HOCHSCHULE
RAVENSBURG-WEINGARTEN
UNIVERSITY
OF APPLIED SCIENCES

Masters in Mechatronics

August 2020

Embedded Project Report

Human Pose Estimation on Tiago Simulation Environment

Submitted To:-

Prof. Dr.-Franz Brummer

Submitted By:-

Tapan Taranath Hegde (32961)

Contents

Chapter 1

Abstract

Realtime multi-person 2D pose estimation is a key part in facilitating machines to have an understanding of people in images and videos. In this project, a real time approach to detect the 2D pose of multiple people in a video is presented. To learn the associate body parts with individuals in an image the method of Part Affinity Fields (PAFs) is used. In previous works over PAFs, the body part location estimation was refined simultaneously across training stages. Here I have chosen a pre-trained model where a PAF-only refines rather than both PAF and body part location refinement. This results in a significant increase in both runtime performance and accuracy. The selected model presents the first combined body and foot keypoint detector, based on an internal annotated foot dataset that is publicly available. During the execution of the project I have also observed that the combined detector not only reduces the inference time compared to running them sequentially, but also maintains the accuracy of each body part point individually.

Keywords : 2D human pose estimation, real-time, multiple person, part affinity fields.

Chapter 2

Introduction

Being able to detect and recognize human activities is essential for several applications, including personal assistive robotics. For example, if a robot could watch and keep track of how often a person drinks water, it could prevent the dehydration of elderly by reminding them. True daily activities do not happen in structured environments (e.g., with closely controlled background), but in uncontrolled and cluttered households and offices.Due to its unstructured and often visually confusing nature,detection of daily activities becomes a much more difficult task. In this project, we are interested in reliably detecting sitting and standing activity that a person performs in a home environment.

Inferring the pose of multiple people in images presents a unique set of challenges. First, each image may contain an unknown number of people that can appear at any position or scale. Second, interactions between people induce complex spatial interference, due to contact, occlusion, or limb articulations, making association of parts difficult. Third, runtime complexity tends to grow with the number of people in the image, making real time performance a challenge.

2.1 Introduction to Part Affinity Fields

In this project, an efficient method for multi- person pose estimation with competitive performance on multiple public benchmarks is used. It is a bottom-up representation of association scores via Part Affinity Fields (PAFs), a set of 2D vector fields that encode the location and orientation of limbs over the image domain.

Figure 1 : **Top** : Multi-person pose estimation. **Bottom left**: PartAffinity F-ields (PAFs). **Bottom right**: 2D vector

The image Figure 1 gives a basic overview of the working methodology of PAF's. The Top image shows the Multi-person pose estimation. Body parts belonging to the same person are linked, including foot key-points (big toes, small toes, and heels). The Bottom left image shows the PartAffinity Fields (PAFs) corresponding to the limb connecting the elbow and wrist with color encoded orientation. The Bottom right image is a 2D vector in each pixel of every PAF. It encodes the position and orientation of the limbs. A more detailed explanation of the working of PAF's will be elaborated in the literature survey section.

Chapter 3

Literature Survey

3.1 Single Person Pose Estimation

The traditional approach to articulated human pose estimation is to perform inference over a combination of local observations on body parts and the spatial dependencies between them. The spatial model for articulated pose is either based on tree-structured graphical models which parametrically encode the spatial relationship between adjacent parts following a kinematic chain, or non-tree models that augment the tree structure with additional edges to capture occlusion, symmetry, and long range relationships. To obtain reliable local observations of body parts, Convolutional Neural Networks (CNNs) have been widely used, and have significantly boosted the accuracy on body pose estimation.

The convolutional pose machines architecture used a multi-stage architecture based on a sequential prediction framework iteratively incorporating global context to refine part confidence maps and preserving multimodal uncertainty from previous iterations. However, all of these methods assume a single person, where the location and scale of the person of interest is given.

3.2 Multi Person Pose Estimation

There is a large body of previous work on human activity recognition. A few common approaches are : use space-time features to model points of interest in video, filtering techniques and sampling of video patches, hierarchical dynamic Bayesian networks, activity recognition through the Hidden Markov Model (HMM).

Some have used a top-down strategy that first detects people and then have estimated the pose of each person independently on each detected region. Although this strategy makes the techniques developed for the single person case directly applicable, it not only suffers from early commitment on person detection, but also fails to capture the spatial dependencies across different people that require global inference.

However, we needed a pre-trained model which could not only have a camera application but a Robot application too. OpenPose represents the first real-time multi-person system to jointly detect human body, hand, facial, and foot keypoints (in total 135 keypoints) on single images.

- **Functionality:**

- 2D real-time multi-person keypoint detection:
    - 15 or 18 or 25-keypoint body/foot keypoint estimation. Running time invariant to number of detected people.
    - 6-keypoint foot keypoint estimation. Integrated together with the 25-keypoint body/foot keypoint detector.
    - Part affinity fields (PAFs), a representation consisting of a set of flow fields that encodes unstructured pairwise relationships between body parts of a variable number of people

3.3 Part Affinity Fields for Part Association

- When people crowd together—as they are prone to do—these midpoints are likely to support false associations. Part Affinity Fields (PAFs) address these limitations.
- They preserve both location and orientation information across the region of support of the limb.
- Each PAF is a 2D vector field for each limb. For each pixel in the area belonging to a particular limb, a 2D vector encodes the direction that points from one part of the limb to the other.
- Each type of limb has a corresponding PAF joining its two associated body parts.
- Consider a single limb shown in the figure 2. Let x vectors be the groundtruth positions of body parts j1 and j2 from the limb c for person k in the image.
- If a point p lies on the limb, the value at L is a unit vector that points from j1 to j2 ; for all other points, the vector is zero-valued.

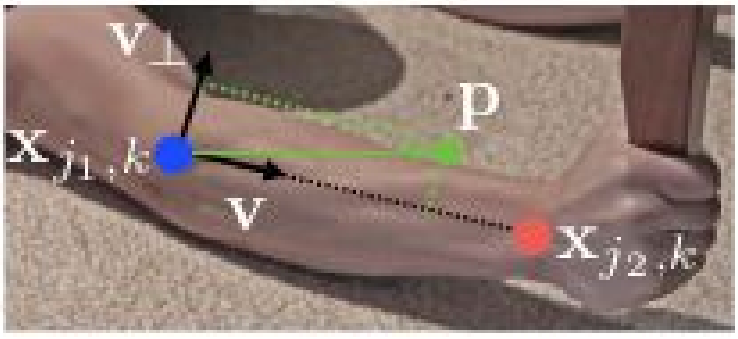

Figure 2 : PAF across a limb

3.4 Human Pose Estimation Technologies/Frameworks used:

- Multi stage CNN (Convolutional Neural Networks) :Convolutional Neural Networks(CNN) are one of the popular Deep Artificial Neural Networks. CNNs are made up of learnable weights and biases. CNNs are very similar to ordinary neural networks but not exactly the same.
- TensorFlow 1.4.1+(detects number of people, gender, position, age) : TensorFlow is an open source software library for numerical computation using data flow graphs. It is an ecosystem for developing deep learning models. It contains all the tools right from building to deployment
- Mobilenet-v2 architecture : It is TensorFlow's first mobile computer vision mode
- COCO data training set : COCO is a large-scale object detection, segmentation, and captioning dataset. COCO has several features: Object segmentation, Recognition in context, Superpixel stuff segmentation, 1.5 million object instances, 80 object categories, 250,000 people with keypoints, 5 captions per image

3.4.1 Dependencies

- ❏ Python3
- ❏ ROS
- ❏ OpenCV3
- ❏ Slidingwindow

Chapter 4

Algorithm Used

The Figure 3 illustrates the overall pipeline of the method. The system takes, as input, a color image of size w × h (Fig. 3a) and produces the 2D locations of anatomical key points for each person in the image (Fig. 3e).

-First, a feedforward network predicts a set of 2D confidence maps S of body part locations (Fig. 3b) and a set of 2D vector fields L of part affinity fields (PAFs),

-PAFs encode the degree of association between parts (Fig. 3c). We refer to part pairs as limbs for clarity, but some pairs are not human limbs (e.g., the face).

-Each image location encodes a 2D vector. Finally, the confidence maps and the PAFs are parsed by greedy inference (Fig. 3d) to output the 2D key points for all people in the image.

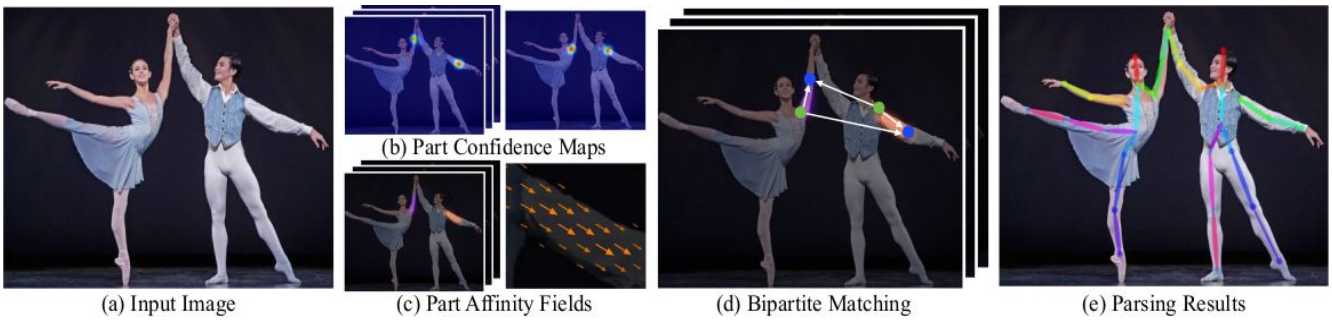

Figure 3: Overall pipeline. (a) Method takes the entire image as the input for a CNN to jointly predict (b) confidence maps for body part detection and (c) PAFs for part association. (d) The parsing step performs a set of bipartite matchings to associate body part candidates. (e) We finally assemble them into full body poses for all people in the image

4.1 Transition made in the code

The vector positions of the Hip, knee and foot points where subscribed to from a ROS Topic.

Distance between the Hip to the knee (a) and the knee to the foot (b) are calculated using the least square method.

Incase a = b (60% tolerance) then 'Person is Sitting' or else print 'Person is Standing'

```
def calcAngle(a, b):
    try:
        ax, ay = a
        bx, by = b
        if (ax == bx):
            return 1.570796
        return math.atan2(by-ay, bx-ax)
    except Exception as e:
        print("unable to calculate angle")
def CalcVectorAng(a,b,c):
    x1,y1=a
    x2,y2=b
    x3,y3=c
    result = math.atan2(y3 - y1, x3 - x1) - math.atan2(y2 -y1, x2 - x1)
    return result

def calcDistance(a,b): #calculate distance between two points.
    try:
        x1, y1 = a
        x2, y2 = b
        return math.hypot(x2 - x1, y2 - y1)
    except Exception as e:
        print("unable to calculate distance")

def Posture(all_peaks):
    g = 0
    if (all_peaks[9] and all_peaks[8]):
        g = 1
    try:
        if (all_peaks[10] and all_peaks[13] and all_peaks[8] and all_peaks[11]):
            f=0
            h=0
            if (all_peaks[8][0]<all_peaks[9][0]):
                f = 1
            if (all_peaks[11][0]<all_peaks[12][0]):
                h = 1
            rightankle = all_peaks[10][0:2]
            rightknee = all_peaks[9][0:2]
            leftankle = all_peaks[13][0:2]
            leftknee = all_peaks[12][0:2]
            hip_left = all_peaks[11][0:2]
            hip_right = all_peaks[8][0:2]
```

Figure 4 : Screenshot of code

Chapter 5

Evaluation methodology

Assessment was done initially by feeding in images to the ROS launch file and testing the results of our approximation. After several iterations with the code we could achieve a 100% accuracy, below are the images of the testing results.

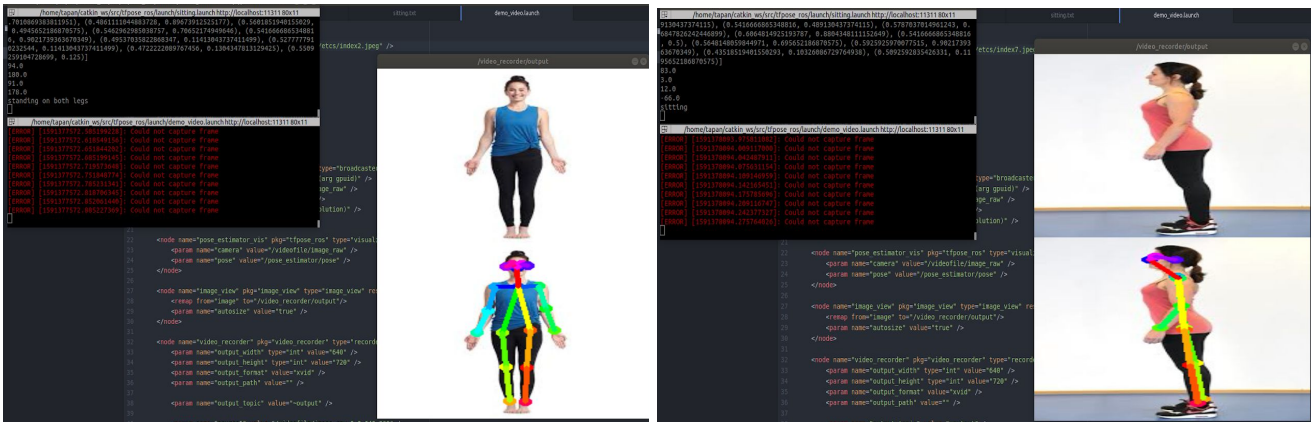

Figure 5 : Person Standing Evaluation

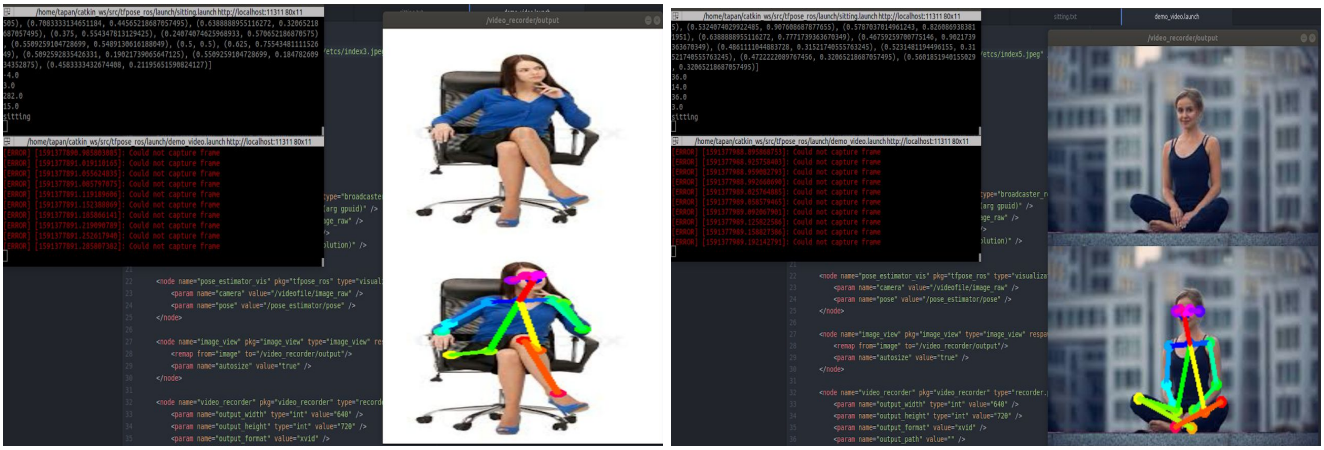

Figure 6 : Person Sitting Evaluation

Chapter 6

Results

The algorithm was ROS launched on Tiago Simulation environment (Gazebo simulation environment running on an Ubuntu computer, http://wiki.ros.org/Robots/TIAGo/Tutorials) with 3 humans inserted in the environment and the successful detection of human poses on a continuous video field was achieved.

Below is the video link of the results achieved :
https://drive.google.com/file/d/1ZmlmFNqaszR4L6nTYrAafEM7_2wpGmIR/view?usp=sharing

Main obstacles faced during the project implementation was integration with ROS as openpose is completely built on Python3 and ROS topics and messages on Python2. To solve this a CV bridge was created and subscribed to before launching the ROS files.

Recommend using CUDA awakened GPU's for better runtime, speed and accuracy.
Hardware used NVIDIA GeForce GTX-1080 Ti GPU.

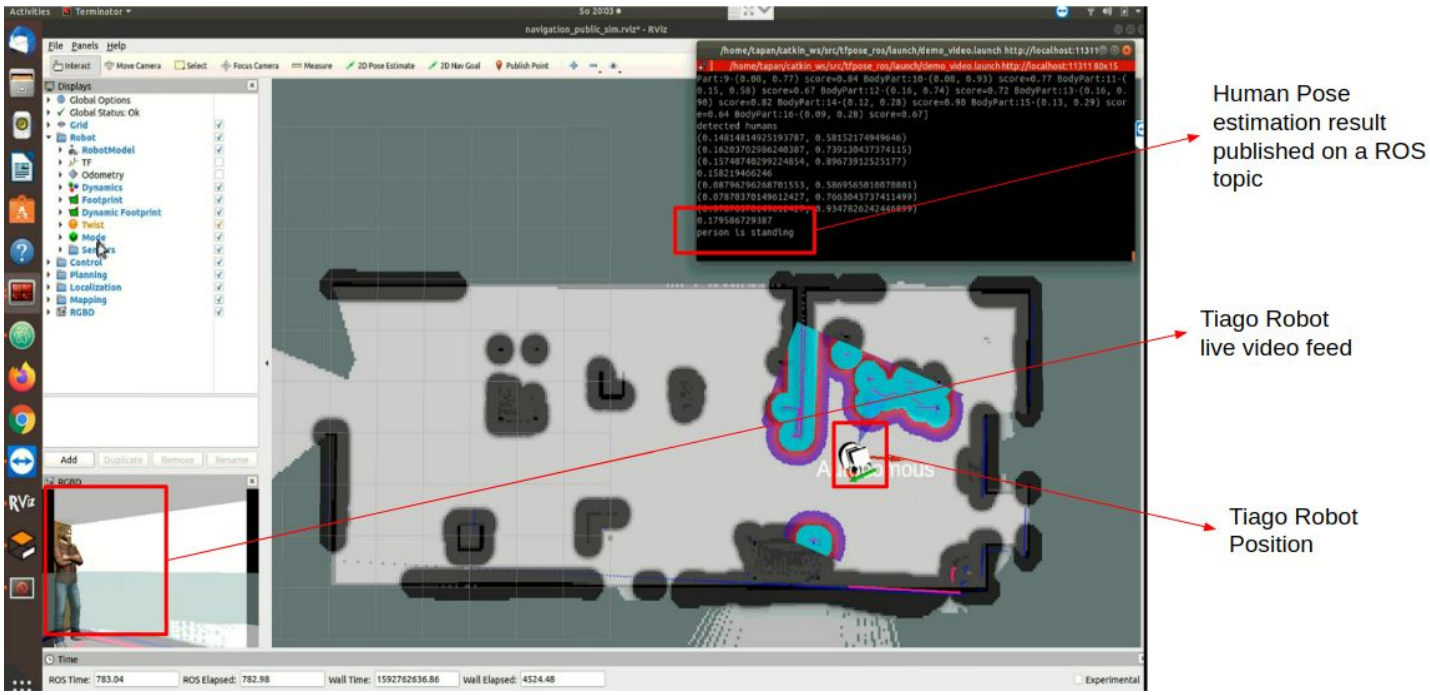

Figure 7 : Tiago Simulation Environment with the Robot detecting 'PERSON IS STANDING'

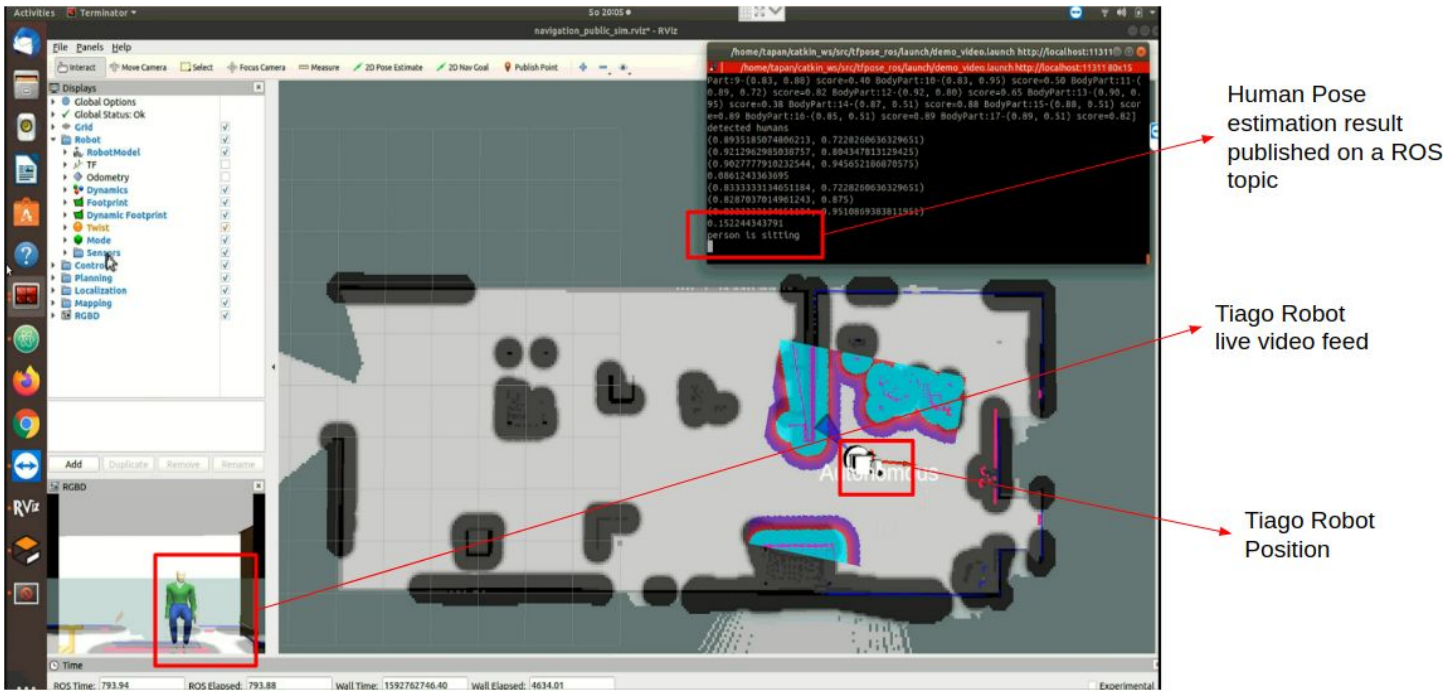

Figure 8 : Tiago Simulation Environment with the Robot detecting 'PERSON IS SITTING'

Chapter 7

Conclusion

Realtime multi-person 2D pose estimation is a critical component in enabling machines to visually understand and interpret humans and their interactions. In this project, we represent the key point association that encodes both position and orientation of human limbs. Second, the architecture jointly learns part detection and association. Third, we demonstrate that a simple algorithm is sufficient to produce high-quality results to conclude on Human Pose. Verifying the results on the Tiago Simulation environment has been of great service as practical implementation during the global pandemic of corona was not possible. For future work, we can use the algorithm to identify different actions of humans in a public area which can be of great use especially in the area of security.

Chapter 8

References

1. D. Mehta, S. Sridhar, O. Sotnychenko, H. Rhodin, M. Shafiei, H.P. Seidel, W. Xu, D. Casas, and C. Theobalt, "Vnect: Real-time 3d human pose estimation with a single rgb camera," in ACM TOG,2017
2. A. Newell, K. Yang, and J. Deng, "Stacked hourglass networks for human pose estimation," in ECCV, 2016
3. G. Gkioxari, B. Hariharan, R. Girshick, and J. Malik, "Using k-poselets for detecting people and localizing their keypoints," in CVPR, 2014
4. L. Pishchulin, E. Insafutdinov, S. Tang, B. Andres, M. Andriluka, P. Gehler, and B. Schiele, "Deepcut: Joint subset partition and labeling for multi person pose estimation," in CVPR, 2016
5. ildonet, https://github.com/ildoonet/tf-pose-estimation
6. B. Xiao, H. Wu, and Y. Wei, "Simple baselines for human pose estimation and tracking," in ECCV, 2018.
7. M. Eichner and V. Ferrari, "We are family: Joint pose estimation of multiple persons," in ECCV, 2010.
8. G.Hidalgo,Z.Cao,T.Simon,S.-E.Wei,H.Joo,andY.Sheikh,"OpenPoselibrary,"https://github.com/CMU-Perceptual-Computing-Lab/openpose.
9. M. Andriluka, L. Pishchulin, P. Gehler, and B. Schiele, "2D human pose estimation: new benchmark and state of the art analysis," in CVPR, 2014.