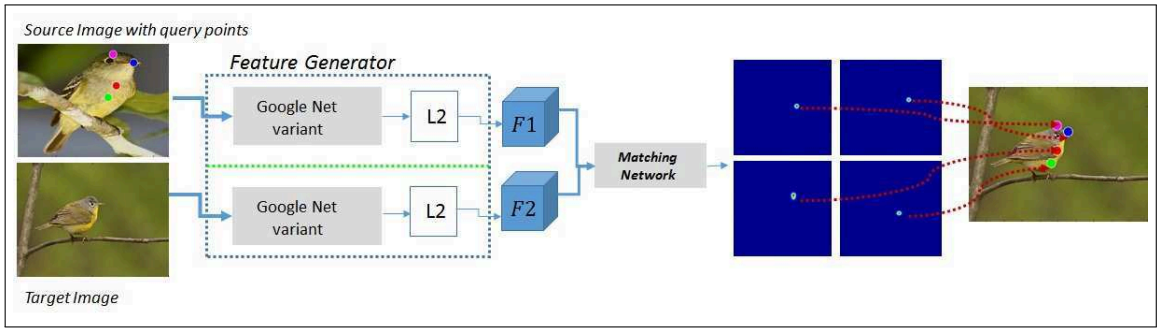


Figure 7.1: Our end-to-end learning framework consists of two parts: (i) a feature generator and (ii) a matching network. The feature generator takes the source image, the query points, and the target image and generates dense $\mathcal{D}$ dimensional features F_1 and F_2. The matching network takes features F_1 and F_2 as inputs and generates a peaked similarity map for each query point.

matching multiple key points. However, due to metric learning it requires an extra hard negative mining step that introduces the extra distance measure and k-nearest neighbor hyper-parameters. In contrast, we present an end-end learning framework that does not require the extra hard negative mining step. Kim *et al.* (2017) proposed a fully convolutional self-simalrity descriptor for dense semantic keypoints matching. However, their approach still requires a matching framework e.g (Ham *et al.* (2016)) on the top to establish correspondences. In contrast, our system is self contained and predicts the correspondences directly. Han *et al.* (2017) recently proposed a system that matches region proposals across a pair of images using appearances and geometry. Our system uses appearances only and does not require region proposals.

The approach from (Long *et al.* (2014)) is similar to our work as it also explores the effect of features at various layers on the task of patch-patch correspondence matching. In contrast our approach studies this effect for the task of image-image semantic key point matching. Our matching network takes inspiration from (Luo *et al.* (2016)), which also uses a dot product layer to efficiently take care of all the negatives. However, their approach is formulated as a patch-image matching task, while our approach implements an image-to-image semantic keypoints matching strategy.

7.4 Correspondence Search

In semantic correspondence search between a source image I_1 with query points p_n and a target image I_2, the goal is to precisely localize the matching point q_n for every query point in the target image I_2. Our framework for semantic correspondence search is shown in Figure 7.1. It consists of a feature generator and a matching network.

The feature generator takes the source image I_1 and the target image I_2 as inputs and generates features F_1 and F_2. The matching network takes the features F_1 and F_2 as inputs and generates N similarity heatmaps for N query points.

Variant	Layers	Pooling	Multi-resolution features
GoogleNet 3c	8	8	no
GoogleNet 4a	10	16	no
GoogleNet 4c	12	16	no
GoogleNet 4e	14	16	no
GoogleNet 5b	17	32	no
GoogleNet Skip1	17	16	yes
GoogleNet Skip2	17	8	yes

Table 7.1: Configuration of GoogleNet Szegedy *et al.* (2015) variants explored.

This section is organised as follows: In Section 7.4.1 we describe the feature generator and various explored variants of the feature generator. Section 7.4.2 explains the matching network. Training and Inference are described in Sections 7.4.3 and 7.4.4, respectively.

7.4.1 Feature Generator

The feature generator is shown in Figure 7.1. It consists of two copies of the batch-normalized Google-Net Szegedy *et al.* (2015) up to layer 17 with shared parameters. Channel wise l2-normalization is applied to the output of each network. The feature generator takes source and target image, I_1 and I_2, as input, passes them through a series of convolution, max-pooling and inception layers, and generates features F_1, $F_2 \in \mathbb{R}^{C \times W/P \times H/P}$, where W and H are the widths and heights of the input images and C is the dimensionality of the features, respectively and P is the overall pooling.

In this work, we explore different variants of the Google-Net to study the effect of pooling and semantics at various layers of the feature generator. The variants used are shown in Figure 7.3. We term the first five layers of the Google-Net with an over all stride of 8 as the Base Network. Each explored variant is then build on the base network and differs from the others in the number of layers and pooling used as described in Table 7.1. The number of layers do not include deconv layers.

We start with the variant that adds inception layers 3a, 3b, 3c from the Google-Net architecture on the top of base network and has same pooling/stride as the base network. We refer to it as GoogleNet 3c. We keep on adding next two layers i.e pooling/inception layers from the Google Net and generate new variants as shown in Figure 7.3 and Table 7.1. We also explore variants with multi-resolution features which we refer to as GoogleNet skip1 and GoogleNet skip2 with 1 and 2 skip connections respectively.

7.4.2 Matching Network

To analyse the performance of various variants, we propose a novel matching network that efficiently generates a peaked similarity heat map for every query point over the target image. This is in contrast to standard practice, where either a matching framework

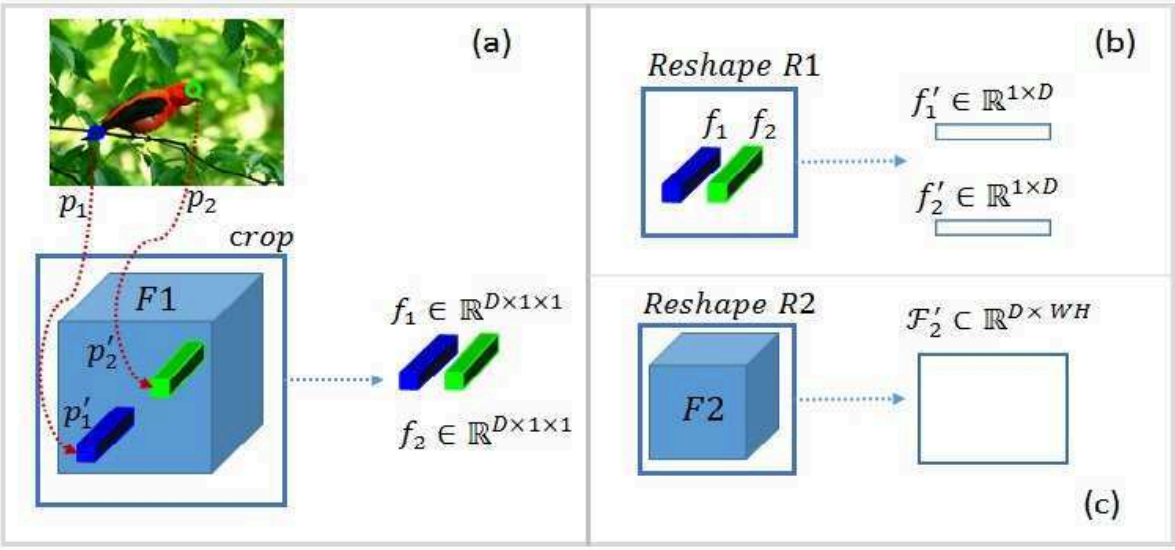


Figure 7.2: Illustration of the Crop and Reshape layers. (a) The Crop layer cropping F_1 around p_1', p_2' and generating cropped features f_1, f_2 for query-points p_1 and p_2. (b) Reshape R1 layer reshaping features f_1 for query point p_1 into a row vector f_1'. (c) Reshape R2 layer reshaping the $3D$ tensor F_2 into a $2D$ matrix F_2'.

(Ham *et al.* (2016), Liu *et al.* (2011), Kim *et al.* (2013)) is applied over the descriptors obtained from a CNN (Kim *et al.* (2017)) or (Choy *et al.* (2016)) where metric learning is applied on top of features learned by the deep network. This requires an extra hard-negative mining step during training and an extra nearest neighbor search during inference. In addition to that, the network is not trained for the end goal of precise localization of the correct correspondence as during training pixels within a radius r of the ground-truth correspondence are taken as positives. Although this provides more positive pairs, it may negatively effect the precise localization for small semantic objects like hands of a person or beak of a bird.

During training, the network is trained to minimize the difference between the ground truth and the predicted similarity map. This eliminates the need for hard negative mining, as the network learns as part of its training procedure to generate dissimilar features for *all* the negatives of any query point.

The matching network is shown in Figure 7.4 and consists of a series of simple layers with no trainable parameters. It takes features F_1, F_2 and N query points as input, passes them through a series of simple layers, and generates N peaked similarity heatmaps for each query-point p_n in image I_1, as shown in Figure **??**. In the following, we explain the operations performed by each layer in detail.

Crop layer. The crop layer takes features F_1, query points p_n and their lower resolution scalings $p_n' = \frac{p_n}{P}$ as inputs and generates a set of cropped features, $\{f_n\} \in \mathbb{R}^{D\times1\times1}$, by

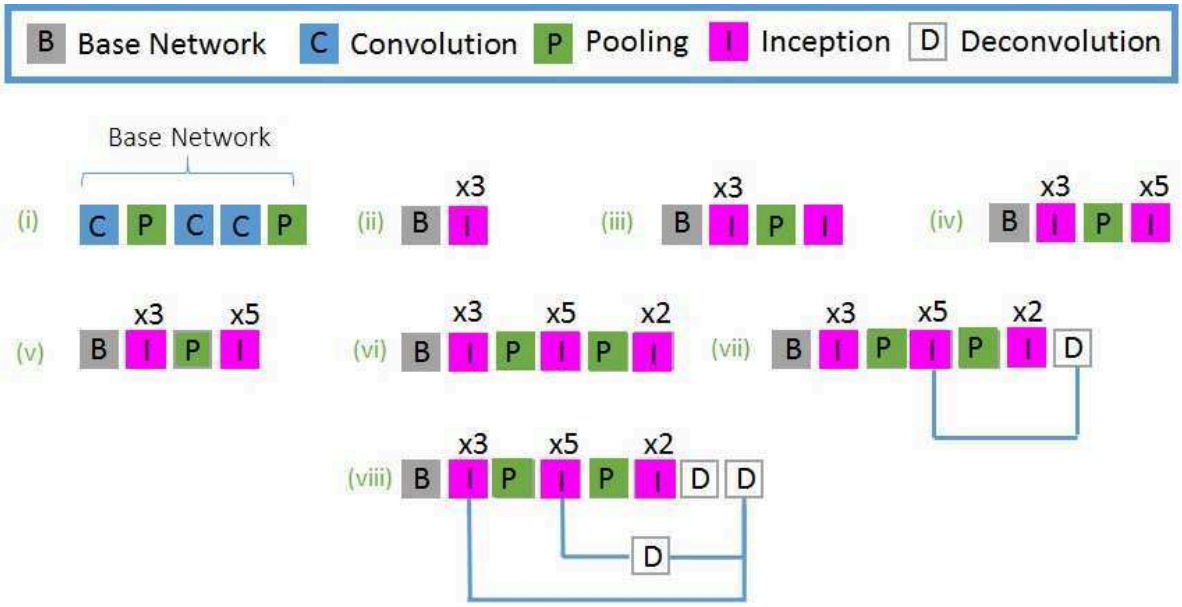


Figure 7.3: GoogleNet variants. (i) Base Network. (ii) GoogleNet 3c. (iii) GoogleNet 4a. (iv) Google Net 4c (v) GoogleNet 4e. (vi) GoogleNet 5b. (vii) GoogleNet Skip1. (viii) GoogleNet Skip2.

cropping F_1 around each p'_n. This is illustrated in Figure 7.2 (a) for two query points p_1 and p_2.

Reshape R1. The Reshape R1 layer receives the set of cropped features $\{f_n\}$. It reshapes each $f_n \in \mathbb{R}^{\mathcal{D}\times 1\times 1}$ into a row vector $f'_n \in \mathbb{R}^{1\times\mathcal{D}}$ and generates a set of reshaped cropped features $\{f'_n\}$. An example is shown in Figure 7.2 (b) for cropped features f_1 and f_2.

Copy. The copy layer generates the set $\{F_2^n\}$ consisting of N shallow copies of the $3D$ tensor F_2.

Reshape R2. The Reshape R2 layer receives the set of $3D$ tensors $\{F_2^n\}$ as inputs and reshapes each F_2 into a $2D$ matrix $\hat{F}_2 \in \mathbb{R}^{\mathcal{D}\times WH}$. This is illustrated in Figure 7.2 (c).

Dot Product. The Dot product layer receives the set of reshaped cropped features $\{f'_n\}$ and N reshaped $2D$ matrices $\{\hat{F}_2^n\}$ as inputs and generates a set of $1D$ similarity vectors $\{\hat{S}^n \in \mathbb{R}^{1\times WH}\}$. Each $\hat{S}^n$ is computed as

$$\hat{S}^n = f'_n \odot \hat{F}_2 \ , \tag{7.1}$$

where $\odot$ is the dot product operation. The dot product layer efficiently computes the similarity of every cropped feature f_n with every feature in F_2.

Soft-Max. The soft-max layer normalizes the similarity score for every element $s_k \in \hat{S}^n$, where $k = \{1, ..., WH\}$, as follows:

$$\hat{s_k} = \frac{\exp(s_k)}{\sum_{i=k}\exp(s_i)} \ . \tag{7.2}$$

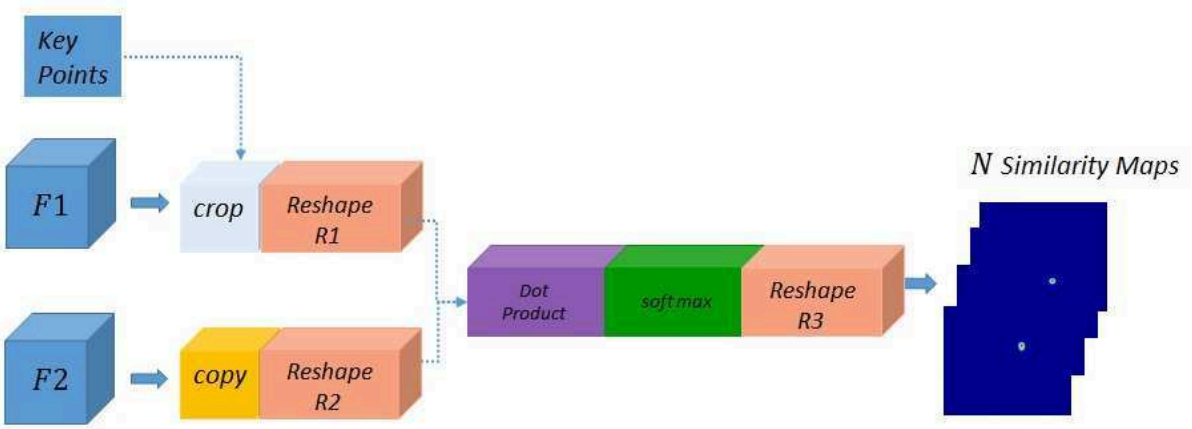


Figure 7.4: Matching Network. It takes F_1, F_2 and N query points as inputs and outputs N similarity maps for N query-points in image $\mathcal{I}_1$.

Reshape R3. The Reshape R3 layer reshapes each $1D$ similarity vector in $\{\hat{S}^n\}$ into a $2D$ similarity map $\hat{S}^n_{x,y} \in \mathbb{R}^{W \times H}$ and generates a set of $2D$ similarity maps $\{\hat{S}^n_{x,y}\}$.

All the above operations in combination with the feature generation are carried out in a single forward pass during training and inference.

7.4.3 Training

We denote training examples as $(I_1, I_2, \{p_n\}, \{q_n\})$. We generate a ground truth similarity map $S^n_{x,y} \in \mathbb{R}^{W/P \times H/P}$, with $x = \{1, ..., W/P\}$ and $y = \{1, ..., H/P\}$, for every query point $p_n = (x_n, y_n)$ with ground truth correspondence $q_n = (x_q, y_q)$ as

$$\mathcal{S}^n_{x,y} = \begin{cases} 1, & \text{if } x = x_{q_n}, y = y_{q_n} \\ 0, & \text{otherwise} \end{cases} . \tag{7.3}$$

During training, given training examples $\{(I_1, I_2, \{p_n\}, \{q_n\}, \{S^n_{x,y}\})\}$, we minimize the binary cross-entropy error between every predicted $\hat{S}^n_{x,y}$ and ground-truth similarity map $S^n_{x,y}$ in an image pair as:

$$\min_w - \sum_{x,y} S^n_{x,y} \log(\hat{S}^n_{x,y}) + (1 - S^n_{x,y})(1 - \log(\hat{S}^n_{x,y})) \ , \tag{7.4}$$

where w are weights of the network. The weights are learned using back-propagation with the Adam optimizer.

7.4.4 Inference

During inference we our method predicts a matching correspondence q_n for every query point p_n as follows:

$$q_n = \arg\max_{x,y} \hat{S}^n_{x,y} \tag{7.5}$$

where q_n is the position of the peak in the predicted similarity map. We up-sample the generated similarity-map $\hat{S}^n_{x,y}$ using bilinear-sampling before localizing the peak.

7.5 Experiments

We evaluate the proposed method on the tasks of semantic keypoints matching, dense matching and instance keypoints mathcing.

Evaluation Metric. Following recent work on semantic correspondence mathcing (Choy *et al.* (2016); Han *et al.* (2017)), we use PCK as the evaluation metric. According to the metric, a predicted correspondence is correct if its euclidean distance in pixels to the ground-truth lies with in a threshold T. Different image resolutions are taken into account by normalizing the euclidean distance with the diagonal of the target image (Choy *et al.* (2016); Han *et al.* (2017)). We use PCK @ T for all experiments unless stated otherwise.

Experiment Protocol. We use the Torch7 (Collobert *et al.* (2011)) framework for training and inference. For experiments on PF-PASCAL (Ham *et al.* (2016)), PF-Willow (Ham *et al.* (2016)) and Pascal-Parts (Zhou *et al.* (2015)) datasets we follow recent work (Han *et al.* (2017)) and train our method on PF-PASCAL dataset only.

Our experimental settings are as follows: We train the network from scratch without any pre-training with a learning rate of .00092 using a stair case decay of .95 applied every 73 epochs. We use the Adam optimizer (Kingma and Ba (2014)) with $\beta_1 = 0.9$ and $\epsilon = 0.1$. We train the network for 160 epochs for each dataset. During each training epoch we augment each pair with random scaling $s \in [0.5, 1.5]$ and rotation $r \in [-20, 20]$ to handle scale and rotation variations across pairs. Horizontal flipping is applied with probability .5. We use image crops of 256×256. During training, we use crops around image centres. During inference we also use crops around bounding box centres, if available. Unless otherwise stated we report performance for both crops.

7.5.1 Semantic Keypoints Matching.

For semantic keypoints matching we evaluate on the CUB-200 (Wah *et al.* (2011)), PF-PASCAL (Ham *et al.* (2016)), PF-Willow (Ham *et al.* (2016)) and Pascal-Parts (Zhou *et al.* (2015)) datasets, respectively.

PF-Pascal dataset. The PF-Pascal dataset (Ham *et al.* (2016)) consists of 1300 image pairs. The annotations are used from the PASCAL-Berkeley data-set (Bourdev and Malik (2009); Everingham *et al.* (2011)). For fair comparison with recent work, we use 700 training pairs, 300 validation pairs and 300 test pairs from (Han *et al.* (2017)) respectively.

GoogleNet variants analysis. We evaluate different variants of GoogleNet described in 7.4.1 on the 300 PF-Pascal test pairs. The PCK achieved by variants at thresholds $T \in [0.01, 0.1]$ is shown in Figure 7.5. The PCK is reported for crops around the bounding boxes.

We first analyse the performance variants 3c and 5b. GoogleNet 3c has good resolution with minimal pooling. GoogleNet 5b has strong semantics and large pooling. Although GoogleNet 3c performs better than GoogleNet 5b at small thresholds the performance

Method	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv	mean
NAMHOG (Ham *et al.* (2016))	72.9	73.6	31.5	52.2	37.9	71.7	71.6	34.7	26.7	48.7	28.3	34.0	50.5	61.9	26.7	51.7	66.9	48.2	47.8	59.0	52.5
PHMHOG (Ham *et al.* (2016))	78.3	76.8	48.5	46.7	45.9	72.5	72.1	47.9	49.0	84.0	37.2	46.5	51.3	72.7	38.4	53.6	67.2	50.9	60.0	63.4	60.3
LOMHOG (Ham *et al.* (2016))	73.3	74.4	54.4	50.9	49.6	73.8	72.9	63.6	46.1	79.8	42.5	48.0	68.3	66.3	42.1	62.1	65.2	57.1	64.4	58.0	62.5
UCN (Choy *et al.* (2016))	64.8	58.7	42.8	59.6	47.0	42.2	61.0	45.6	49.9	52.0	48.5	49.5	53.2	72.7	53.0	41.4	83.3	49.0	73.0	66.0	55.6
SCNet-A (Han *et al.* (2017))	67.6	72.9	69.3	59.7	**74.5**	72.7	73.2	59.5	51.4	78.2	39.4	50.1	67.0	62.1	69.3	**68.5**	78.2	63.3	57.7	59.8	66.3
SCNet-AG (Han *et al.* (2017))	83.9	81.4	70.6	62.5	60.6	81.3	81.2	59.5	53.1	81.2	**62.0**	58.7	65.5	73.3	51.2	58.3	60.0	69.3	61.5	**80.0**	69.7
SCNet-AG+ (Han *et al.* (2017))	85.5	84.4	66.3	70.8	57.4	82.7	82.3	71.6	54.3	95.8	55.2	59.5	68.6	75.0	56.3	60.4	60.0	**73.7**	66.5	76.7	72.2
GoogleNet Skip1 (bbxcrop)	**88.9**	**94.5**	81.2	56.2	65.7	**91.0**	**83.2**	**76.1**	**68.6**	**97.9**	56.8	**75.5**	91.1	88.7	72.4	68.1	85.7	55.7	81.0	62.3	**81**
GoogleNet Skip2 (bbxcrop)	86.3	93.0	**82.4**	**59.4**	60.0	90.4	89.2	78.3	62.9	73.0	36.4	73.6	**92.2**	**92.6**	**73.0**	66.0	**90.0**	55.6	**83.0**	44.3	80
GoogleNet Skip1 (imcrop)																					69.4
GoogleNet Skip2 (imcrop)																					69

Table 7.2: PCK comparison with state-of-the-art on the 300 test pairs of the PF-Pascal dataset (Ham *et al.* (2016)) across 20 object categories at fixed threshold $T = 0.1$. GoogleNet Skip1 and GoogleNet Skip2 outperform state-of-the-art significantly for bbox crop setting.

could be significantly improved by combining features from higher layers. Same is true for GoogleNet 5b at large thresholds.

Google Net 4a, 4c and 4e slightly differ in the number of layers and have same pooling. However, this slight change in the architecture improves the performance significantly at larger thresholds especially from variant 4a to 4c.

GoogleNet skip1 has multi-resolution features from variants 4e and 5b respectively with an overall pooling of 16. Although features from 5b alone performs sub-optimally but when combined with features from 4e, improve the performance at larger thresholds. At lower thresholds it still performs sub-optimally mainly due to more pooling.

GoogleNet skip2 has multi-resolution features from variants 3c, 4e and 5b respectively and achieves the best performance across all thresholds except at $T = 0.1$ where GoogleNet Skip1 outperforms. This suggests that well-explored combination of multi-resolution features is indeed crucial for achieving best performance in the context for image-image semantic correspondence matching at various thresholds.

Comparison with State-of-the-art. We compare the PCK achieved by GoogleNet skip1 and GoogleNet skip2 at fixed threshold $T = 0.1$ to recent work in Table 7.2 on the 300 PF-Pascal (Ham *et al.* (2016)) test pairs. The comparison is done across 20 object categories. GoogleNet Skip1 (bbxcrop) and GoogleNet Skip2 (bbxcrop) outperform state-of-the-art significantly with a mean PCK of 81 and 80 respectively. GoogleNet Skip1 (imcrop) and GoogleNet Skip3 (imcrop) achieve competitive performance of 69.4 and 69 respectively.

The results in Table 7.2 suggest that by paying special attention to feature generator design alone in the form of combining features at multiple resolutions is beneficial for the the overall performance and makes usage of feature transformer layer (Choy *et al.* (2016)) or explicit geometry information (Han *et al.* (2017)) unnecessary.

PF-Willow dataset. We evaluate the transferability of the features learned by GoogleNet skip1 and GoogleNet skip2 on the PF-Willow dataset (Ham *et al.* (2016)).

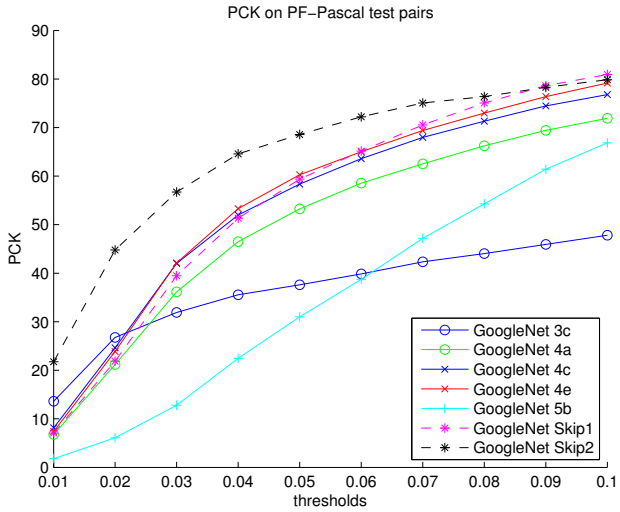


Figure 7.5: The PCK achieved by different variants of GoogleNet on the 300 PF-Pascal dataset Ham *et al.* (2016) test-pairs at thresholds $T \in [0.01, 0.1]$. PCK for variants with multi-resolution features is shown in dashed lines. GoogleNet Skip2 achieves the best PCK at various thresholds along with GoogleNet Skip1 outperforming at the threshold of 0.1.

The PF-Willow dataset consists of 10 objects sub-classes with 10 keypoint annotations for each image. We evaluate the performance using the PCK at thresholds of 0.05, 0.1 and 0.15 respectively and comapare it with recent state-of-the-art. The comparison is shown in Table 7.3 . The PCK is averaged over the 10 object sub-classes. GoogleNet Skip1 with bounding box crops outperforms SCNet for PCK@.05 and PCK@0.1 and achieves competitive performance for PCK@.15 based on raw point correspondences computed using appearance only.

GoogleNet skip1 (bbx crop) achieves state-of-the performance for thresholds of 0.05 and 0.1 respectively and achieves a competitive performance of .8333 at threshold 0.15.

Pascal Parts Dataset. We also evaluate transferability on the sampled Pascal-Parts dataset (Zhou *et al.* (2015)) using PCK at a fixed threshold of 0.05 and compare against recent state-of-the-art. The PCK is averaged over all the classes. The results are shown in Table 7.4. Both the variants outperform all recent state-of-the-art methods in both bbx-crop and image crop settings.

Qualitative Results PF-Pascal. We show qualitative results for keypoints transfer on the PF-Pascal dataset in Figure 7.6 (a). Left image is the source image with query points. Middle image is the target image with ground-truth correspondences. Right image contains the keypoint transfers from the query-image to the target image through correspondence matching.

Method	PCK @ 0.05	PCK @ 0.1	PCK @ 0.15
SIFTFlow (Liu *et al.* (2011))	0.247	0.380	0.504
DAISY w/SF (Yang *et al.* (2014))	0.324	0.456	0.555
FCSS w/SF (Kim *et al.* (2017))	0.354	0.532	0.681
FCSS w/PF (Kim *et al.* (2017))	0.295	0.584	0.715
LOMHOG (Ham *et al.* (2016))	0.284	0.568	0.682
UCN (Choy *et al.* (2016))	0.291	0.417	0.513
SCNet-A (Han *et al.* (2017))	0.390	0.725	0.873
SCNet-AG (Han *et al.* (2017))	0.394	0.721	**0.871**
SCNet-AG+ (Han *et al.* (2017))	0.386	0.704	0.853
GoogleNet Skip1 (bbxcrop)	**0.520**	**0.780**	0.833
GoogleNet Skip2 (bbxcrop)	0.50	0.653	0.77
GoogleNet Skip1 (imcrop)	0.411	0.690	0.780
GoogleNet Skip2 (imcrop)	0.392	0.540	0.680

Table 7.3: PCK comparison with state-of-the art on the PF-Willow (Ham *et al.* (2016)) dataset @ thresholds of 0.05, 0.1 and 0.15 respectively. The PCK is averaged over the 10 sub-classes from the dataset.

Method	PCK @ 0.05
NAMHOG (Ham *et al.* (2016))	0.13
PHMHOG (Ham *et al.* (2016))	0.17
LOMHOG (Ham *et al.* (2016))	0.17
FCSS w/SF (Kim *et al.* (2017))	0.28
FCSS w/PF (Kim *et al.* (2017))	0.29
SCNet-A (Han *et al.* (2017))	0.17
SCNet-AG (Han *et al.* (2017))	0.17
SCNet-AG+ (Han *et al.* (2017))	0.18
GoogleNet Skip1 (bbxcrop)	0.33
GoogleNet Skip2 (bbxcrop)	**0.371**
GoogleNet Skip1 (imcrop)	0.32
GoogleNet Skip2 (imcrop)	0.37

Table 7.4: PCK comparison with state-of-the-art on the sampled Pascal-Parts (Zhou *et al.* (2015)) dataset at a fixed threshold of 0.05.

We show qualitative results with similarity heatmaps generated by the matching network on the PF-Willow (Ham *et al.* (2016)) dataset in Figure 7.7. Results show that the matching network generates peaky similarity heatmaps that are used to precisely localize the correspondence. Results from rows 1 and 2 our method handles rotation, scale variations and parts symmetry like wheels of the car and bikes very well without using any spatial transformer layers or explicit geometry.

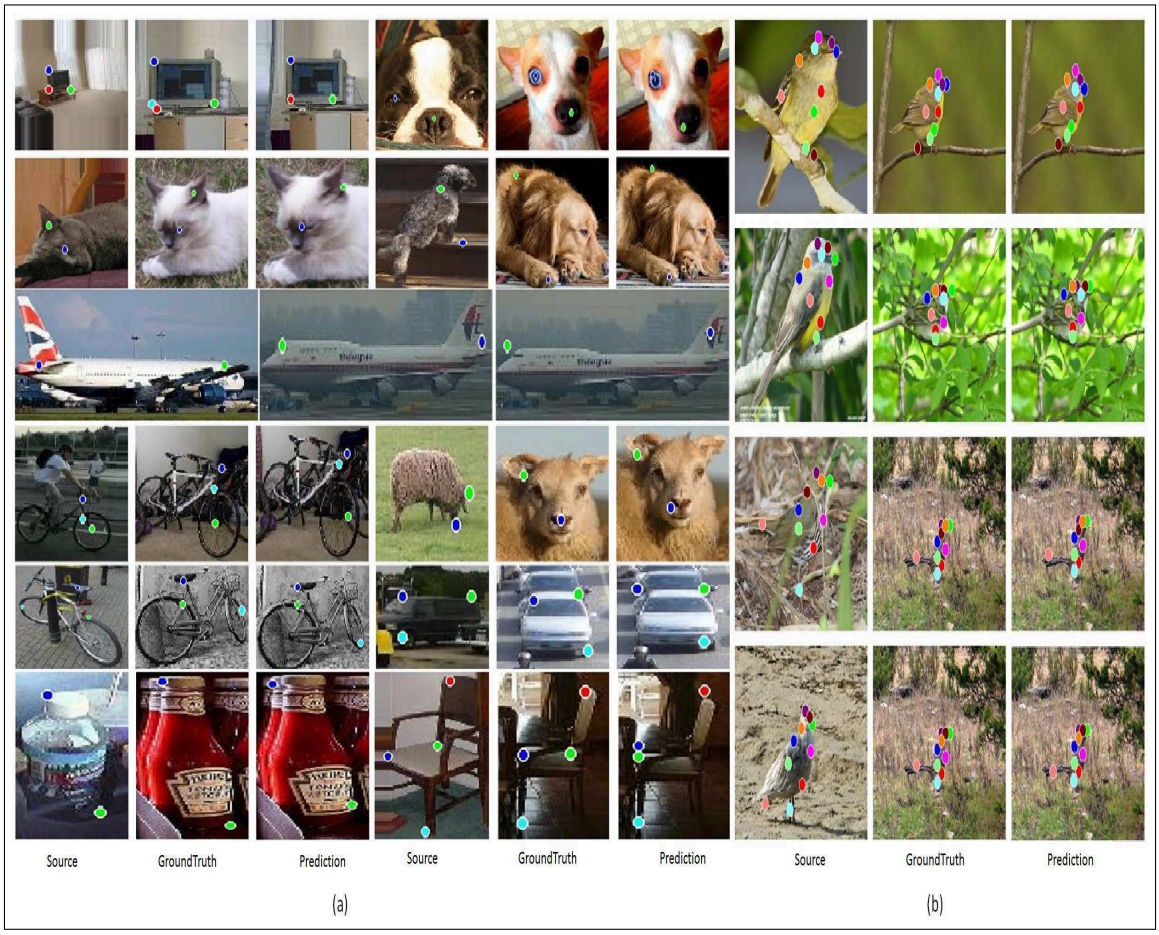


Figure 7.6: Qualitative results for keypoints transfer on the PF-Pascal (Ham *et al.* (2016)) and CUB-200 (Wah *et al.* (2011)) datasets respectively. (a) PF-Pascal (Ham *et al.* (2016)) dataset. (Left) Source image with query points. (Middle) Target image with ground truth correspondences. (Right) Target image with keypoint transfers through correspondence matching. (b) CUB-200 (Wah *et al.* (2011)) dataset. (Left) Source image with query points. (Middle) Target image with ground truth correspondences. (Right) Target image with keypoint transfers through correspondence matching.

CUB 200 dataset. We train and evaluate our method on the CUB 200 (Wah *et al.* (2011)) dataset. During training we use the 1676 pairs from (Choy *et al.* (2016)). During evaluation we follow (Choy *et al.* (2016)) and evaluate on the clean 5000 test pairs from Warpnet (Kanazawa *et al.* (2016)) using the PCK evaluation measure. We report the performance for GoogleNet skip1 image crop settings.

In Figure 7.8 we plot the PCK at thresholds $T \in [0.01, 0.1]$ on the clean 5000 pair of the CUB 200 dataset and compare it against state-of-the-art universal correspondence network (Choy *et al.* (2016)). GoogleNet skip2 outperforms UCN at all thresholds

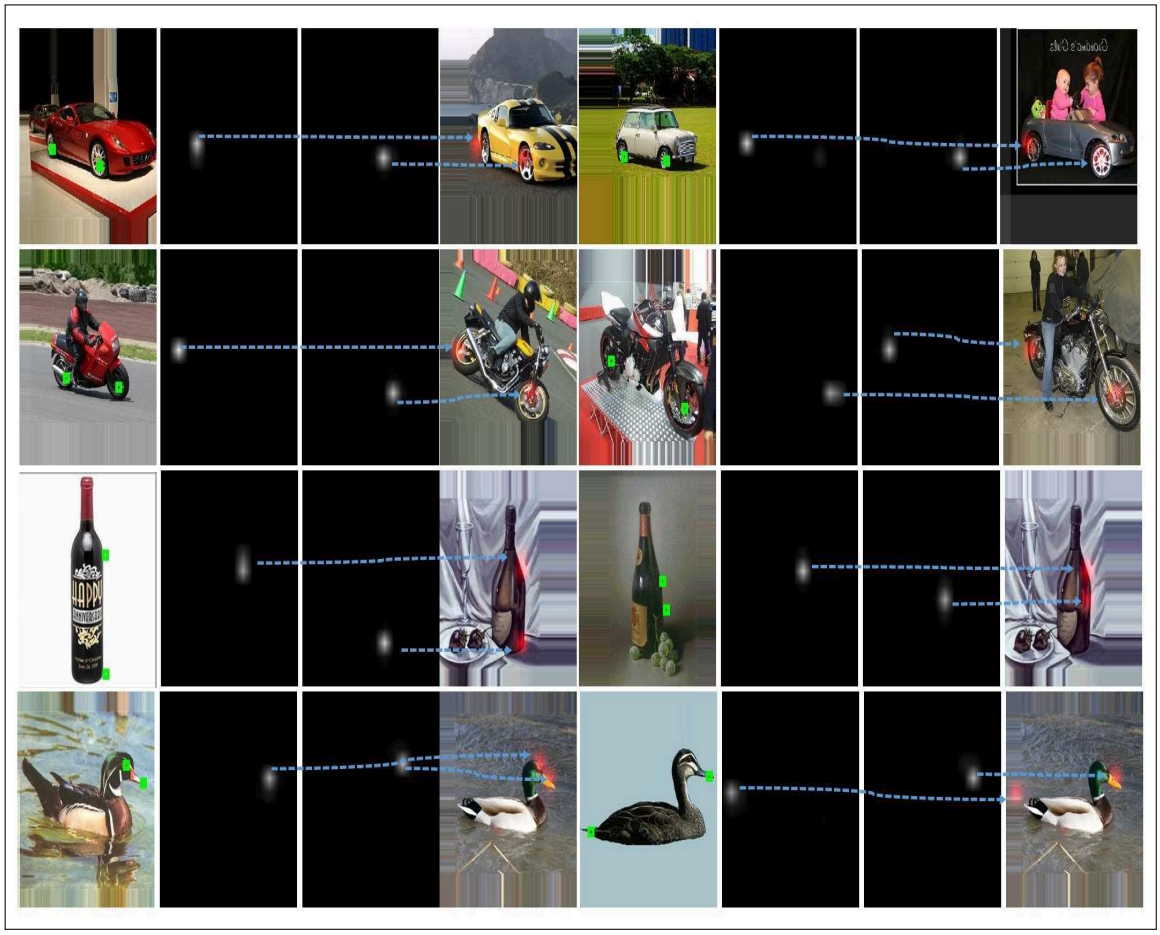


Figure 7.7: Qualitative results on the PF-Willow (Ham *et al.* (2016)) dataset with similarity heatmaps. Each row shows two examples. Left image is the source image with query points shown by green rectangles. Predicted similarity heatmaps are shown in the middle images. On the right target image is shown with overlaid heatmaps.

without using any spatial transformer layers. Here again results support our claim that well-studied combination of features from a cnn architecture is important and sufficient for good performance.

Qualitative Results. In Figure 7.6 we show qualitative results for keypoint transfer on the CUB-200 (Wah *et al.* (2011)) dataset. The keypoints are transfered from the query image to target image through correspondence matching.

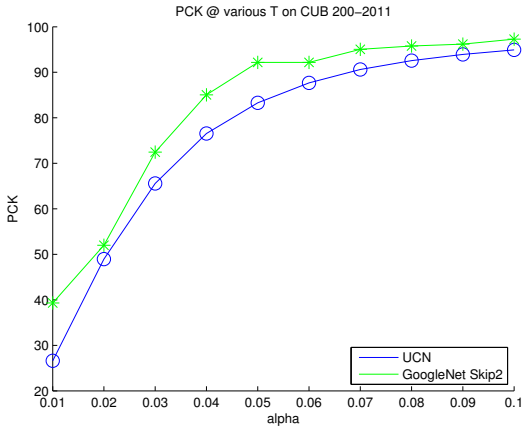


Figure 7.8: PCK comparison with state-of-the-art universal correspondence network at thresholds Choy *et al.* (2016) $T \in [0.01, 0.1]$ on the clean 5000 pairs from (Kanazawa *et al.* (2016)) of the CUB-200 dataset (Wah *et al.* (2011)) dataset.

7.5.2 Dense Matching

For dense correspondences, we evaluate on the KITTI-Flow 2015 (Menze and Geiger (2015)) and MPI Sintel (Butler *et al.* (2012)) benchmarks respectively. We use the train/test split of the training set employed in (Choy *et al.* (2016)) for both KITTI-Flow and MPI-Sintel. We follow the performance measure of PCK @ $10px$ from deep matching that is used in (Choy *et al.* (2016)) for evaluation. According to the evaluation measure, a predicted correspondence is considered as correctly matched if it lies within 10 pixels of the ground-truth.

We use image resolutions of 1242×375 and 1024×436 for KITTI-Flow and MPI-Sintel during training and inference, respectively. We randomly sample 1000 query-match points for each image pair during each training epoch.

The results for dense correspondences are shown in Table 7.5. Our approach significantly outperforms all previous state-of-the-art methods using just raw correspondences from our framework without any post-processing steps. In particular, our approach also achieves better performance than DAISY (Yang *et al.* (2014)), DSP (Kim *et al.* (2013)), and DM (Revaud *et al.* (2016)), which apply global optimization as a post-processing step to achieve more precise correspondences. We do not use spatial transformer layers (Jaderberg *et al.* (2015)) and therefore do not experience overfitting with less training examples for the KITTI-Flow (Menze and Geiger (2015)) dataset, as reported in (Choy *et al.* (2016)).

Qualitative Results. Our qualitative results for dense correspondences are shown in Figure 7.9.

Figure 7.9: Qualitative results for dense correspondences on the MPI-Sintel 9Butler *et al.* (2012)) and KITTI-Flow (Menze and Geiger (2015)) datasets respectively. (Left) Source image with dense query points. (Right) Target image with predicted points.

Method	Kitti-Flow	MPI-Sintel
SIFT-NN (Lowe (2004))	48.9	68.4
HOG-NN (Dalal and Triggs (2005))	53.7	71.2
SIFT-flow (Lowe (2004))	67.3	89.0
DaisyFF (Yang *et al.* (2014))	79.6	87.3
DSP (Kim *et al.* (2013))	58.0	85.3
DM (Revaud *et al.* (2016))	85.6	89.2
UCN-HN (Choy *et al.* (2016))	86.5	91.5
UCN-HN-S (Choy *et al.* (2016))	83.4	90.7
Ours	**91.8**	**92.0**

Table 7.5: Dense Correspondence Matching Performance for PCK@10 on the (Menze and Geiger (2015)) and MPI Sintel (Butler *et al.* (2012)) test sets. Our raw-correspondences out-perform all previous state-of-the-art.

7.5.3 Instance Keypoints Matching

We evaluate on the Posetrack dataset (Iqbal and Gall (2017)) for the task of instance keypoints matching. Posetrack is a new and challenging dataset for the task of multi-person pose tracking. The dataset consists of 30 sequences for training and 30 sequences for testing. The number of frames per sequence vary between 41 and 151. Each sequence consists of 2 to 16 persons. The dataset has a total of 16000 annotated poses.

The task of instance keypoints matching is shown in Figure 7.11. Ideally the matching method should locate the right correspondence on the correct instance. Instance keypoints matching has applications in multi-person pose tracking from videos. Despite recent success in single frame multi-person pose estimation (He *et al.* (2017), Newell *et al.* (2017)), multi-person pose tracking still remains a challenge. The task also includes people association over time. The task is difficult since the pose and appearance of people changes over time. People may overlap each other or may be truncated.

For evaluation we use PCKh. According to the measure a match is correct if the distance between the match and the ground-truth is less than a threshold T. To take different sizes of people into account the distance is normalized by the head size of a person.

We evaluate the proposed approach in two different settings :

- Prediction Mode : The system has to search the entire target image for locating the right correspondence and instance.
- Association Mode : Semantic correspondences are available and system has to find the right instance only.

The two settings are shown in Figure 7.12. Quantitative results for both settings are given in Table 7.6. The results shown are for neighbouring frames in the video

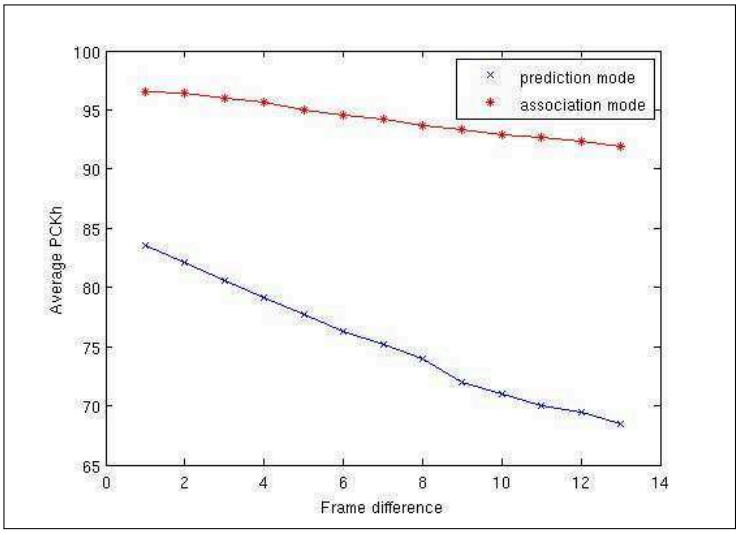


Figure 7.10: Average PCKh evaluated at various frame lengths. The performance drops significantly from 83.5 to 68.5 in prediction mode. The performance changes slightly in association mode.

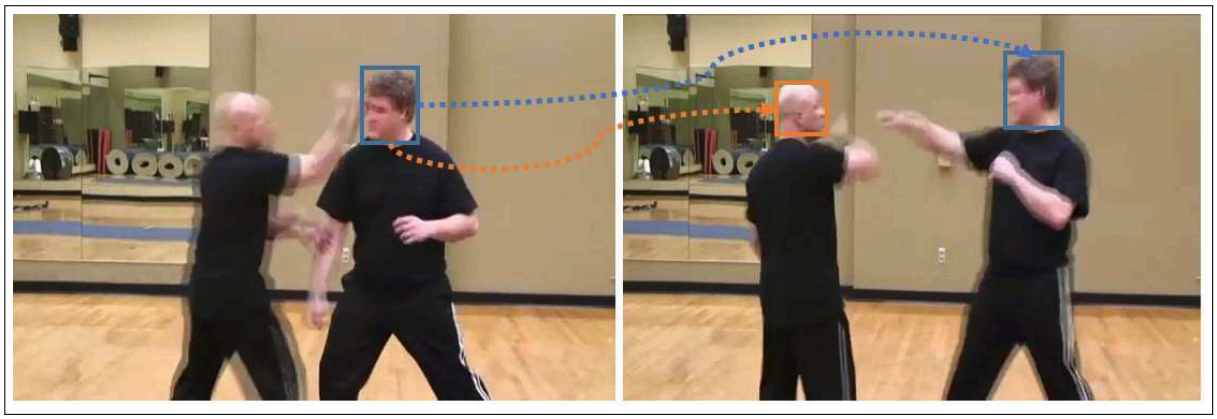


Figure 7.11: Instance keypoints matching task. (Left) Source image with head (shown in blue rectangle as the query point). The matching method should locate head(orange and blue rectangle) in the right image on the correct instance(blue rectangle).

sequence. The model performs better in association mode than in prediction mode since the correspondences are already available in association mode.

We evaluate the performance of our approach for various frame lengths in both settings. Results are shown in Figure 7.10. As expected, the performance drops as frame difference increases for the case of prediction mode. For association mode the performance drops slightly. Qualitative results are shown in Figure 7.13. The results show that our method successfully locates the right correspondence on the right instance,

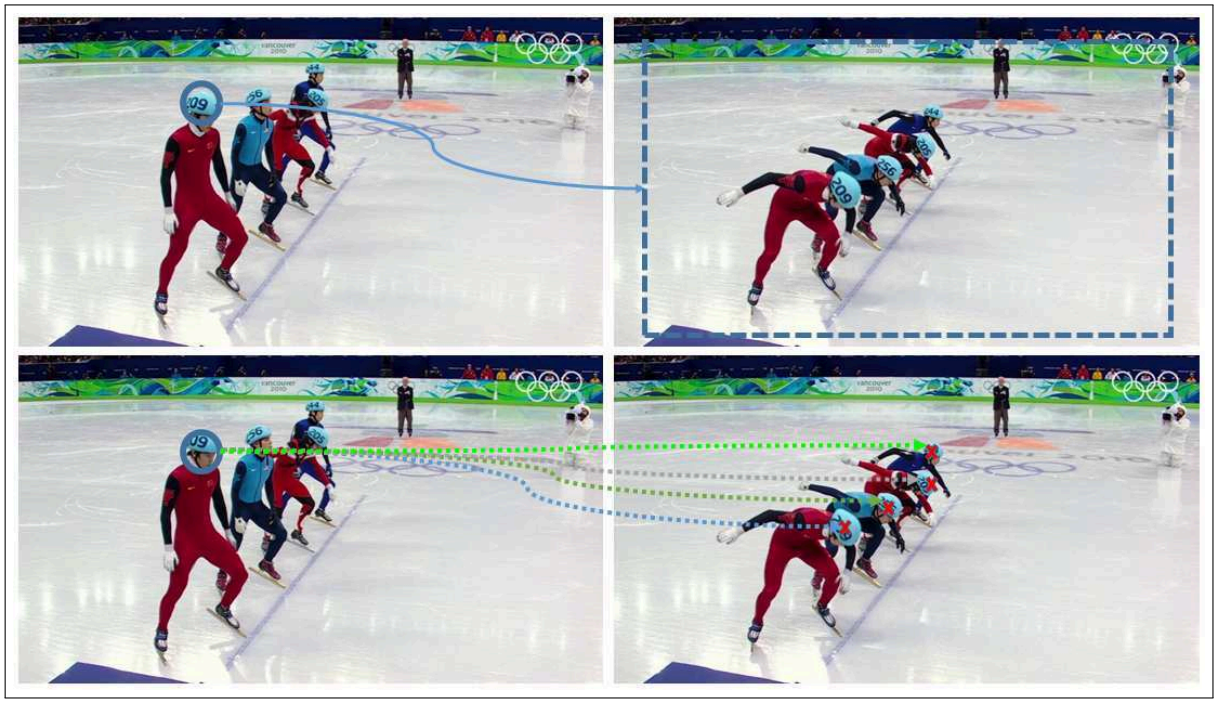


Figure 7.12: Instance keypoints matching settings. (Top) Prediction Mode: System has to search the entire target image to locate right correspondence and instance. (Bottom) Assoication Mode: System has to locate the right instance only. Correspondences are available.

Method	Head	Shou	Elbow	Wri	Hip	Knee	Ankle	Total
Prediction Mode	78	83.6	86.2	86.4	84.4	83.2	82.8	83.4
Association Mode								96.7

Table 7.6: Instance keypoints matching results on the PoseTrack (Iqbal and Gall (2017)) dataset. The results are evaluated using PCKh @ 50. The results are computed for neighbouring frames.

7.6 Conclusion

In this chapter, we have proposed a fully supervised method for the task of correspondence matching. We have evaluated the proposed method on three variants of correspondence matching, namely, semantic matching, dense matching and instance keypoints matching. The proposed methods achieves competitive results on the poupular benchmarks for the three variants. Our proposed system has a novel matching network that efficiently generates peaked similarity heatmaps for every semantic query point over the target image. The matching network can be appended to any base network e,g VGG or GoogleNet to make it end-to-end trainable for precise location of semantic keypoints

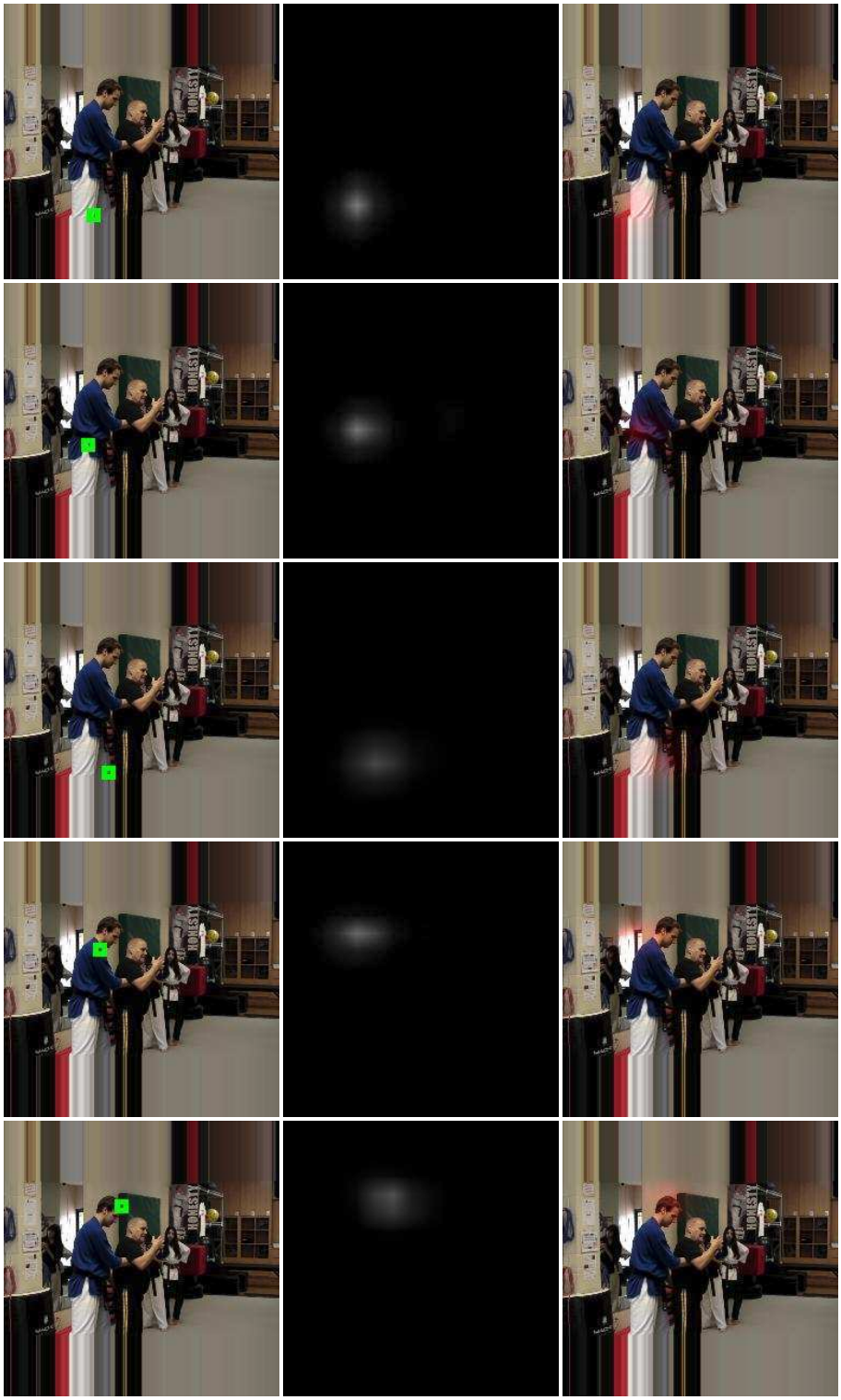


Figure 7.13: Qualitative results for Instance keypoints matching on the Posetrack dataset (Iqbal and Gall (2017)). (left) Frame t with a query point. (Middle) Similarity map generated by our method. (right) Similarity map overlaid on the frame $t+1$.

and thus makes the extra hard-negative mining step obsolete. With the matching network we have studied the effect of pooling and semantics at various layers on the overall performance for task of image-image semantic keypoint matching. We have shown with experiments on PF-Willow dataset (Ham *et al.* (2016)) that a well-explore combination of multi-resolution features is indeed important and sufficient to achieve state-of-the performance.

The proposed method assumes that keypoints are available beforehand. However, in many computer vision systems it is desirable to jointly detect keypoints and perform matching. Exploring a method that jointly detects keypoints and performs matching is an interesting direction for future work.

8 Conclusion

In this thesis, we have approached the problem of human pose estimation. In the start of the thesis, we have proposed a semantic occlusion model for 3D human pose estimation under partial occlusion from depth data. Instead of discarding the occluder information, our method exploits the rich context of the occluding objects to infer the positions of occluded joints. The approach outperforms the existing Kinect 2 SDK for occluded joints. We have also introduced the Pose Occlusion test set for evaluation of pose occlusion approaches.

We then proposed an efficient convolutional neural network (CNN) architecture for 2D single person pose estimation from RGB data. The proposed CNN runs efficiently on a mid-range GPU and is on par with recent state-of-the-art methods for 2D single person pose estimation. Then we approached the more challenging task of multi-person pose estimation in uncontrolled settings. We have proposed a bottom up approach for multi-person pose estimation with a novel pair-wise associative voting. The relative offsets are used to associate different body joints in a greedy manner. Finally we have proposed a framework for correspondence matching with an efficient matching layer. The proposed framework is applied to the task of semantic key points matching, dense matching and instance key points matching. The proposed framework achieves competitive performance.

8.1 Summary and Contributions

We now summarize in detail the contributions made in this thesis.

In chapter 4, we have proposed a semantic occlusion model for 3D human pose estimation under partial occlusion from objects. The model is incorporated in to an existing regression forest framework for 3D human pose estimation. The model exploits the rich context of commonly occurring occluding objects like table and chair to infer the 3D positions of occluding joints. The model is tested on partially occluded sitting and standing people. The model outperforms the commercially available Kinect 2 SDK for occluded joints. We have also proposed a Pose Occlusion test set for evaluating human

pose estimation approaches that handle partial occlusions from objects.

In chapter 5, we have proposed an efficient convolutional neural network (CNN) for 2D single person pose estimation from RGB images. The proposed CNN is an adaptation of the GoogleNet architecture into a fully convolutional neural network for human pose estimation. We have also proposed a data augmentation procedure that significantly improves the performance. The proposed CNN is trained with a binary cross entropy loss. The CNN predicts a dense belief map for every body joint. The belief map contains a per pixel likelihood for every body joint. The CNN runs efficiently on a mid-range graphical processing unit and is on par with recent state-of-the-art methods for 2D single person pose estimation.

In chapter 6, we have proposed a bottom up approach for multi-person pose estimation. Our approach simultaneously predicts body parts and relative offsets maps respectively. Given a root joint, the relative offsets are used to associate different body joints in a greedy manner by assuming a tree structure on the skeleton hierarchy. Our approach achieves comparable performance to recent state-of-the-art on the MS COCO dataset.

Finally in chapter 7, we have proposed a framework for correspondence matching. The proposed framework consists of an embedding network and an efficient matching network. The embedding network densely generates embeddings for query and target images. The matching network efficiently generates similarity heat-maps for all the query key points over the target image. The framework is trained end-to-end with a classification loss. The framework is applied to the tasks of semantic key points matching, dense matching and instance key points matching. The framework outperforms the existing state-of-the-art approaches. In the context of multi-person pose tracking, instance key points matching is used to associate body parts of a person over time in a video sequence.

8.2 Future Extensions

In this section we discuss potential future extensions for contributions made in this thesis.

End-to-End 3D Human Pose Estimation under Partial Occlusion. In chapter 4, we have used a regression forest framework with hand crafted features for inferring 3D body poses under partial occlusion. The current success in feature learning using convolutional neural networks (CNNs) has shifted the trend towards end-to-end learning methods. Exploring an end-to-end method for the task of 3D human pose estimation under partial occlusion from objects is an interesting future direction.

Handling rare poses. Recent 2D pose estimation methods work well for normal recurring body poses. The methods perform sub-optimally for a long tail of rare poses.

Recently zero shot or few shot learning has been used in object detection to learn appearance embeddings of objects. The appearance embeddings are used to detect rarely occurring or new objects. Exploring zero-shot or few shot learning to learn pose embeddings and then using them to detect rare poses is an interesting future work.

Learning temporal fusion for multi-person tracking. Multi-person pose tracking is a recent problem. Trend is to use existing still image multi-person pose estimation methods for joint detections in frames of a clip and then use graph based post-processing for spatio-temporal associations. Despite good performance, the temporal information is fused only during the post-processing phase and is not exploited during the learning phase. Exploring an approach that fuses spatial and temporal information during the learning phase is an interesting future direction.

Memory Networks for long term person association in Videos. Long term association is a sub-problem in multi-person pose tracking for associating body parts of a person over time. In this thesis, we have proposed to use correspondence matching applied to a pair of frames for body parts association over time. The approach does not take in to account the long term appearance of a person in the video. Recently memory networks have been introduced in the natural language processing community to exploit long term dependencies for several language processing tasks. They have performed better for tasks that benefit from long-term contextual dependencies when compared to RNNs and LSTMS. Additionally, they support out of order data processing compared to sequential processing supported by RNNs and LSTMs. Exploring memory networks for long term person association in videos is an interesting future work.

Pose Graphs based Image Retrieval. Existing methods focus on low level body parts detection and association from still images. In some applications it is desirable to find images with people with a specific interaction happening. Exploring approaches that take an image and output a pose graph for the image instead of the body poses of people in the image is a less explored direction.

3D Multi-person Pose Estimation. Current deep learning based methods focus on 2D poses of humans from images. Less attention has been given to 3D poses of humans from RGB images. Current limitations include lifting of 2D poses to 3D, limited datasets with ground-truth 3D poses available etc.. Estimation of 3D multi-person pose estimation from RGB data is an open and interesting challenge to explore.

Bibliography

Agarwal, A. and Triggs, B. (2006). Recovering 3d Human Pose from Monocular Image. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **28**(1), 44–58.

Andriluka, M., Roth, S., and Schiele, B. (2009). Pictorial Structures revisited: People Detection and articulated pose estimation. In *IEEE Conference on Computer Vision and Pattern Recognition*.

Andriluka, M., Roth, S., and Schiele, B. (2010). Monocular 3D Pose Estimation and Tracking by Detection. In *IEEE Conference on Computer Vision and Pattern Recognition*.

Andriluka, M., Pishchulin, L., Gehler, P., and Schiele, B. (2014). 2D Human Pose Estimation: New BenchMark. In *IEEE Conference on Computer Vision and Pattern Recognition*.

Baak, A., Müller, M., Bharaj, G., Seidel, H. P., and Theobalt, C. (2011). A data-driven approach for real-time full body pose reconstruction from a depth camera. In *International Conference on Computer Vision*.

Bay, H., Ess, A., Tuytelaars, T., and Gool., L. V. (2008). Speeded-up robust features (SURF). *Computer Vision and Image Understanding (CVIU)*.

Bo, L., Sminchisescu, C., Kanaujia, A., and Metaxas, D. (2008). Fast Algorithms for Large Scale Conditional 3d Prediction. In *IEEE Conference on Computer Vision and Pattern Recognition*.

Bourdev, L. and Malik, J. (2009). Poselets: Body part detectors trained using 3d human pose annotations. In *International Conference on Computer Vision*.

Bromley, J., Guyon, I., Lecun, Y., Sckinger, E., and Shah., R. (1994). Signature verification using a Siamese time delay neural network. In *Neural Information Processing Systems*.

Bulat, A. and Tzimiropoulos, G. (2016). Human pose estimation via convolutional part heatmapregression. In *ECCV*.

Burgos-Artizzu, X. P., Perona, P., and Dollar, P. (2013). Robust face landmark estimation under occlusion. In *International Conference on Computer Vision*.

Butler, D. J., Wulff, J., Stanley, G. B., and Black, M. J. (2012). A naturalistic open source movie for optical flow evaluation. In *ECCV*.

Cao, Z., Simon, T., Wei, S., and Sheikh, Y. (2017). Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields. In *IEEE Conference on Computer Vision and Pattern Recognition*.

Carreira, J., Agarwal, P., Fragkiadaki, K., and Malik, J. (2016). Human Pose Estimation with Iterative Error Feedback. In *IEEE Conference on Computer Vision and Pattern Recognition*.

Chen, C. and Remanan, D. (2017). 3d human pose estimation= 2d pose estimation+ matching. In *IEEE Conference on Computer Vision and Pattern Recognition*.

Chen, X. and Yuille, A. (2014). Articulated Pose Estimation by a Graphical Model with Image Dependent Pairwise Relations. In *Neural Information Processing Systems*.

Chen, Y., Wang, Z., Peng, Y., Zhang, Z., Yu, G., and Sun, J. (2018). Cascaded Pyramid Network for Multi-Person Pose Estimation . In *IEEE Conference on Computer Vision and Pattern Recognition*.

Choy, C. B., Gwak, J., Savarese, S., and Chandraker, M. (2016). Universal Correspondence Network. In *Neural Information Processing Systems*.

Chu, X., Yang, W., Ouyang, W., Ma, C., Yuille, A., and Wang, X. (2017). Multi-Context Attention for Human Pose Estimation. In *IEEE Conference on Computer Vision and Pattern Recognition*.

CMU Mocap (2008). http://mocap.cs.cmu.edu/.

Collobert, R., Kavukcuoglu, K., and Farabe, C. (2011). Torch7: A matlab-like environment for machine learning.

Dalal, N. and Triggs, B. (2005). Histograms of oriented gradients for human detection. In *IEEE Conference on Computer Vision and Pattern Recognition*.

Dantone, M., Gall, J., Fanelli, G., and van Gool, L. (2012). Real-time Facial Feature Detection using Conditional Regression Forests. In *IEEE Conference on Computer Vision and Pattern Recognition*.

Dantone, M., Gall, J., Leistner, C., and van Gool, L. (2013). Human Pose Estimation using Body Parts Dependent Joint Regressors. In *IEEE Conference on Computer Vision and Pattern Recognition*.

Everingham, M., Gool, L. V., I., C. K., Williams, Winn, J., and Zisserman, A. (2011). The PASCAL Visual Object Classes Challenge 2011 (VOC2011) Results. Technical report.

Fan, X., Zheng, K., Lin, Y., and Wang, S. (2015). Combining local appearance and holistic view: Dual Source deep neural networks for human pose estimation. In *IEEE Conference on Computer Vision and Pattern Recognition*.

Felzenswalb, P. and Huttenlocher, D. (2005). Pictorial Structures for object recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*.

Ganapathi, V., Plagemann, C., Koller, D., and Thrun, S. (2010). Real time motion capture using a single time-of-flight camera. In *IEEE Conference on Computer Vision and Pattern Recognition*.

Ganapathi, V., Plagemann, C., Koller, D., and Thrun, S. (2012). Real Time Human Pose Tracking from Range Data. In *European Conference on Computer Vision*.

Ghiasi, G., Yang, Y., Ramanan, D., and Fowlkes, C. C. (2014). Parsing Occluded People. In *IEEE Conference on Computer Vision and Pattern Recognition*.

Girdhar, R., Gkioxari, G., Torresani, L., and M. Paluri, D. T. (2017). Detect-and-Track: Efficient Pose Estimation in Videos.

Girshick, R., Shotton, J., Kohli, P., Criminisi, A., and Fitzgibbon, A. (2011a). Efficient Regression of General-Activity Human Poses from Depth Images. In *International Conference on Computer Vision*.

Girshick, R. B., Felzenszwalb, P. F., and McAllester, D. A. (2011b). Object Detection with Grammar Models. In *Neural Information Processing Systems*.

Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep Learning*. MIT Press. http://www.deeplearningbook.org.

Grest, D., Woetzel, J., and Koch, R. (2005). Nonlinear Body Pose Estimation from Depth Images. In *DAGM Annual Pattern Recognition Symposium*.

Ham, B., Cho, M., Schmid, C., and Ponce, J. (2016). Proposal flow. In *CVPR*.

Han, K., R., R. S., Ham, B., Wong, K.-Y. K., Cho, M., Schmid, C., and Ponce, J. (2017). Scnet: Learning semantic correspondence. In *ICCV*.

Hariharan, B., Arbelaez, P., Girshick, R., and Malik, J. (2015). Hypercolumns for Object Segmentation and Fine-grained Localization. In *IEEE Conference on Computer Vision and Pattern Recognition*.

Hawes, N. (2016). The strands project: Long-term autonomy in everyday environments.

He, K., Gkioxari, G., Dollr, P., and Girshick, R. (2017). Mask R-CNN. In *International Conference on Computer Vision*.

Hu, P. and Ramanan, D. (2016). Bottom Up and Top Down Reasoning with Hierarchical Rectified Gaussians. In *IEEE Conference on Computer Vision and Pattern Recognition*.

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

Insafutdinov, E., Andriluka, M., Pishchulin, L., S. Tang, E. L., Andres, B., and Schiele, B. (2017). Arttrack: Articulated multi-person tracking in the wild. In *IEEE Conference on Computer Vision and Pattern Recognition*.

Ioffe, S. and Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift.

Iqbal, U. and Gall, A. M. J. (2017). PoseTrack: Joint Multi-Person Pose Estimation and Tracking. In *IEEE Conference on Computer Vision and Pattern Recognition*.

Iqbal, U. and Gall, J. (2016). Multi-Person Pose Estimation with Local Joint-to-Person Associations. In *European Conference on Computer Vision*.

Iqbal, U., Milan, A., Andriluka, M., Ensafutdinov, E., Pishchulin, L., Gall, J., and .Schiele, B. (2017). PoseTrack: A benchmark for human pose estimation and tracking. *arXiv:1710.10000 [cs]*.

Jaderberg, M., Simonyan, K., Zisserman, A., and Kavukcuoglu., K. (2015). Spatial Transformer Networks. In *Neural Information Processing Systems*.

Johnson, S. and Everingham, M. (2010). Clustered Pose and Nonlinear Appearance Models for Human Pose Estimation. In *British Machine Vision Conference*.

Johnson, S. and Everingham, M. (2011). Learning Effective Human Pose Estimation from Inaccurate Annotation. In *IEEE Conference on Computer Vision and Pattern Recognition*.

Kanazawa, A., Jacobs, D. W., and Chandraker, M. (2016). WarpNet: Weakly Supervised Matching for Single-view Reconstruction. In *IEEE Conference on Computer Vision and Pattern Recognition*.

Kim, J., Liu, C., Sha, F., and Grauman., K. (2013). Deformable spatial pyramid matching for fast dense correspon- dences. In *CVPR*.

Kim, S., Min, D., Ham, B., Jeon, S., Lin, S., and Sohn, K. (2017). Fully convolutional self-similarity for dense semantic corre- spondence. In *CVPR*.

Kinect For Windows SDK 2.0 (2012). http://www.microsoft.com/en-us/kinectforwindows/develop/.

Kingma, D. and Ba, J. (2014). Adam: A Method for Stochastic Optimization.

Kostrikov, I. and Gall, J. (2014). Depth Sweep Regression Forests for Estimating 3D Human Pose from Images. In *British Machine Vision Conference*.

Li, S. and Chen, A. (2014). 3d human pose estimation from monocular images with deep convolutional neural network. In *Asian Conference on Computer Vision*.

Lin, G., Shen, C., van dan Hengel, A., and Reid, I. (2016). Efficient piece wise training of deep strucutred models for semantic segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition*.

Lin, T., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollr, P., and Zitnick, C. (2014). Microsoft coco: Common objects in context. In *European Conference on Computer Vision*.

Linder, T., Breuers, S., Leibe, B., and Arras, K. (2016). On Multi-Modal People Tracking from Mobile Platforms in Very Crowded and Dynamic Environments. In *IEEE International Conference on Robotics and Automation*.

Liu, C., Yuen, J., and Torralba, A. (2011). Sift flow: Dense correspondence across scenes and its applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Long, J., Zhang, N., and Darrell, T. (2014). Do convnets learn correspondence? In *Neural Information Processing Systems*.

Long, J., Shelhamer, E., and Darrel, T. (2015). Fully Convolutional Netowrks for Semantic Segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition*.

Lowe, D. G. (2004). Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision*, **60**(2).

Luo, W., Schwing, A. G., and Urtasun, R. (2016). Efficient Deep Learning for Stereo Matching. In *IEEE Conference on Computer Vision and Pattern Recognition*.

Menze, M. and Geiger, A. (2015). Object scene flow for autonomous vehicles. In *CVPR*.

Microsoft Corporation Redmond WA (2010). Kinect for Xbox 360.

Newell, A., Yang, K., and Deng, J. (2016). Stacked hourglass networks for human pose estimation. In *ECCV*.

Newell, A., Huang, Z., and .Deng, J. (2017). Associative Embedding: End-to-End Learning for Joint Detection and Grouping. In *Neural Information Processing Systems*.

Papandreou, G., Zhu, T., Kanazawa, N., A. Toshev, J. T., and nad K. Murphy, C. B. (2017a). Towards accurate multi-person pose estimation in the wild. In *IEEE Conference on Computer Vision and Pattern Recognition*.

Papandreou, G., Zhu, T., Kanazawa, N., and A. Toshev, J. Tompson, C. B. K. M. (2017b). Towards accurate multiperson pose estimation in the wild. In *IEEE Conference on Computer Vision and Pattern Recognition*.

Papandreou, G., Zhu, T., Chen, L., S. Gidaris, J. T., and Murphy, K. (2018). PersonLab: Person Pose Estimation and Instance Segmentation with a Bottom-Up, Part-Based, Geometric Embedding Model. In *European Conference on Computer Vision*.

Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L., and Lerer, A. (2017). Automatic differentiation in pytorch.

Pishchulin, L., Andriluka, M., Gehler, P., and Schiele, B. (2013a). Poselet Conditioned Pictorial Structures. In *IEEE Conference on Computer Vision and Pattern Recognition*.

Pishchulin, L., Andriluka, M., Gehler, P., and Schiele, B. (2013b). Strong appearance and expressive spatial models for human pose estimation. In *International Conference on Computer Vision*.

Pishchulin, L., Insafutdinov, E., Tang, S., Andres, B., Andriluka, M., Gehler, P., and Schiele, B. (2016). DeepCut : Joint Subset Partition and Labeling for Multi Person Pose Estimation. In *IEEE Conference on Computer Vision and Pattern Recognition*.

Plagemann, C., Ganapathi, V., Koller, D., and Thrun, S. (2010). Real-Time Identification and Localization of Body Parts from Depth Images. In *IEEE International Conference on Robotics and Automation*.

Poppe, R. (2007). Vision-based Human Motion Analysis. *Computer Vision and Image Understanding (CVIU)*, **108**(1-2), 4–18.

Poser (2010). http://my.smithmicro.com/.

Rafi, U., J.Gall, and Leibe, B. (2015). A Semantic Occlusion Model for Human Pose Estimation from a Single Depth Image. In *in CVPR 2015 Chalearn Looking at People Workshop*.

Rafi, U., I.Kostrikov, Gall, J., and Leibe, B. (2016). An Efficient Convolutional Network for Human Pose Estimation. In *British Machine Vision Conference*.

Ramakrishna, V., Munoz, D., Hebert, M., Bagnell, J., and Sheikh, Y. (2014). Articulated Pose Estimation via Inference Machines. In *European Conference on Computer Vision*.

Ren, S., He, K., Girshick, R., and Sun, J. (2015). Faster R-CNN: To- wards Real-Time object detection with region proposal net- works. In *Neural Information Processing Systems*.

Revaud, J., Weinzaepfel, P., Harchaoui, Z., and Schmid, C. (2016). DeepMatching: Hierarchical Deformable Dense Matching. *International Journal of Computer Vision*.

Sapp, B. and Taskar, B. (2013). MODEC : Multimodel Decomposable Models for Human Pose Estimation. In *IEEE Conference on Computer Vision and Pattern Recognition*.

Schroff, F., Kalenichenko, D., and Philbin, J. (2015). FaceNet: A Unified Embedding for Face Recognition and Clustering. In *IEEE Conference on Computer Vision and Pattern Recognition*.

Shotton, J., Fitzgibbon, A., Cook, M., Sharp, T., Finocchio, M., Moore, R., Kipman, A., and Blake, A. (2011). Real-time Human Pose Recognition in Parts from Single Depth Images. In *IEEE Conference on Computer Vision and Pattern Recognition*.

Sigal, L. and Black, M. (2006). Measure locally, reason globally:Occlusion-sensitive articulated pose estimation. In *IEEE Conference on Computer Vision and Pattern Recognition*.

Sun, M. and Savarese, S. (2011). Articulated part-based model for joint object detetion and pose estimation. In *International Conference on Computer Vision*.

Sun, M., Kohli, P., and Shotton, J. (2012). Conditional Regression Forests for Human Pose Estimation. In *IEEE Conference on Computer Vision and Pattern Recognition*.

Sweet Home 3D (2014). www.sweethome3d.com.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., and Rabinovich, A. (2015). Going Deeper with Convolutions. In *IEEE Conference on Computer Vision and Pattern Recognition*.

Taylor, J., Shotton, J., Sharp, T., and Fitzgibbon, A. (2012). The Vitruvian Manifold: Inferring Dense Correspondences for One-Shot Human Pose Estimation. In *IEEE Conference on Computer Vision and Pattern Recognition*.

Tian, Y., Zitnick, L., and Narasimhan, S. (2012). Exploring the spatial hierarchy of mixture models for human pose estimation. In *European Conference on Computer Vision*.

Tola, E., Lepetit, V., and Fua., P. (2010). DAISY: An Efficient Dense Descriptor Applied to Wide Baseline Stereo. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Tompson, J., Jain, A., LeCun, Y., and Bregler, C. (2014). Joint Training of a Convolutional Network and a Graphical Model for Human Pose Estimation. In *Neural Information Processing Systems*.

Tompson, J., Goroshin, R., Jain, A., LeCun, Y., and Bregler, C. (2015). Efficient Object Localization Using Convolutional Networks. In *IEEE Conference on Computer Vision and Pattern Recognition*.

Toshev, A. and Szegedy, C. (2014). DeepPose: Human Pose Estimation via Deep Neural Networks. In *IEEE Conference on Computer Vision and Pattern Recognition*.

Wah, C., Branson, S., Welinder, P., Perona, P., and Belongie, S. (2011). The Caltech-UCSD Birds-200-2011 Dataset. Technical report.

Wang, J., Song, Y., Leung, T., Rosenberg, C., Wang, J., Philbin, J., Chen, B., and Wu, Y. (2014). Learning fine-grained image similarity with deep ranking. In *IEEE Conference on Computer Vision and Pattern Recognition*.

Wang, T., He, X., and Barnes, N. (2013). Learning Structured Hough Voting for Joint Object Detection and Occlusion Reasoning. In *IEEE Conference on Computer Vision and Pattern Recognition*.

Wang, Y. and Mori, G. (2008). Multiple tree models for occlusion and spatial constraints in human. In *European Conference on Computer Vision*.

Wei, S., Ramakrishna, V., Kanade, T., and Sheikh, Y. (2016). Convolutional Pose Machines. In *IEEE Conference on Computer Vision and Pattern Recognition*.

Xiao, B., Wu, H., and Wei, Y. (2018). Simple Baselines for Human Pose Estimation and Tracking. In *European Conference on Computer Vision*.

Yang, H., Lin, W. Y., and Lu, J. (2014). DAISY filter flow: A generalized approach to discrete dense correspondences. In *IEEE Conference on Computer Vision and Pattern Recognition*.

Yang, W., Ouyang, W., Li, H., and Wang, X. (2016). End-to-End Learning of Deformable Mixture of Parts and Deep Convolutional Neural Networks for Human Pose Estimation. In *IEEE Conference on Computer Vision and Pattern Recognition*.

Yang, W., Li, S., Ouyang, W., Li, H., and Wang, X. (2017). Learning feature pyramids for human pose estimation. In *ICCV*.

Yang, Y. and Ramanan, D. (2011). Articulated Pose Estimation using Flexible Mixtures of Parts. In *IEEE Conference on Computer Vision and Pattern Recognition*.

Ye, M., Wang, X., Yang, R., Ren, L., and Pollefey, M. (2011). Accurate 3D pose estimation from a single depth image. In *International Conference on Computer Vision*.

Yu, X., Lin, Z., Brandt, J., and Metaxas, D. N. (2014). Consensus of Regression for Occlusion-Robust Facial Feature Localization. In *European Conference on Computer Vision*.

Zagoruyko, S. and Komodakis, N. (2015). Learning to Compare Image Patches via Convolutional Neural Networks. In *IEEE Conference on Computer Vision and Pattern Recognition*.

Zbontar, J. and LeCun, Y. (2015). Computing the stereo matching cost with a CNN. In *IEEE Conference on Computer Vision and Pattern Recognition*.

Zhou, T., Lee, Y. J., Yu, S. X., and Efros, A. A. (2015). Flowweb. In *CVPR*.