

Keypoint Communities

Duncan Zauss, Sven Kreiss, Alexandre Alahi
 EPFL VITA lab
 CH-1015 Lausanne
 duncan.zauss@epfl.ch

Abstract

We present a fast bottom-up method that jointly detects over 100 keypoints on humans or objects, also referred to as human/object pose estimation. We model all keypoints belonging to a human or an object –the pose– as a graph and leverage insights from community detection to quantify the independence of keypoints. We use a graph centrality measure to assign training weights to different parts of a pose. Our proposed measure quantifies how tightly a keypoint is connected to its neighborhood. Our experiments show that our method outperforms all previous methods for human pose estimation with fine-grained keypoint annotations on the face, the hands and the feet with a total of 133 keypoints. We also show that our method generalizes to car poses.

1. Introduction

Recent large-scale datasets with fine-grained annotations of complex poses present a new challenge for pose estimation methods. Beyond detecting a coarse person bounding box and a small set of keypoints for large body joints, we now have large datasets that include over 100 extra fine-grained keypoints in the face, the hands and the feet. Resolving these fine details will allow us to build robust representations of humans for downstream tasks like action recognition [24, 2], and intent prediction [25, 31].

Training our current pose estimation algorithms on poses with mixed coarse and fine keypoints presents a challenge as they assume a uniform importance of all keypoints in a pose. We introduce a principled keypoint weighting method to take into account the difference of the importance of coarse and fine-grained keypoints. Figure 1 shows a complex person and car poses. For instance, the person pose contains coarse keypoints like the hips and shoulders and fine-grained keypoints like the ones along the eyebrows.

In [12], Jin *et al.* share a large scale annotation for complex human body poses and propose their method, ZoomNet, that set the state-of-the-art for this type of complex



Figure 1: Our proposed keypoint weighting method for complex poses yields state-of-the-art results for whole body human pose estimation and for complex car poses, whilst running at high frame rates.

human body pose. Their method first localizes a person with their major keypoints and then estimates the areas of the hands and face. On the estimated areas, it runs a separate head that is zoomed-in on that area to determine fine-grained keypoint locations. In contrast, we propose a fast, bottom-up method that directly estimates all keypoints in parallel. Our method does not need predefined areas for fine-grained estimation and, therefore, generalizes to any pose, like a fine-grained car pose. Such a car pose is proposed in the ApolloCar3D dataset [35] and we show that our method generalizes to this pose as well.

Complex, coarse and fine-grained poses present a challenge for current pose estimation methods that assume a uniform distribution of keypoints across a person or object. A cluster of fine-grained keypoints overly emphasizes that region and focuses the neural network optimization on that area reducing the importance of other regions that only have a single keypoint. We propose a method that quantifies how tightly connected these keypoints are and that rebalances the training weights such that all areas of a pose are equally

well connected to the rest of the pose. We introduce the details in Section 3.

Our contributions are (i) a method to weigh the importance of keypoints and their connections in complex poses based on graph-based methods for community detection, (ii) an efficient implementation for fine-grained human poses and (iii) generalization from human poses to fine-grained car poses. We show the impact of our contribution with state-of-the-art results on the challenging COCO WholeBody dataset [12] and the ApolloCar3D dataset [35]. The software is open source and publicly available¹.

2. Related Work

There is an extensive literature on pose estimation. While many works have focused on human pose estimation, there are recent works that extend the method to animal pose estimation [23] and car pose estimation [35]. Recent datasets include more keypoints that represent finer details on human and car poses and are reviewed below.

Human Pose Estimation. The recent release of the COCO WholeBody dataset [12] with 133 keypoints for a single human pose presents new challenges for existing methods. The authors, Jin *et al.*, established baseline numbers of existing methods on their dataset and proposed ZoomNet, a new neural network architecture that refines regions with fine-grained annotations with dedicated networks.

In general, state-of-the-art methods for human pose estimation are based on Convolutional Neural Networks [37, 10, 6, 27, 29, 40, 36, 38, 28, 16, 7]. There are two major approaches for pose estimation. Bottom-up methods estimate each body joint first and then group them into poses. Top-down methods first run a person detector to estimate person bounding boxes before estimating body joint locations within each bounding box.

The first bottom-up methods were introduced, *e.g.*, by Pishchulin *et al.* with DeepCut [30]. They solve the keypoint association problem with an integer linear program. In these early methods, the processing time for a single image was of the order of hours. Newer methods introduced additional concepts to reduce prediction time, *e.g.*, in Part Affinity Fields [6], Associative Embedding [27], PersonLab [29] and multi-resolution networks with associate embedding [7]. Composite Fields as introduced in PifPaf [17] predict more precise associations than OpenPose’s Part Affinity Fields [6] and PersonLab’s mid-range fields [29] which allows for particularly fast and greedy decoding with high precision.

¹https://github.com/DuncanZauss/Keypoint_Communities

Car Pose Estimation. The new ApolloCar3D dataset [35] with its 66 keypoints for car pose estimators presents similar challenges than the WholeBody dataset [12]. The authors [35] presented baseline performance numbers using Convolutional Pose Machines (CPM) [38] and also quantified the performance of human labelers on their dataset.

It is only recently that methods that were developed for human pose estimation have been applied to other classes. Car poses provide finer detail for a car than a 2D or 3D detection bounding box. While human pose estimation focuses on the location of body joints within the human body, car poses annotate points on the surface of the car.

One of the earlier works by Reddy *et al.* proposes Occlusion-Net [32] that highlights the issue of self-occlusion for these keypoints on the surface of an object. As the car is viewed from different sides, the set of visible keypoints changes drastically due to self-occlusion. Their work includes extensive modeling with a 3D graph network and self-supervised training with the CarFusion dataset [8] to predict 2D and 3D keypoints. In OpenPose [5], Cao *et al.* show qualitative results for car pose estimation. Simple Baseline [33] trains a top-down pose estimator on car annotations of the Pascal3D+ dataset [39]. Other works choose different representations for finer details beyond bounding boxes. In GSNet [14], a car orientation in 3D space is predicted along with a 3D shape estimate.

Keypoint weighting In general, previous methods used uniform distributions to weigh the keypoints in training for single networks or used separate networks for the different fine-grained regions. We show that with our keypoint weighting method the performance of single neural networks for predicting poses that contain fine-grained and coarse features can be improved significantly.

3. Method

We need to devise a training procedure for poses that combine coarse keypoints that localize large body parts (hips, shoulders, *etc.*) and fine-grained keypoints like the outline of a hand. Individual fine-grained keypoints are highly predictable from neighboring keypoints whereas coarse keypoints are more independent. However, the importance of a keypoint is not only based on its individual predictability, but also on how it contributes to a local group of keypoints. In other words, every individual keypoint in a group of five keypoints might have negligible importance, the group of five keypoints together is still important.

3.1. Pose Estimation Architecture

Figure 2 provides an overview of our architecture. Our method is independent of the particular choice of pose estimator, and could be used for any top-down or bottom up

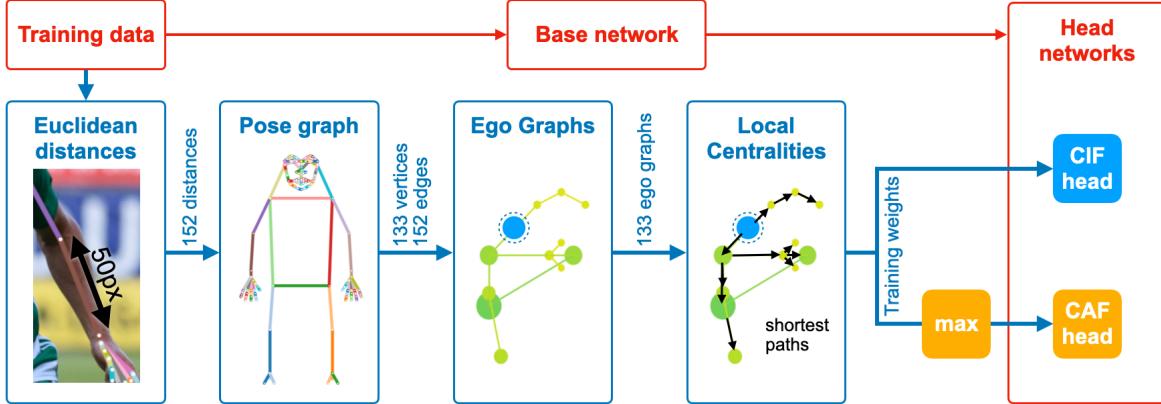


Figure 2: Overview of our method. We obtain the average euclidean distance for every connection in the pose graph from the training dataset. We then create ego graphs of radius three for every vertex and compute the local centrality for the ego vertex. The centrality is directly related to the training weight of a joint. The training weight of a connection is obtained by taking the max of the weights from the vertices of this connection.

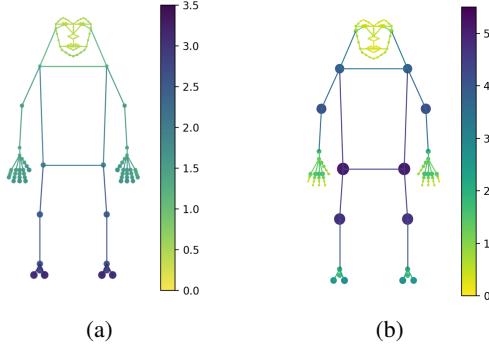


Figure 3: Visualization of proposed weighting for joints and connections. The colors indicate the training weights. For the joints, also the circle radius is proportional to the joint weight. In (a), all shortest paths are taken into account. In (b), only shortest paths within a radius of three are taken into account.

pose estimation algorithm. We will use OpenPifPaf [17], which is a bottom-up pose detector based on Composite Fields. A backbone in form of a ResNet [11] or ShuffleNetV2 [21] processes single images to create a common representation for the head networks. The head networks are the Composite Intensity Fields (CIFs) and Composite Association Fields (CAFs) that are 1×1 convolutions followed by subpixel convolutions [34]. The heads are trained to detect keypoints and to associate keypoints respectively. Per keypoint type k and for every location i, j in the output field, the CIF head predicts an intensity component $c_k^{i,j}$ to indicate a keypoint is nearby, a two-dimensional vector component $v_k^{i,j}$ to precisely regress to the keypoint location, the uncertainty of the location $b_k^{i,j}$ and a scale component

$s_k^{i,j}$ to estimate the size of a keypoint. The learnt scale of the keypoint $s_k^{i,j}$ depends on the size of that specific joint in the image and is used in the decoding step as the width of an un-normalized Gaussian convolution to create high-resolution confidence fields. Similarly, the CAF head also has an intensity component to indicate the vicinity of an association between two keypoints, two vector components that regress to the two keypoints instances to associate, and two scale components to estimate the two keypoint sizes. The CIF loss with an extension to weigh all loss components by keypoint type k with w_k is:

$$\mathcal{L}_{\text{CIF}} = \sum_k w_k \left[\sum_{m_{k,c}} \text{BCE}(c, \hat{c}) \right] \quad (1)$$

$$+ \sum_{m_{k,v}} \text{Laplace}(v, \hat{v}, \hat{b}) \quad (2)$$

$$+ \sum_{m_{k,s}} \text{Laplace}\left(1, \frac{\hat{s}}{s}, b_s\right) \quad (3)$$

where c, v, b and s are components of the composite field with suppressed indices (k, i, j) for the keypoint type and feature map location and where symbols with a hat indicate predicted quantities. With $m_{k,c}$, $m_{k,v}$ and $m_{k,s}$, we indicate keypoint specific masks over the feature map. BCE is a binary cross entropy loss with Focal loss extension [19] and Laplace is a linear regression loss for vector components that is attenuated by a predicted \hat{b} or a fixed b_s [15, 18]. The probabilistic interpretation of the loss function as the negative logarithm of a joint likelihood function requires that the three components of the loss are equally weighted with respect to each other. The CAF head is trained with an equivalent loss with two vector components (2) and two

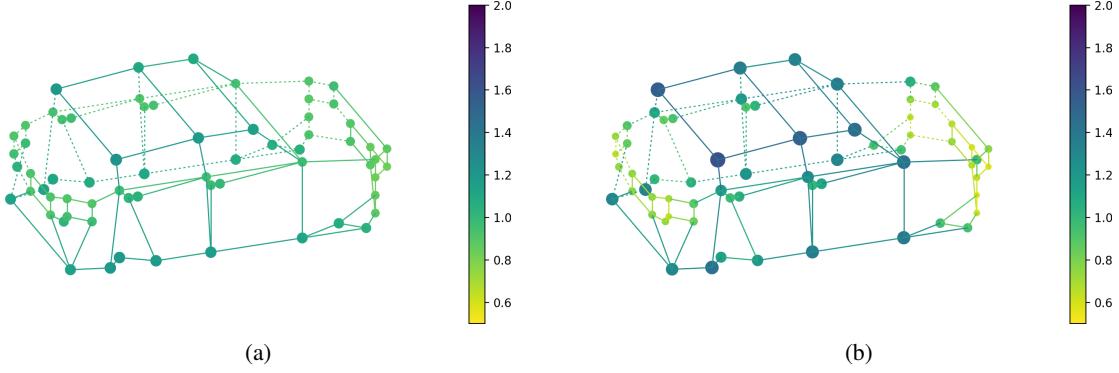


Figure 4: Visualization of proposed weighting for the skeleton of an instance from the ApolloCar3D dataset [35]. In (a), all shortest paths are taken into account. In (b), only shortest paths within a radius of three are taken into account. The dotted lines indicate connections on the left side of the car for a clearer visualization.

scale components (3). We now focus our attention on the blue branch of Figure 2 that determines the training weights w_k for the CIF and CAF heads.

3.2. Graph Centrality

We represent a pose as a graph $\{V, E\}$ with vertices V representing each keypoint, and edges E representing the Euclidean distance in the image plane between the keypoints. That Euclidean distance is estimated with an average over all training annotations. The importance of a keypoint does not only depend on its direct neighbors, but on its connectedness in the neighborhood. Therefore, we consider graph centrality measures.

There exists an enormous number of graph centrality measures, for example closeness centrality [1], eigenvector centrality [3], Katz centrality [13], betweenness centrality [9] and harmonic centrality [22] among others.

Traditionally, centrality measures are used to determine the centrality or importance of persons in social networks or more generally the importance of nodes in complex graphs. Highly central nodes get assigned a high centrality value. For our application highly central nodes are part of a community and thus we use the inverse of the centrality.

We want to use our understanding of the particular problem do derive the best metric for our use case. To train complex poses, we aim to make every keypoint equally well connected to the rest of the pose. Most centrality measures are based on shortest paths from a vertex to all other vertices. The “connectedness” of a vertex is represented in the average length of the shortest paths that originate at the vertex. For example, the ankle keypoint is not very well connected and the average length of all shortest paths that originate at the ankle is high as all the paths to the face and hand keypoints are long. To rebalance our training such that all vertices are connected equally well, we want to assign an equal weight to an average unit of distance in the shortest

paths.

In practice, we restrict the centrality computation to a neighborhood by extracting an ego graph of radius three (the subgraph with all the vertices around a particular vertex that can be reached in three steps) for every vertex and computing the centrality for that vertex only within that subgraph.

The weighted length (weighted by Euclidean distance) of the shortest path between two vertices v_1 and v_2 is $d(v_1, v_2)$. As we are interested in equally weighting a unit length, the harmonic average is appropriate. For every vertex v_i , we compute the harmonic average h of all the shortest paths originating at v_i :

$$h(v_i) = \left[\sum_{v_j \in V \setminus \{v_i\}} \frac{1}{d(v_i, v_j)} \right]^{-1} \quad (4)$$

where we can identify the harmonic centrality H [22] in the square brackets, leading to $h = H^{-1}$.

Numerically, the closeness centrality [1] and harmonic centrality are similar and it might be helpful to interpret this weighting in terms of assigning a high weight to keypoints with low “closeness”.

3.3. Training Weights

We use the graph centrality measure to derive training weights for keypoints and their connections. The keypoint weights are obtained directly from the centrality by normalizing $\sum_i h(v_i)$ to the number of keypoints. This normalization facilitates easier comparisons between the different weighting methods.

We also need to obtain training weights for keypoint connections. Again, the weight of the connection should not depend just on its own length, but also take into account the structure of the local cluster of keypoints this edge is a part

of. Given we already have a principled method for the vertices, we derive the weight w_{ij} for the edge that connects vertices v_i and v_j from h :

$$w_{ij} \propto \max(h(v_i), h(v_j)) . \quad (5)$$

We normalize the sum of edge weights to the total number of edges.

The resulting weights for the human WholeBody pose [12] and the car pose [35] are shown in Figures 3 and 4. We show two configurations of our method. One where we use the entire pose to compute our graph centrality measure and one where we use an ego graph of radius three. The WholeBody skeleton has clear hierarchical clusters of keypoints in the hands and face and one level down in the eyes and fingers. The ApolloCar3D skeleton is more uniformly distributed. There are keypoint agglomerations in the area of the lights and number plates both in the front and in the rear of the car. The keypoints in the roof are the most separate from the other keypoints. In contrast to the COCO WholeBody skeleton the communities are not as strongly separated. For the WholeBody pose, the computed weights range from 0.21 to 5.15 and for ApolloCar3D from 0.57 to 1.63. Our method automatically determines the keypoint communities for the WholeBody pose and produces highly varied training weights. Groups of keypoints that are highly predictable from each other receive a lower weight. Our method also successfully determines the more uniform distribution of keypoints in the car pose and produces less varied training weights.

For any generic pose and training dataset, this method automatically creates training weights for keypoints and associations in a principled way and we show its effectiveness on challenging experiments in the next section.

4. Experiments

We conduct extensive experiments on complex poses to demonstrate the effectiveness and efficiency of our method. We investigate a human pose with 17 COCO keypoints as the main skeleton that was then extended with an additional 116 keypoints for fine-grained details in the face, the hands and the feet. We also demonstrate that our method generalizes to a fine-grained car pose with 66 keypoints.

Datasets. For human pose estimation, we conduct experiments on the COCO WholeBody [12] dataset. This dataset contains extra annotations on the 64,000 training and 5,000 validation images of COCO [20] for face, hands and feet. The full pose contains 133 keypoints with 152 connections. There are about 130,000 instances with annotations for the left hand, the right hand and the face. The body annotations are taken from COCO [20] that contains about 250,000 instances.

Method	WB	body	foot	face	hand
HRNet [36]	43.2	65.9	31.4	52.3	30.0
ZoomNet [12]	54.1	74.3	79.8	62.3	40.1
AE [27]	27.4	40.5	7.7	47.7	34.1
OpenPose [6]	33.8	56.3	53.2	48.2	19.8
Ours	60.4	69.6	63.4	85.0	52.9

Table 1: Average precision (AP) results in percent on the COCO WholeBody dataset [12]. WB indicates evaluation on all 133 keypoints. The first two methods are top-down methods and the lower three are bottom-up methods. Reference numbers from [12].

For car pose estimation, we use the ApolloCar3D dataset [35]. It provides car annotations with 66 keypoints and we assigned them 108 connections. The dataset consists of 4283 training and 200 validation images with 52942 and 2674 annotated instances respectively. As cars are only visible from one side and often partially occlude each other, only an average of 16.2 keypoints are annotated per instance.

Evaluation. We follow the evaluation method proposed in the COCO WholeBody [12] dataset paper. It is based on keypoint-based average precision (AP) that was popularized with the COCO keypoint task [20]. The evaluation weighs every one of the 133 keypoints equally.

For the ApolloCar3D dataset, we report the detection rate which was proposed by Song *et al.* [35]. A keypoint is counted as detected if the distance from the prediction to the ground truth is less than 10 pixels. Additionally, we also report the keypoint-based AP as it is already common for human pose detection [20]. We compute AP based on object keypoint similarity with sigmas of 0.05 for all car keypoints.

Implementation Details. We extend OpenPifPaf [18] with an option to weigh the training of keypoint and connection types. We populate the weights for the given pose in the generic fashion described in Section 3.

We train models with ShuffleNetV2 [21] backbones that were pretrained without weighting the MS COCO keypoint task. The head networks CIF (Composite Intensity Field) and CAF (Composite Association Field) are single 1×1 convolutions followed by a subpixel convolution [34]. The total stride after the backbone is 16 and decreased to 8 in the head networks. We train for 100 epochs with a learning rate of 0.0001 with an SGD [4] optimizer with Nesterov momentum [26] of 0.95 and a batch size of 16.

Results on the COCO WholeBody dataset. Quantitative results on the COCO WholeBody dataset [12] are shown in



Figure 5: Qualitative results from the COCO WholeBody validation set [12]. Our method resolves multiple persons per image and captures their facial expressions and gestures like hailing a cab. The bottom-left image is processed with human and car pose estimators.



Figure 6: Qualitative results with images from the transport domain that we obtained from flickr. The two images in the bottom row were processed with our human and car pose estimation network.

Table 1. Our result is based on a single model that is evaluated for all (WB) or a subset of the predicted keypoints. Our method outperforms previous methods and achieves especially high precision on fine-grained regions such as the face or hands. The “body” task is equivalent to the COCO keypoint task on the val set [20]. Our method is based on OpenPifPaf [18] which only achieves 71.6% on the COCO val set with a model trained on that specific task and we therefore did not expect it to outperform ZoomNet [12]. Our method makes up for its lower body AP with excellent results for face and hand AP and is nearly twice as precise as any other bottom-up method.

Qualitative results are shown in Figure 6 and Figure 5. The additional keypoints in the face can serve as a powerful representation from which human emotions such as happiness and surprise but also attention and intent can be derived. The fine and coarse-grained WholeBody pose can be used to predict actions from images. For the example image on the bottom right of Figure 5, it can be predicted that the person is eating while holding a cell phone and for the person on the middle picture in the bottom row it can be predicted that the person is surprised that a cat jumped on the table. Through adding fine-grained hand keypoints the interaction between humans and objects can be detected. In the transportation domain, the fine grained keypoints on the hand region help to understand the intentions of pedestrians. For example in the bottom left image of Figure 6 it is possible to detect that the pedestrian wants to hail a cab and is not intending to cross the road even though she is standing at the side of the road.

Results on the ApolloCar3D dataset. Our method is not specific to human poses and can be applied to any pose. To demonstrate that our method generalizes, we apply it to the ApolloCar3D dataset [35], where every car instance is annotated with up to 66 keypoints. Our method achieves an average precision (AP) of 72.0% with all sigmas for the computation of the object keypoint similarity set to 0.05. The previous work [35] evaluates detection rate instead of AP and a comparison of our method with their Convolutional Pose Machines [38] evaluation and human annotators is shown in Table 2. We achieve a detection rate of 91.9% and thus outperform the previous state-of-the-art from [35] which achieved a detection rate of 75.4%. They also report the detection rate for human annotators at 92.4%. Our proposed method reduces the gap to human level performance from 17.0% to just 0.5%.

We share qualitative results on the validation set of ApolloCar3D [35] in Figure 7. We predict fine-grained keypoints for close-by and far-away cars. Cameras for self-driving technology cover a wide angle and therefore have to perceive small car instances even for cars at moderate distances. It is safety relevant to determine the locations of

Method	Detection rate[%]
Human annotators	92.4
CPM [38]	75.4
Ours	91.9

Table 2: Detection rate on the ApolloCar3D dataset [35]. The metrics for the Convolutional Pose Machines and the human annotators are from [35].

Method	WB	body	foot	face	hand
Baseline	55.3	60.8	54.2	86.2	50.6
Equal	-3.1%	-4.8%	-8.1%	+1.5%	+0.8%
Global	-1.2%	-2.5%	-1.1%	+0.8%	+2.6%
Crafted	-1.2%	-1.6%	-1.7%	+1.2%	-1.0%

Table 3: Ablation studies. Average precision (AP) results in percent on the COCO WholeBody val dataset [12]. We report the percentage gain of other weighting methods in comparison to our baseline method. In addition to our main method, we compare with applying no weighting (Equal), applying our method to the global graph instead of ego graphs (Global) and using hand-crafted weights (Crafted). All results are produced with a ShuffleNetV2k16 backbone. WB indicates an evaluation on all 133 keypoints.

break and indicator lights for downstream tasks which we achieve with high precision.

Ablation Studies. We study the effects of different loss weightings on the average precision (AP). In Table 3 the results from training with different weighting schemes are shown. For this ablation study we train for 50 epochs starting from a model that was pretrained on the 133 keypoints without weighting. “Crafted” means that the body and foot keypoints are weighted three times higher than the rest of the keypoints. Hand-crafting the weights can be seen as choosing 133 additional hyperparameters which is generally infeasible and the motivation for our method.

The weighting based on the local harmonic centrality, which is our baseline method, achieves an AP of 55.3%, which is a significant improvement over the unweighted training, which results in an AP of 53.6%. The weighting based on the local harmonic centrality also achieves a higher AP than the weighting based on the vanilla harmonic centrality and the hand-crafted weights. This shows a) that the local influence between the keypoints is more important than the global one and b) that using our proposed centrality measure is more optimal than hand-crafting the weights.

We study the effect of different parameter choices on the precision and prediction time trade-off of our method. The runtime of our method is influenced by the decoding algorithm. Since the WholeBody pose has 133 joints and 152



Figure 7: Qualitative results from the ApolloCar3D validation set [35]. We demonstrate excellent detection rates and spatial localizations of all the visible car keypoints even at far distances.

Seed thr.	CAF thr.	AP	t [ms]	t_{dec} [ms]
ZoomNet [12]		54.1	175	-
0.2	0.001	60.4	153	60
0.5	0.001	58.4	120	27
0.5	0.01	54.6	112	20

Table 4: Ablation studies. Average precision (AP) results in percent on the COCO WholeBody val dataset [12] and their associated prediction time for different decoding methods. We show ZoomNet’s average precision and runtime for better comparison with our method. Our neural network runs in 93ms on a NVIDIA GTX 1080 Ti. The decoding starts with the joints where the confidence exceeds the seed threshold. The vector and scale from a cell of a CAF field is only used if the confidence is above the CAF threshold.

associations the duration of the decoding is more prevalent than for poses with a lower number of joints and associations. The first step in the decoding process is to determine seed joints from which the decoding starts and from which connections to the other joints will be created with the help of the CAF fields. All joints that have a confidence that is higher than a certain seed threshold will be used as seed joints. Increasing the seed threshold will reduce the number of seeds and thus cause a faster decoding process. However, with a higher seed threshold, some humans may not be detected which can result in a lower average precision. Using a higher CAF threshold generally increases the

decoding speed in exchange for a lower accuracy as fewer associations are considered for decoding. The quantitative results that the variation of these parameters yield can be seen in Table 4. With our standard decoder setting, we already achieve a higher AP than ZoomNet [12] whilst being 22 ms faster. With a seed threshold of 0.5 and a CAF threshold of 0.01 our model achieves an AP of 54.5 with a prediction time of 112 ms, which is an excellent trade-off between inference speed and precision. Our ShuffleNetV2K16 backbone achieves an AP of 50.9 at a frame rate of 15.2 frames per second on a NVIDIA GTX 1080 Ti, which makes it suitable for most real-time applications that require fine-grained pose estimation.

5. Conclusion

We have proposed a generic and principled method to train complex poses with fine and coarse-grained details. Our experiments demonstrate our ability to perceive detailed facial expressions and hand gestures and produce state-of-the-art results on standard pose benchmarks for human and car poses. We have shown that our method operates at state-of-the-art prediction speeds and we have studied the trade-offs between accuracy and prediction speeds.

6. Acknowledgments

This project has received funding from the Initiative for Media Innovation based at Media Center, EPFL, Lausanne, Switzerland.

References

- [1] Alex Bavelas. Communication patterns in task-oriented groups. *The journal of the acoustical society of America*, 22(6):725–730, 1950. 4
- [2] Lorenzo Bertoni, Sven Kreiss, and Alexandre Alahi. Perceiving humans: from monocular 3d localization to social distancing. *IEEE Transactions on Intelligent Transportation Systems*, 2021. 1
- [3] Phillip Bonacich. Power and centrality: A family of measures. *American journal of sociology*, 92(5):1170–1182, 1987. 4
- [4] Léon Bottou. Large-scale machine learning with stochastic gradient descent. In *Proceedings of COMPSTAT’2010*, pages 177–186. Springer, 2010. 5
- [5] Zhe Cao, Gines Hidalgo Martinez, Tomas Simon, Shih-En Wei, and Yaser A Sheikh. Openpose: realtime multi-person 2d pose estimation using part affinity fields. *IEEE transactions on pattern analysis and machine intelligence*, 2019. 2
- [6] Zhe Cao, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Realtime multi-person 2d pose estimation using part affinity fields. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7291–7299, 2017. 2, 5
- [7] Bowen Cheng, Bin Xiao, Jingdong Wang, Honghui Shi, Thomas S Huang, and Lei Zhang. Higherhrnet: Scale-aware representation learning for bottom-up human pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5386–5395, 2020. 2
- [8] N Dinesh Reddy, Minh Vo, and Srinivasa G Narasimhan. Carfusion: Combining point tracking and part detection for dynamic 3d reconstruction of vehicles. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1906–1915, 2018. 2
- [9] Linton C Freeman. A set of measures of centrality based on betweenness. *Sociometry*, pages 35–41, 1977. 4
- [10] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Computer Vision (ICCV), 2017 IEEE International Conference on*, pages 2980–2988. IEEE, 2017. 2
- [11] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016. 3
- [12] Sheng Jin, Lumin Xu, Jin Xu, Can Wang, Wentao Liu, Chen Qian, Wanli Ouyang, and Ping Luo. Whole-body human pose estimation in the wild. In *European Conference on Computer Vision (ECCV)*, 2020. 1, 2, 5, 6, 7, 8
- [13] Leo Katz. A new status index derived from sociometric analysis. *Psychometrika*, 18(1):39–43, 1953. 4
- [14] Lei Ke, Shichao Li, Yanan Sun, Yu-Wing Tai, and Chi-Keung Tang. Gsnet: Joint vehicle pose and shape reconstruction with geometrical and scene-aware supervision. In *European Conference on Computer Vision (ECCV)*, pages 515–532. Springer, 2020. 2
- [15] Alex Kendall and Yarin Gal. What uncertainties do we need in bayesian deep learning for computer vision? In *Advances in neural information processing systems*, pages 5574–5584, 2017. 3
- [16] Muhammed Kocabas, Salih Karagoz, and Emre Akbas. Multiposenet: Fast multi-person pose estimation using pose residual network. In *European Conference on Computer Vision (ECCV)*, pages 417–433, 2018. 2
- [17] Sven Kreiss, Lorenzo Bertoni, and Alexandre Alahi. Pifpaf: Composite fields for human pose estimation. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 2, 3
- [18] Sven Kreiss, Lorenzo Bertoni, and Alexandre Alahi. Open-PifPaf: Composite Fields for Semantic Keypoint Detection and Spatio-Temporal Association. *IEEE Transactions on Intelligent Transportation Systems*, 2021. 3, 5, 7
- [19] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision (ICCV)*, pages 2980–2988, 2017. 3
- [20] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European Conference on Computer Vision (ECCV)*, pages 740–755. Springer, 2014. 5, 7
- [21] Ningning Ma, Xiangyu Zhang, Hai-Tao Zheng, and Jian Sun. Shufflenet v2: Practical guidelines for efficient cnn architecture design. In *European Conference on Computer Vision (ECCV)*, pages 116–131, 2018. 3, 5
- [22] Massimo Marchiori and Vito Latora. Harmony in the small-world. *Physica A: Statistical Mechanics and its Applications*, 285(3-4):539–546, 2000. 4
- [23] Alexander Mathis, Pranav Mamidanna, Kevin M Cury, Taiga Abe, Venkatesh N Murthy, Mackenzie Weygandt Mathis, and Matthias Bethge. Deeplabcut: markerless pose estimation of user-defined body parts with deep learning. Technical report, Nature Publishing Group, 2018. 2
- [24] Sina Mokhtarzadeh, Mina Ghadimi, Ahmad Nickabadi, and Alexandre Alahi. Convolutional relational machine for group activity recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 1
- [25] Taylor Mordan, Matthieu Cord, Patrick Perez, and Alexandre Alahi. Detecting 32 pedestrian attributes for autonomous vehicles. *IEEE Transactions on Intelligent Transportation Systems - under review*, 2020. 1
- [26] Yurii Nesterov. A method of solving a convex programming problem with convergence rate $O(1/k^2)$. In *Soviet Mathematics Doklady*, volume 27, pages 372–376, 1983. 5
- [27] Alejandro Newell, Zhiao Huang, and Jia Deng. Associative embedding: End-to-end learning for joint detection and grouping. In *Advances in Neural Information Processing Systems*, pages 2277–2287, 2017. 2, 5
- [28] Alejandro Newell, Kaiyu Yang, and Jia Deng. Stacked hourglass networks for human pose estimation. In *European Conference on Computer Vision (ECCV)*, pages 483–499. Springer, 2016. 2
- [29] George Papandreou, Tyler Zhu, Liang-Chieh Chen, Spyros Gidaris, Jonathan Tompson, and Kevin Murphy. Personlab: Person pose estimation and instance segmentation with a

- bottom-up, part-based, geometric embedding model. In *European Conference on Computer Vision (ECCV)*, pages 269–286, 2018. 2
- [30] Leonid Pishchulin, Eldar Insafutdinov, Siyu Tang, Bjoern Andres, Mykhaylo Andriluka, Peter V Gehler, and Bernt Schiele. Deepcut: Joint subset partition and labeling for multi person pose estimation. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4929–4937, 2016. 2
- [31] Haziq Razali and Alexandre Alahi. Pedestrian intention prediction: A convolutional bottom-up approach. *Transportation Research Part C*, 2021. 1
- [32] N Dinesh Reddy, Minh Vo, and Srinivasa G Narasimhan. Occlusion-net: 2d/3d occluded keypoint localization using graph networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7326–7335, 2019. 2
- [33] Héctor Corrales Sánchez, Antonio Hernández Martínez, Rubén Izquierdo Gonzalo, Noelia Hernández Parra, Ignacio Parra Alonso, and David Fernandez-Llorca. Simple baseline for vehicle pose estimation: Experimental validation. *IEEE Access*, 8:132539–132550, 2020. 2
- [34] Wenzhe Shi, Jose Caballero, Ferenc Huszár, Johannes Totz, Andrew P Aitken, Rob Bishop, Daniel Rueckert, and Zehan Wang. Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1874–1883, 2016. 3, 5
- [35] Xibin Song, Peng Wang, Dingfu Zhou, Rui Zhu, Chenye Guan, Yuchao Dai, Hao Su, Hongdong Li, and Ruigang Yang. ApolloCar3d: A large 3d car instance understanding benchmark for autonomous driving. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5452–5462, 2019. 1, 2, 4, 5, 7, 8
- [36] Ke Sun, Bin Xiao, Dong Liu, and Jingdong Wang. Deep high-resolution representation learning for human pose estimation. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5693–5703, 2019. 2, 5
- [37] Alexander Toshev and Christian Szegedy. Deeppose: Human pose estimation via deep neural networks. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1653–1660, 2014. 2
- [38] Shih-En Wei, Varun Ramakrishna, Takeo Kanade, and Yaser Sheikh. Convolutional pose machines. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4724–4732, 2016. 2, 7
- [39] Yu Xiang, Roozbeh Mottaghi, and Silvio Savarese. Beyond pascal: A benchmark for 3d object detection in the wild. In *Proceeding of the IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 75–82. IEEE, 2014. 2
- [40] Bin Xiao, Haiping Wu, and Yichen Wei. Simple baselines for human pose estimation and tracking. In *Proceedings of the European conference on computer vision (ECCV)*, pages 466–481, 2018. 2