



HAL
open science

Multitask Metamodel for Keypoint Visibility Prediction in Human Pose Estimation

Romain Guesdon, Carlos F Crispim-Junior, Laure Tougne

► **To cite this version:**

Romain Guesdon, Carlos F Crispim-Junior, Laure Tougne. Multitask Metamodel for Keypoint Visibility Prediction in Human Pose Estimation. International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISAPP), Feb 2022, Virtual Conference, France. 10.5220/0010831200003124 . hal-03471147

HAL Id: hal-03471147

<https://hal.science/hal-03471147v1>

Submitted on 8 Dec 2021

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Multitask Metamodel for Keypoint Visibility Prediction in Human Pose Estimation

Romain Guesdon, Carlos Crispim-Junior, and Laure Tougne

Univ Lyon, Lyon 2, LIRIS UMR 5205

Lyon, France, F-69676

{romain.guesdon, carlos.crispim-junior, laure.tougne} @liris.cnrs.fr

Abstract: The task of human pose estimation (HPE) aims to predict the coordinates of body keypoints in images. Even if nowadays, we achieve high performance on HPE, some difficulties remain to be fully overcome. For instance, a strong occlusion can deceive the methods and make them predict false-positive keypoints with high confidence. This can be problematic in applications that require reliable detection, such as posture analysis in car-safety applications. Despite this difficulty, actual HPE solutions are designed to always predict coordinates for each keypoint. To answer this problem, we propose a new metamodel that predicts both keypoints coordinates and their visibility. Visibility is an attribute that indicates if a keypoint is visible, non-visible, or not labeled. Our model is composed of three modules: the feature extraction, the coordinate estimation, and the visibility prediction modules. We study in this paper the performance of the visibility predictions and the impact of this task on the coordinate estimation. Baseline results are provided on the COCO dataset. Moreover, to measure the performance of this method in a more occluded context, we also use the driver dataset DriPE. Finally, we implement the proposed metamodel on several base models to demonstrate the general aspect of our metamodel.

1 INTRODUCTION

Human Pose Estimation (HPE) is the task that aims to locate body keypoints on images. These keypoints can be body joints (shoulders, elbows, hips, ankles, etc.) or facial markers (eyes, ears, nose). Additional keypoints on the face, hands or feet are sometimes used (Hidalgo et al., 2019; Cao et al., 2019).

One of the difficulties of HPE is handling keypoints occlusion. Even if recent solutions have been able to reach high performance, state-of-the-art datasets depict many pictures with few occlusion, especially in pictures presenting one person (Andriluka et al., 2014; Lin et al., 2015). In contrast, in some specific contexts like crowds or narrow spaces, body parts have a high probability of being occluded or getting out of the field of view.

Strong occlusion can lead the network to predict with high confidence keypoints that are not annotated, as we can see in Figure 1. Furthermore, the networks may predict many false-positive keypoints (Guesdon et al., 2021), which can be problematic in applications where reliable predictions with significant precision are required, e.g., for action recognition or driver’s posture analysis (Das et al., 2017; Zhao

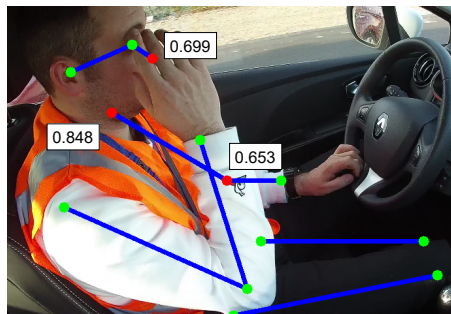


Figure 1: HPE prediction. Red points represent false positives, i.e., keypoints that were predicted even if not annotated due to strong occlusion. Confidence scores are provided in the boxes (maximum score = 1.0).

et al., 2020b). Despite the difficulty caused by occlusion, actual HPE networks are designed to predict coordinates for each keypoints during inference, even if the keypoint is outside of the image. Networks usually predict a confidence score; however, it covers the confidence of both the presence and the coordinates of the keypoints. Therefore, this score cannot be used to properly distinguish keypoints that the network could consider as absent from the image.

State-of-the-art datasets provide visibility labels, an attribute that depicts the perceptibility of each key-

36 point. A labeled keypoint can be visible, or non- 83
37 visible when the keypoint is lightly occluded but with 84
38 enough information to be located. If the keypoint is 85
39 heavily occluded or out of the field of view, it is not 86
40 labeled. However, state-of-the-art networks do not 87
41 consider these visibility labels. Furthermore, the few 88
42 existing methods using visibility only consider binary 89
43 visibility, *i.e.*, labeled or non-labeled keypoints (Stoffl 90
44 et al., 2021; Kumar et al., 2020).

45 This paper proposes a novel HPE metamodel ¹ 92
46 that can predict both the visibility and the coordinates 93
47 of the keypoints. Our solution can be implemented 94
48 with most of the deep-learning HPE methods and al- 95
49 lows these base models to predict keypoint visibil- 96
50 ity. The model can predict the three classes of labels, 97
51 which provides a finer description of the keypoint vis- 98
52 ibility. 99

53 This paper is organized as follows. We present in 101
54 Section 2 the related work on human pose estimation 102
55 and visibility prediction. Section 3 presents our meta- 103
56 model and its detailed architecture, especially the vis- 104
57 ibility module. We describe in Section 4 the details 105
58 about the experiments, and present the results in Sec- 106
59 tion 5. Finally, we discuss in Section 6 our conclu- 107
60 sions and future work. 108

61 2 RELATED WORK 109

62 This section presents existing work on human pose 114
63 estimation and visibility keypoints prediction. 115

64 The task of human pose estimation is divided into 116
65 two categories. Single-person HPE focuses on the de- 117
66 tection in pictures presenting one person, in opposi- 118
67 tion to multiperson detection. The first approach to 119
68 solve single-person HPE using deep learning was pro- 120
69 posed in (Toshev and Szegedy, 2014). This solution is 121
70 based on the deep architecture AlexNet (Krizhevsky 122
71 et al., 2012), which is used to estimate and refine 123
72 the coordinates. An Iterative Error Feedback net- 124
73 work was proposed in (Carreira et al., 2016) based 125
74 on the convolutional network GoogleNet (Szegedy 126
75 et al., 2015). The authors of (Sun et al., 2017) used 127
76 ResNet50 (He et al., 2016) to predict a parametrized 128
77 bones representation. However, all these methods try 129
78 to directly predict the keypoints coordinates from the 130
79 images, which affects the robustness of these methods 131
80 due to the high non-linearity of this approach. Other 132
81 solutions categorized as detection-based methods aim 133
82 to predict 2D matrices called heatmaps where each 134

¹Source code is publicly available on: https://gitlab.liris.cnrs.fr/aura_autobehave/vis-pred 135

pixel represents the probability for a joint to be lo- 83
84 cated here. The work of (Newell et al., 2016) pro- 85
86 posed an hourglass module that can be stacked to pre- 87
88 dict and refine features at several scales, which has in- 89
90 spired many other works (Chu et al., 2017; Ke et al., 91
92 2018; Tang and Wu, 2019; Tang et al., 2018). Besides 93
94 hourglass architectures, other detection-based meth- 95
96 ods have been proposed. The architecture in (Chen 97
98 et al., 2017) combines a heatmap generator with two 99
100 discriminators. Simple Baseline (Xiao et al., 2018), 101
102 is an architecture based on the ResNet network (He 103
104 et al., 2016) with a deconvolution stage to generate 105
106 the final heatmaps. Finally, Unipose (Artacho and 107
108 Savakis, 2020) combines atrous and cascade convo- 109
110 lutions to produce a multi-scale representation. 110

111 In addition to finding the keypoints in the picture, 111
112 multiperson HPE brings a new difficulty: to associate 112
113 the different persons to the detected keypoints. State- 113
114 of-the-art performance is achieved by methods called 114
115 top-down approaches that first detect the subjects in 115
116 the picture and then locate the keypoints for each per- 116
117 son individually. These methods usually combine a 117
118 person detector with a single-person HPE architec- 118
119 ture (Xiao et al., 2018; Sun et al., 2019; Lin et al., 119
120 2017; Cai et al., 2020; Li et al., 2019). Conversely, 120
121 the bottom-up approaches first detect every keypoints 121
122 in the image before associating them to form people 122
123 instances (Newell et al., 2017; Cao et al., 2017; Nie 123
124 et al., 2018). Top-down approaches tend to outper- 124
125 form the bottom-up methods while taking advantage 125
126 of both state-of-the-art person detectors and HPE ar- 126
127 chitectures. 127

128 Among top-down methods, the Simple Baseline 128
129 (SBI) network (Xiao et al., 2018) presents competi- 129
130 tive performance while preserving a small size, which 130
131 makes it practical for modifications and tests. In addi- 131
132 tion, it can be used for multiperson HPE by combin- 132
133 ing it with a person detector. 133

134 Recent work on human pose estimation has mainly 134
135 focused on improving the prediction of the keypoints' 135
136 coordinates. Therefore, methods which estimate the 136
137 visibility of HPE keypoints are scarce. In (Zhao et al., 137
138 2020a), visibility prediction is used to propose a new 138
139 evaluation method for multiperson pose estimation in 139
140 heavily occluded contexts. Visibility is predicted as 140
141 an occlusion score and is used to compute a metric 141
142 that highlights the performance of the evaluated net- 142
143 works on occluded points. The multi-instance HPE 143
144 network in (Stoffl et al., 2021) uses transformers to 144
145 predict keypoint visibility, which serves as a second- 145
146 ary task for end-to-end training. Besides, keypoint 146
147 visibility is predicted in (Kumar et al., 2017; Kumar 147
148 et al., 2020) as an annex task for face detection. 148

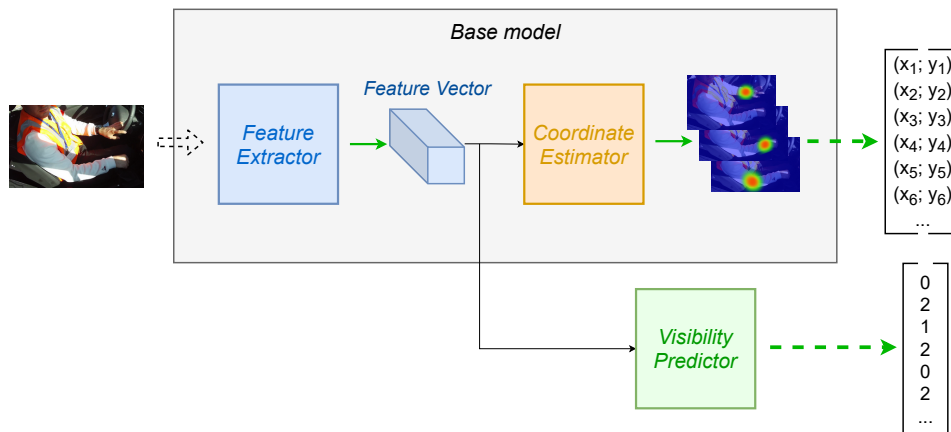


Figure 2: Architecture of our multitask metamodel for keypoint and visibility estimations.

136 However, prior works only predict binary visibility 168
 137 ity and do not take advantage of the three visibility 169
 138 labels provided by the current datasets (visible, non- 170
 139 visible, non-labeled). Furthermore, the authors provide 171
 140 few quantified results on the actual performance 172
 141 of the visibility predictions. Finally, these works propose 173
 142 a fixed network where the visibility prediction part 174
 143 is mostly ancillary. In this context, we propose a 175
 144 metamodel that allows HPE methods to predict both 176
 145 keypoints coordinates and ternary visibility. 177

are computed as the local maximum of each heatmap. 168
 The majority of the HPE networks can be split into a 169
 feature extraction and a heatmap generation modules, 170
 which allows most of the architectures to be compatible 171
 with our metamodel. 172

In addition to these two regular modules, we add a 173
 visibility branch (Figure 3). This module takes as input 174
 the same feature vector as the coordinate estimation 175
 module and outputs the visibility prediction for each 176
 keypoint. The detailed architecture is presented in 177
 the next section. 178

146 3 PROPOSED METAMODEL 179

3.2 Visibility branch

147 This section presents the architecture of the proposed 180
 148 HPE visibility metamodel. First, we describe the 181
 149 overall architecture. Then, we provide a more detailed 182
 150 description of our visibility module. 183

151 3.1 Metamodel 187

152 The proposed architecture is split into three parts: the 188
 153 feature extraction, the coordinate estimation, and the 189
 154 visibility prediction modules. First, the feature ex- 190
 155 traction module processes the input image to gener- 191
 156 ate a feature vector. Examples of feature extractor 192
 157 are encoder architectures (Newell et al., 2016; 193
 158 Tang and Wu, 2019; Artacho and Savakis, 2020; Li 194
 159 et al., 2019), or image recognition backbones such 195
 160 as ResNet (He et al., 2016) or EfficientNet (Tan and 196
 161 Le, 2019). Then, the generated vector serves as the 197
 162 input of the two other modules. Coordinate estimation 198
 163 can be performed by modules such as decoder or 199
 164 deconvolution stages, usually followed by a convolu- 200
 165 tion layer which generates the final heatmaps (Newell 201
 166 et al., 2016; Tang and Wu, 2019; Artacho and Savakis, 202
 167 2020; Li et al., 2019). Final coordinate predictions 203

We model the visibility prediction problem as a classifi- 180
 cation task. We follow the COCO dataset formalism 181
 and define the visibility using integer labels: 0 when 182
 the keypoint is not labeled, 1 when it is labeled but 183
 not visible, and 2 when it is fully visible. Therefore, 184
 we associate to each keypoint one of the three labels. 185
 The visibility module takes as input the feature vector 186
 computed by the feature extraction module. It is com- 187
 posed of a convolutional module, followed by a fully 188
 connected network (FCN) that generates the final vis- 189
 ibility predictions. 190

More precisely, a residual block (He et al., 2016) 191
 first processes the input features. This block is com- 192
 posed of three successive convolution layers with re- 193
 spective kernel sizes of 3x3, 1x1, and 3x3, which 194
 form a bottleneck. An additional skip connection en- 195
 ables the features to be directly propagated to the next 196
 layer. We use this block in our branch since it has 197
 shown good results in feature computation for HPE 198
 ((Newell et al., 2016; Tang and Wu, 2019)). Then, 199
 a convolutional layer of kernel size 1x1 with Batch- 200
 Norm and 2x2 max pooling reduces the size and the 201
 number of channels of the features. Finally, features 202
 are flattened and a fully connected network with three 203

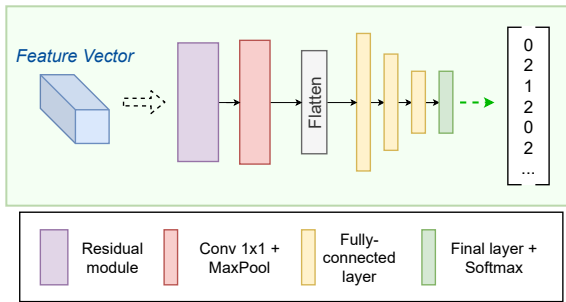


Figure 3: Architecture of our visibility predictor module.

hidden layers (4096, 2048, and 1024 neurons) followed by a SoftMax produces the predictions. Since the COCO dataset provides 17 annotated keypoints with three possible visibility classes, the output layer is composed of 51 neurons. The SoftMax function is applied to groups of three visibility neurons (one group representing one keypoint).

3.3 Cost function

The global cost function used to train the network is defined as follows:

$$L = (1 - \alpha) \cdot L_H + \alpha \cdot L_V \quad (1)$$

where L_H is an L2 distance between the predicted heatmaps and the ground-truth. The ground-truth heatmaps are generated using Gaussian centered around the location of the keypoint, with a standard deviation of 1px.

The function L_V is the cross-entropy loss applied to the predictions of the visibility classes. Weighted cross-entropy is used to compensate for the imbalanced distribution of keypoints within the three visibility classes. Therefore, the weights are computed as the size of the biggest class divided by the size of each class. Finally, α is the parameter used to balance the ratio between the loss functions associated with the two tasks. This regulates the impact of each tasks on the training of the feature extractor weights.

4 EXPERIMENTS

In this section, we provide details about how the experiments have been carried out, such as used datasets, training, network base models, and evaluation procedure.

4.1 Datasets

We adopted two datasets for the experiments. First, the COCO dataset (Lin et al., 2015), which is one

of the largest and most used datasets for 2D human pose estimation in a general context. It is composed of 118k pictures for training and 5k for validation. However, because of the high number of pictures in this dataset, the visibility annotations present some inconsistencies. Also, the non-visible keypoints are weakly represented in the COCO dataset, with only 7% of the total keypoints. Therefore, we evaluated our architecture on a second dataset called DriPE (Guesdon et al., 2021). Figure 4 illustrates some samples. This dataset possesses 10k manually annotated images of drivers in consumer vehicles (7.4k images for training, 1.3k images each for training and testing). The car environment and the side view-angle of the cameras produce strong occlusion which induces 19% of non-visible keypoints.

4.2 Basic Training

Most of the results on our architecture are provided using the Simple Baseline (SBI) network as the base model (Xiao et al., 2018). This network combines ResNet50 as feature extractor with a deconvolution stage (as coordinate estimator) to generate the final heatmaps. The feature extractor is initialized with weights pre-trained on ImageNet. The networks are trained on the COCO dataset for 140 epochs with a learning rate of 1E-3, decreased by a factor of 10 at epochs 90 and 120.

Finetuning on DriPE is done during 10 epochs with a learning rate of 1E-4. We use data augmentation operations (rotation, flipping, etc.) for both datasets. Following the state of the art, the input images are cropped around the subjects using the ground-truth, for both training and evaluation. Training is performed on a computer with an Nvidia GTX 1080 graphic card, an Intel Core i990k processor, and 32 GB of RAM.



Figure 4: Image samples from DriPE dataset. Faces on the figure have been blurred for anonymity purpose.

4.3 Multitask Training

We tested in our experiments three strategies for multitask training. As detailed in the previous section, weights of the feature extractor are initialized on ImageNet and the visibility predictor’s weights are initialized randomly. For the first strategy (S1), we train the keypoint estimation and the visibility prediction tasks jointly with a fixed α set to 0.25 (value chosen empirically). For the second and third strategies (S2 and S3), we pre-train the feature extraction and coordinate prediction modules on COCO dataset, in the same way as regular HPE networks are trained. Then, we resume the training for 80 epochs, while incrementing α by 0.1 every 20 epochs, starting from $\alpha=0$. In S2, the whole model is updated during these 80 epochs. However, in S3, only the visibility predictor is trained during this step, while the remaining weights (feature extractor and coordinate estimator) are frozen.

4.4 Base models

We implemented for the experiments three base models with our method, besides Simple Baseline. We first used EfficientNet as a feature extractor (Tan and Le, 2019), which is more recent than ResNet. We employed two different sizes: B0 (the smallest) and B6 (the second largest). We followed the same training strategy and reused the heatmap generator from the Simple Baseline model.

We also set up our metamodel with the MSPN network (Li et al., 2019), as a feature extractor and a heatmap generator. Because MSPN uses a multi-stage architecture, we extracted the feature vector from the output of the last encoder to feed the visibility module. We initialized the model with the weights already trained on COCO for human pose estimation.

4.5 Evaluation

The performance of the coordinate prediction module was measured using two metrics. First, we used the regular metric for the COCO dataset called AP OKS (Lin et al., 2015). This metric computes the average precision and recall using a score called OKS. However, this metric is person-centered and does not provide information on the model performance of each keypoint detection. Furthermore, this metric only considers labeled keypoints, *i.e.*, visible and non-visible keypoints, which puts aside false-positive predictions. Therefore, we also evaluated the models with the mAPK metric (Guesdon et al., 2021). This metric provides an evaluation at a keypoint level

and allows to measure the performance of the model on each body part separately.

5 RESULTS

In this section, we present and discuss the performance of the proposed metamodel. More precisely, we first study the quality of the visibility predictions using different strategies to train the models. Then, we study the impact of the visibility prediction on the keypoint detection task using both AP OKS and mAPK metrics. Finally, we discuss the performance of the proposed solution with different base models.

5.1 Visibility prediction

We tried out several strategies to train the model, described in Section 4.3. The performance of the three resulting networks is presented in Table 1.

Table 1: F1-score of the network for visibility prediction on COCO 2017 val set with different training strategies.

| Strategy | non-labeled | non-visible | visible | total |
|----------|-------------|-------------|-------------|-------------|
| S1 | 0.72 | 0.21 | 0.76 | 0.71 |
| S2 | 0.75 | 0.34 | 0.79 | 0.74 |
| S3 | 0.77 | 0.37 | 0.80 | 0.76 |

First, we can observe in Table 1 that pre-training the network on the keypoint estimation task (S2 and S3) outperforms the joint training of the three modules (S1). Indeed, we can notice an increase of 5% of the total F1-score between S1 and S3. This improvement is mostly perceptible in the non-visible class (gain of 16%). However, training on the visibility task while freezing the rest of the network (S3) does not impact the overall performance. Indeed, we trained several models and present in Table 1 the model for each strategy with the best performance. Nevertheless, we observed little performance differences between the networks trained with and without freezing. In the end, this experiment demonstrates that already trained HPE networks can be used with our metamodel and reach optimal performance. This allows saving time and computing power, especially with a large dataset like COCO.

Regarding the performance of visibility prediction, results in Table 1 show that we are able to predict keypoint visibility with a total F1-score up to 76%. However, we can notice that the model has difficulties to predict the "non-visible" class, with a maximum F1-score of 37%. Two reasons can explain this gap. First, non-visible keypoint is a subjective notion, since it corresponds to the keypoints which are oc-

360 cluded but where we have enough information in the 382
 361 image to deduce the location of the keypoint. Because
 362 the assessment of the "enough information" is left to
 363 the annotator, it leads to inconsistency in the annota-
 364 tions. Secondly, the keypoints labeled as non-visible
 365 represent only 7% of the COCO keypoints (Figure 5).
 366 Even if this distribution gap is taken into considera-
 367 tion in the computation of the weighted cross-entropy
 368 cost function L_v , it still has a negative impact on the
 369 learning process.

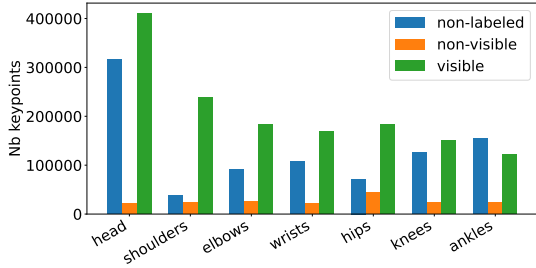


Figure 5: Distribution of the keypoint visibility labels in the COCO dataset.

370 To study the impact of the distribution of examples
 371 of the three visibility classes, we finetuned our net-
 372 work on DriPE dataset (Guesdon et al., 2021). This
 373 dataset presents a more homogeneous keypoints class
 374 distribution, as shown in Figure 6.

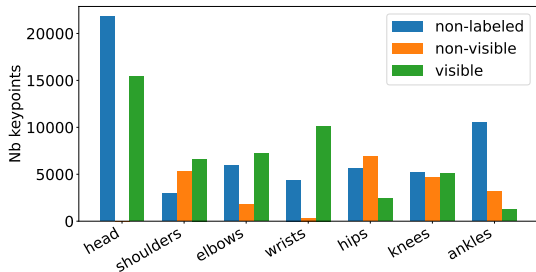


Figure 6: Distribution of the keypoint visibility labels in the DriPE dataset.

Table 2: Performance of the network for visibility prediction on DriPE dataset before and after finetuning.

| F1-score | non-labeled | non-visible | visible | total |
|--------------------|-------------|-------------|-------------|-------------|
| COCO baseline | 0.71 | 0.34 | 0.64 | 0.60 |
| Finetuned on DriPE | 0.81 | 0.70 | 0.76 | 0.76 |

375 As we can see in Table 2, after finetuning, the
 376 model achieves an F1-score of 70% for the non-
 377 visible keypoints. These results demonstrate that with
 378 a better distribution of the visibility classes and more
 379 homogeneous images, our metamodel is able to bet-
 380 ter estimate the visibility of keypoints, in particular
 381 for non-visible classes.

5.2 Keypoint estimation

383 We now study the impact of the addition of the
 384 visibility module on the performance of the key-
 385 point detection. We use for this study the mAPK
 386 metric (Guesdon et al., 2021), which provides a
 387 more keypoint-centered performance measurement
 388 than AP OKS (Lin et al., 2015). Similar to AP OKS,
 389 mAPK measures both average precision (AP) and av-
 390 erage recall (AR). We provide results for both COCO
 391 (Table 3) and DriPE (Table 4) datasets. The "SBI +
 392 visibility" network refers to the implementation of our
 393 metamodel with the Simple Baseline network. The
 394 "non-0" term defines the experiment where all key-
 395 point coordinates predicted by the visibility module as
 396 "non-labeled" are considered as not predicted for the
 397 computation of the mAPK metric. This strategy aims
 398 to improve the precision on scenes where some key-
 399 points are outside the image or strongly occluded of
 400 the keypoint prediction module, which is classically
 401 designed to predict coordinates for each type of key-
 402 point during inference.

Table 3: HPE on the COCO 2017 validation set with mAPK.

| | configuration | Head | Sho. | Elb. | Wri. | Hip | Knee | Ank. | Mean |
|----|--------------------------|------|------|------|------|------|------|------|------|
| AP | SBI | 0.66 | 0.76 | 0.73 | 0.70 | 0.74 | 0.74 | 0.74 | 0.72 |
| | SBI + visibility | 0.66 | 0.76 | 0.72 | 0.70 | 0.73 | 0.73 | 0.73 | 0.72 |
| | SBI + visibility + non-0 | 0.71 | 0.78 | 0.77 | 0.73 | 0.73 | 0.76 | 0.74 | 0.75 |
| AR | SBI | 0.73 | 0.77 | 0.73 | 0.70 | 0.70 | 0.72 | 0.72 | 0.72 |
| | SBI + visibility | 0.73 | 0.76 | 0.73 | 0.69 | 0.70 | 0.72 | 0.72 | 0.72 |
| | SBI + visibility + non-0 | 0.43 | 0.72 | 0.58 | 0.68 | 0.68 | 0.66 | 0.35 | 0.59 |

Table 4: HPE on the DriPE test set with mAPK.

| | configuration | Head | Sho. | Elb. | Wri. | Hip | Knee | Ank. | Mean |
|----|--------------------------|------|------|------|------|------|------|------|------|
| AP | SBI | 0.85 | 0.90 | 0.94 | 0.96 | 0.98 | 0.95 | 0.68 | 0.89 |
| | SBI + visibility | 0.84 | 0.90 | 0.94 | 0.96 | 0.98 | 0.95 | 0.68 | 0.89 |
| | SBI + visibility + non-0 | 0.86 | 0.90 | 0.94 | 0.97 | 0.98 | 0.96 | 0.72 | 0.90 |
| AR | SBI | 0.87 | 0.96 | 0.96 | 0.97 | 0.98 | 0.95 | 0.80 | 0.93 |
| | SBI + visibility | 0.87 | 0.96 | 0.96 | 0.97 | 0.98 | 0.95 | 0.80 | 0.93 |
| | SBI + visibility + non-0 | 0.44 | 0.96 | 0.85 | 0.97 | 0.98 | 0.93 | 0.77 | 0.84 |

403 Firstly, we can observe that our metamodel
 404 (SBI + visibility) achieves performance similar to the
 405 SBI baseline on keypoint detection. It indicates that
 406 adding the visibility task has no negative impact on
 407 the primary task, regardless of the dataset used.

408 Secondly, the non-0 strategy slightly improves the
 409 average precision of the keypoint detection, which
 410 denotes a decrease in the number of false positives.
 411 However, this precision increase comes with a nega-
 412 tive trade-off regarding the average recall, caused by
 413 an increase of the false negatives. The decrease of the
 414 recall is significant for the keypoints on the head, el-
 415 bow, and ankles. Prediction of the visibility on the
 416 face is a delicate task since almost none of these key-
 417 points are labeled as non-visible due to the COCO an-
 418 notation style. Ankles are also difficult keypoints to

Table 5: Performance of the network for keypoint detection on COCO 2017 with different base models.

| Base model | parameters | AP | AP ⁵⁰ | AP ⁷⁵ | AP ^M | AP ^L | AR | AR ⁵⁰ | AR ⁷⁵ | AR ^M | AR ^L |
|-----------------|------------|------|------------------|------------------|-----------------|-----------------|------|------------------|------------------|-----------------|-----------------|
| SBI | 71.2M | 71.9 | 91.5 | 79.0 | 69.2 | 76.4 | 75.3 | 92.8 | 81.8 | 72.1 | 80.1 |
| EfficientNet B0 | 55.6M | 67.1 | 90.4 | 74.9 | 63.9 | 71.7 | 70.3 | 91.1 | 77.0 | 66.8 | 75.5 |
| EfficientNet B6 | 95.5M | 72.5 | 92.4 | 80.1 | 69.8 | 76.9 | 75.8 | 93.0 | 82.7 | 72.6 | 80.7 |
| MSPN 2-stg | 104.6M | 71.8 | 92.5 | 81.4 | 69.0 | 76.1 | 75.3 | 93.5 | 83.8 | 71.9 | 80.3 |

Table 6: Performance of the network for keypoint detection on DriPE with different base models.

| Base model | parameters | AP | AP ⁵⁰ | AP ⁷⁵ | AP ^M | AP ^L | AR | AR ⁵⁰ | AR ⁷⁵ | AR ^M | AR ^L |
|-----------------|------------|------|------------------|------------------|-----------------|-----------------|------|------------------|------------------|-----------------|-----------------|
| SBI | 71.2M | 96.5 | 99.9 | 99.9 | - | 96.5 | 97.5 | 99.9 | 99.9 | - | 97.5 |
| EfficientNet B0 | 55.6M | 91.8 | 99.0 | 99.0 | - | 91.8 | 94.7 | 99.9 | 99.6 | - | 94.7 |
| EfficientNet B6 | 95.5M | 99.4 | 99.0 | 99.0 | - | 94.4 | 96.5 | 99.9 | 99.6 | - | 96.5 |
| MSPN 2-stg | 104.6M | 97.8 | 99.0 | 99.0 | - | 97.8 | 99.0 | 99.9 | 99.9 | - | 99.0 |

419 predict in a general context, even if it is less observ- 450
420 able in the DriPE dataset due to the lower number of 451
421 labeled ankles. In the end, an increase of precision 452
422 can be useful in applications that require high confi-
423 dence in the predicted keypoints.

Table 7: Performance of the network for visibility prediction on COCO 2017 with different base models.

| Base model | parameters | non-labeled | non-visible | visible | total |
|-----------------|------------|-------------|-------------|---------|-------|
| SBI | 71.2M | 0.77 | 0.37 | 0.80 | 0.76 |
| EfficientNet B0 | 55.6M | 0.74 | 0.32 | 0.77 | 0.73 |
| EfficientNet B6 | 95.5M | 0.75 | 0.34 | 0.80 | 0.76 |
| MSPN 2-stg | 104.6M | 0.69 | 0.34 | 0.69 | 0.67 |

424 We present qualitative results in Figure 7. As 455
425 we observed in Tables 3 and 4, the gain in precision 456
426 comes mostly from face keypoints. This is illustrated 457
427 by face keypoints which were predicted even with 458
428 the strong occlusion and the lack of information (Fig- 459
429 ure 7-A,B). However, the precision of other parts pre- 460
430 diction has also been improved, such as knees (Fig- 461
431 ure 7-C). Finally, the negative trade-off regarding the 462
432 recall is caused by keypoints that were correctly pre- 463
433 dicted by the coordinate estimator but predicted as 464
434 non-labeled by the visibility predictor (Figure 7-D). 465

435 5.3 Other base models

436 We evaluated our metamodel with different HPE archi- 466
437 tectures: EfficientNet B0 and B6, and MSPN. The 467
438 performance of these implementations can be found 468
439 in Tables 5 and 7. The two tasks were trained succes- 469
440 sively while freezing the feature extractor during the 470
441 visibility task training. 471

442 As we can observe, the models achieve good perfor- 472
443 mance on pose estimation while reaching perfor- 473
444 mance on visibility prediction similar to the one pre- 474
445 sented in Table 1. These results intend to demon- 475
446 strate that our metamodel can be deployed with net- 476
447 works of varied sizes and architectures while preserv- 477
448 ing the performance on both tasks. Please note that we 478
449 trained each network only once except SBI which is 479

used as the baseline for our study. Therefore, these re-
sults may not reflect the optimal performance of each
network.

Table 8: Performance of the network for visibility prediction on DriPE with different base models.

| Base model | parameters | non-labeled | non-visible | visible | total |
|-----------------|------------|-------------|-------------|---------|-------|
| SBI | 71.2M | 0.81 | 0.70 | 0.76 | 0.76 |
| EfficientNet B0 | 55.6M | 0.72 | 0.54 | 0.72 | 0.69 |
| EfficientNet B6 | 95.5M | 0.78 | 0.58 | 0.63 | 0.67 |
| MSPN 2-stg | 104.6M | 0.57 | 0.55 | 0.46 | 0.51 |

Finally, we finetuned and evaluated the networks
on DriPE dataset (Tables 6 and 8). The models still
achieve 60% of visibility prediction while reaching
over 90% of precision and recall on the keypoint es-
timation. We can notice that the performance of the
MSPN network is below what we could expect for
such a large number of parameters. An adjustment
of the training and finetuning parameters could im-
prove performance, especially considering the size of
the network. Also, because of the multiscale and mul-
tistage architecture of MSPN, concatenating several
scale levels to extract the feature vector from the net-
work could improve the results.

466 6 CONCLUSIONS

In this paper, we have presented a new metamodel for
human pose estimation and visibility prediction. This
method achieves good performance on visibility pre-
diction while preserving the performance of the key-
point estimation of the base model. We demonstrated
that these results can be achieved using different base
models. We also showed that the metamodel performs
well on two public datasets regarding the visibility
prediction: the COCO dataset, a general and state-of-
the-art dataset, and the DriPE dataset which contains
images with stronger occlusion. Finally, we used the
predicted visibility to improve the keypoint detection,
by discarding the keypoints predicted as non-labeled.

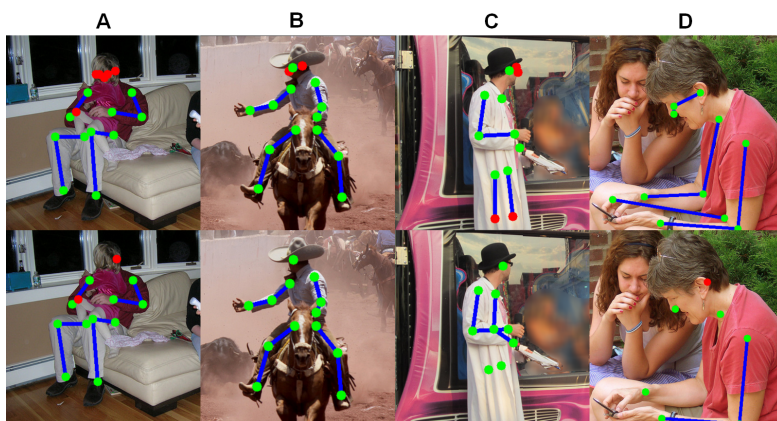


Figure 7: Qualitative comparison of keypoints prediction filtered with a confidence threshold (top row) and with the visibility predicted by our metamodel (bottom row). Red dots represent the false-positive keypoints.

480 Our results show that this strategy can improve the 511
 481 precision of the detection, even though it may reduce 512
 482 the recall, especially for head and ankles keypoints. 513

483 Future work will investigate strategies to improve 514
 484 the precision of keypoint coordinates estimation using 515
 485 visibility prediction with a lesser negative trade-off on 516
 486 recall. For instance, we could combine the predicted 517
 487 confidence of the two tasks for a final prediction. Fur- 518
 488 thermore, it would be interesting to study the integra- 519
 489 tion of the proposed metamodel to multi-scale archi- 520
 490 tectures, like MSPN architecture. These architectures 521
 491 present a higher performance on keypoint estimation, 522
 492 but the proposed integration still does not take full ad- 523
 493 vantage of the multiscale features available. Finally, it 524
 494 would be interesting to study the influence of the gain 525
 495 of keypoint estimation accuracy in practical applica- 526
 496 tions, such as action recognition or posture analysis in 527
 497 car-safety applications. 528
 529

498 Acknowledgements

499 This work was supported by the Pack Ambition 536
 500 Recherche 2019 funding of the French AURA Region 537
 501 in the context of the AutoBehave project. 538
 539

502 REFERENCES

503 Andriluka, M., Pishchulin, L., Gehler, P., and Schiele, B. 545
 504 (2014). 2d human pose estimation: New benchmark 546
 505 and state of the art analysis. In *2014 IEEE Conference* 547
 506 *on Computer Vision and Pattern Recognition*, pages 548
 507 3686–3693. 549
 508 Artacho, B. and Savakis, A. (2020). Unipose: Unified human 550
 509 pose estimation in single images and videos. In 551
 510 *Proceedings of the IEEE/CVF Conference on Com-* 552

puter Vision and Pattern Recognition (CVPR), pages 7035–7044.
 514
 515 Cai, Y., Wang, Z., Luo, Z., Yin, B., Du, A., Wang, H., Zhang, X., Zhou, X., Zhou, E., and Sun, J. (2020). Learning delicate local representations for multi-person pose estimation. In Vedaldi, A., Bischof, H., Brox, T., and Frahm, J.-M., editors, *Computer Vision – ECCV 2020*, pages 455–472, Cham. Springer International Publishing.
 516
 517 Cao, Z., Hidalgo, G., Simon, T., Wei, S.-E., and Sheikh, Y. (2019). Openpose: realtime multi-person 2d pose estimation using part affinity fields. *IEEE transactions on pattern analysis and machine intelligence*, 43(1):172–186.
 518
 519 Cao, Z., Simon, T., Wei, S.-E., and Sheikh, Y. (2017). Real-time multi-person 2d pose estimation using part affinity fields. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1302–1310.
 520
 521 Carreira, J., Agrawal, P., Fragkiadaki, K., and Malik, J. (2016). Human pose estimation with iterative error feedback. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4733–4742.
 522
 523 Chen, Y., Shen, C., Wei, X.-S., Liu, L., and Yang, J. (2017). Adversarial posenet: A structure-aware convolutional network for human pose estimation. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 1221–1230.
 524
 525 Chu, X., Yang, W., Ouyang, W., Ma, C., Yuille, A. L., and Wang, X. (2017). Multi-context attention for human pose estimation. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5669–5678.
 526
 527 Das, S., Koperski, M., Bremond, F., and Francesca, G. (2017). Action recognition based on a mixture of rgb and depth based skeleton. In *2017 14th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, pages 1–6.
 528
 529 Guesdon, R., Crispim-Junior, C., and Tougne, L. (2021). Dripe: A dataset for human pose estimation in real-world driving settings. In *Proceedings of the*

- 553 *IEEE/CVF International Conference on Computer Vi-* 613
554 *sion (ICCV) Workshops*, pages 2865–2874. 614
- 555 He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep resid- 615
556 ual learning for image recognition. In *Proceedings of* 616
557 *the IEEE Conference on Computer Vision and Pattern* 617
558 *Recognition (CVPR)*. 618
- 559 Hidalgo, G., Raaj, Y., Idrees, H., Xiang, D., Joo, H., Simon, 619
560 T., and Sheikh, Y. (2019). Single-network whole-body 620
561 pose estimation. In *Proceedings of the IEEE/CVF* 621
562 *International Conference on Computer Vision*, pages 622
563 6982–6991. 623
- 564 Ke, L., Chang, M.-C., Qi, H., and Lyu, S. (2018). Multi- 624
565 scale structure-aware network for human pose esti- 625
566 mation. In Ferrari, V., Hebert, M., Sminchisescu, 626
567 C., and Weiss, Y., editors, *Computer Vision – ECCV* 627
568 *2018*, pages 731–746, Cham. Springer International 628
569 Publishing. 629
- 570 Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). 630
571 Imagenet classification with deep convolutional neu- 631
572 ral networks. In Pereira, F., Burges, C. J. C., Bottou, 632
573 L., and Weinberger, K. Q., editors, *Advances in Neu-* 633
574 *ral Information Processing Systems 25*, pages 1097– 634
575 1105. Curran Associates, Inc. 635
- 576 Kumar, A., Alavi, A., and Chellappa, R. (2017). Kepler: 636
577 Keypoint and pose estimation of unconstrained faces 637
578 by learning efficient h-cnn regressors. In *2017 12th* 638
579 *IEEE International Conference on Automatic Face* 639
580 *Gesture Recognition (FG 2017)*, pages 258–265. 640
- 581 Kumar, A., Marks, T. K., Mou, W., Wang, Y., Jones, M., 641
582 Cherian, A., Koike-Akino, T., Liu, X., and Feng, C. 642
583 (2020). Luvli face alignment: Estimating landmarks’ 643
584 location, uncertainty, and visibility likelihood. In *Pro-* 644
585 *ceedings of the IEEE/CVF Conference on Computer* 645
586 *Vision and Pattern Recognition*, pages 8236–8246. 646
- 587 Li, W., Wang, Z., Yin, B., Peng, Q., Du, Y., Xiao, T., Yu, 647
588 G., Lu, H., Wei, J., and Sun, J. (2019). Rethinking on 648
589 multi-stage networks for human pose estimation. 649
- 590 Lin, T.-Y., Dollár, P., Girshick, R., He, K., Hariharan, B., 650
591 and Belongie, S. (2017). Feature pyramid networks 651
592 for object detection. In *2017 IEEE Conference on* 652
593 *Computer Vision and Pattern Recognition (CVPR)*, 653
594 pages 936–944. 654
- 595 Lin, T.-Y., Maire, M., Belongie, S., Bourdev, L., Girshick, 655
596 R., Hays, J., Perona, P., Ramanan, D., Zitnick, C. L., 656
597 and Dollár, P. (2015). Microsoft coco: Common ob- 657
598 jects in context. 658
- 599 Newell, A., Huang, Z., and Deng, J. (2017). Associative 659
600 embedding: End-to-end learning for joint detection 660
601 and grouping. In Guyon, I., Luxburg, U. V., Ben- 661
602 gio, S., Wallach, H., Fergus, R., Vishwanathan, S., and 662
603 Garnett, R., editors, *Advances in Neural Information* 663
604 *Processing Systems 30*, pages 2277–2287. Curran As- 664
605 sociates, Inc.
- 606 Newell, A., Yang, K., and Deng, J. (2016). Stacked Hour- 607
608 glass Networks for Human Pose Estimation. In Leibe, 608
609 B., Matas, J., Sebe, N., and Welling, M., editors, *Com-* 609
610 *puter Vision – ECCV 2016*, pages 483–499, Cham. 610
Springer International Publishing.
- 611 Nie, X., Feng, J., Xing, J., and Yan, S. (2018). Pose partition 611
612 networks for multi-person pose estimation. In *Com-* 612
puter Vision – ECCV 2018, pages 684–699, Cham. 613
Springer International Publishing.
- Stoffl, L., Vidal, M., and Mathis, A. (2021). End-to-end 614
trainable multi-instance pose estimation with trans- 615
formers. 616
- Sun, K., Xiao, B., Liu, D., and Wang, J. (2019). Deep high- 617
resolution representation learning for human pose es- 618
timation. In *2019 IEEE/CVF Conference on Com-* 619
puter Vision and Pattern Recognition (CVPR), pages 620
5686–5696. 621
- Sun, X., Shang, J., Liang, S., and Wei, Y. (2017). Composi- 622
tional human pose regression. In *2017 IEEE Interna-* 623
tional Conference on Computer Vision (ICCV), pages 624
2621–2630. 625
- Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., 626
Anguelov, D., Erhan, D., Vanhoucke, V., and Rabi- 627
novich, A. (2015). Going deeper with convolutions. 628
In *Proceedings of the IEEE Conference on Computer* 629
Vision and Pattern Recognition (CVPR). 630
- Tan, M. and Le, Q. (2019). Efficientnet: Rethinking model 631
scaling for convolutional neural networks. In *Internat-* 632
ional Conference on Machine Learning, pages 6105– 633
6114. PMLR. 634
- Tang, W. and Wu, Y. (2019). Does learning specific fea- 635
tures for related parts help human pose estimation? In 636
2019 IEEE/CVF Conference on Computer Vision and 637
Pattern Recognition (CVPR), pages 1107–1116. 638
- Tang, W., Yu, P., and Wu, Y. (2018). Deeply learned compo- 639
sitional models for human pose estimation. In Ferrari, 640
V., Hebert, M., Sminchisescu, C., and Weiss, Y., edi- 641
tors, *Computer Vision – ECCV 2018*, pages 197–214, 642
Cham. Springer International Publishing. 643
- Toshev, A. and Szegedy, C. (2014). Deeppose: Human pose 644
estimation via deep neural networks. In *2014 IEEE* 645
Conference on Computer Vision and Pattern Recogni- 646
tion, pages 1653–1660. 647
- Xiao, B., Wu, H., and Wei, Y. (2018). Simple baselines 648
for human pose estimation and tracking. In Ferrari, 649
V., Hebert, M., Sminchisescu, C., and Weiss, Y., edi- 650
tors, *Computer Vision – ECCV 2018*, pages 472–487, 651
Cham. Springer International Publishing. 652
- Zhao, L., Xu, J., Zhang, S., Gong, C., Yang, J., and Gao, 653
X. (2020a). Perceiving heavily occluded human poses 654
by assigning unbiased score. *Information Sciences*, 655
537:284–301. 656
- Zhao, M., Beurier, G., Wang, H., and Wang, X. (2020b). 657
A pipeline for creating in-vehicle posture database 658
for developing driver posture monitoring systems. 659
In *DHM2020: Proceedings of the 6th International* 660
Digital Human Modeling Symposium, August 31- 661
September 2, 2020, volume 11, pages 187–196. IOS 662
Press. 663