# Multitask Metamodel for Keypoint Visibility Prediction in Human Pose Estimation

Romain Guesdon, Carlos F Crispim-Junior, Laure Tougne

# 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 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.

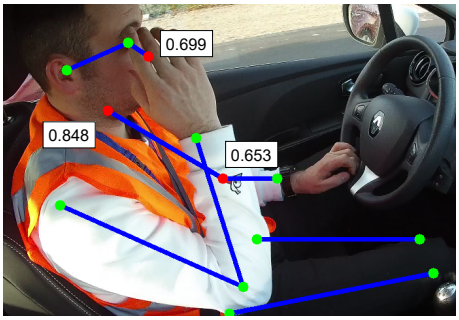


Figure 1: HPE prediction. Red points represent false positives, *i.e.,* keypoints that were predicted even if not annotated due to strong occlusion. Confidence scores are provided in the boxes (maximum score = 1.0).

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

point. A labeled keypoint can be visible, or non-visible when the keypoint is lightly occluded but with enough information to be located. If the keypoint is heavily occluded or out of the field of view, it is not labeled. However, state-of-the-art networks do not consider these visibility labels. Furthermore, the few existing methods using visibility only consider binary visibility, *i.e.,* labeled or non-labeled keypoints (Stoffl et al., 2021; Kumar et al., 2020).

This paper proposes a novel HPE metamodel [1] that can predict both the visibility and the coordinates of the keypoints. Our solution can be implemented with most of the deep-learning HPE methods and allows these base models to predict keypoint visibility. The model can predict the three classes of labels, which provides a finer description of the keypoint visibility.

This paper is organized as follows. We present in Section 2 the related work on human pose estimation and visibility prediction. Section 3 presents our metamodel and its detailed architecture, especially the visibility module. We describe in Section 4 the details about the experiments, and present the results in Section 5. Finally, we discuss in Section 6 our conclusions and future work.

## 2 RELATED WORK

This section presents existing work on human pose estimation and visibility keypoints prediction.

The task of human pose estimation is divided into two categories. Single-person HPE focuses on the detection in pictures presenting one person, in opposition to multiperson detection. The first approach to solve single-person HPE using deep learning was proposed in (Toshev and Szegedy, 2014). This solution is based on the deep architecture AlexNet (Krizhevsky et al., 2012), which is used to estimate and refine the coordinates. An Iterative Error Feedback network was proposed in (Carreira et al., 2016) based on the convolutional network GoogleNet (Szegedy et al., 2015). The authors of (Sun et al., 2017) used ResNet50 (He et al., 2016) to predict a parametrized bones representation. However, all these methods try to directly predict the keypoints coordinates from the images, which affects the robustness of these methods due to the high non-linearity of this approach. Other solutions categorized as detection-based methods aim to predict 2D matrices called heatmaps where each pixel represents the probability for a joint to be located here. The work of (Newell et al., 2016) proposed an hourglass module that can be stacked to predict and refine features at several scales, which has inspired many other works (Chu et al., 2017; Ke et al., 2018; Tang and Wu, 2019; Tang et al., 2018). Besides hourglass architectures, other detection-based methods have been proposed. The architecture in (Chen et al., 2017) combines a heatmap generator with two discriminators. Simple Baseline (Xiao et al., 2018), is an architecture based on the ResNet network (He et al., 2016) with a deconvolution stage to generate the final heatmaps. Finally, Unipose (Artacho and Savakis, 2020) combines atrous and cascade convolutions to produce a multi-scale representation.

In addition to finding the keypoints in the picture, multiperson HPE brings a new difficulty: to associate the different persons to the detected keypoints. State-of-the-art performance is achieved by methods called top-down approaches that first detect the subjects in the picture and then locate the keypoints for each person individually. These methods usually combine a person detector with a single-person HPE architecture (Xiao et al., 2018; Sun et al., 2019; Lin et al., 2017; Cai et al., 2020; Li et al., 2019). Conversely, the bottom-up approaches first detect every keypoints in the image before associating them to form people instances (Newell et al., 2017; Cao et al., 2017; Nie et al., 2018). Top-down approaches tend to outperform the bottom-up methods while taking advantage of both state-of-the-art person detectors and HPE architectures.

Among top-down methods, the Simple Baseline (SBl) network (Xiao et al., 2018) presents competitive performance while preserving a small size, which makes it practical for modifications and tests. In addition, it can be used for multiperson HPE by combining it with a person detector.

Recent work on human pose estimation has mainly focused on improving the prediction of the keypoints' coordinates. Therefore, methods which estimate the visibility of HPE keypoints are scarce. In (Zhao et al., 2020a), visibility prediction is used to propose a new evaluation method for multiperson pose estimation in heavily occluded contexts. Visibility is predicted as an occlusion score and is used to compute a metric that highlights the performance of the evaluated networks on occluded points. The multi-instance HPE network in (Stoffl et al., 2021) uses transformers to predict keypoint visibility, which serves as a secondary task for end-to-end training. Besides, keypoint visibility is predicted in (Kumar et al., 2017; Kumar et al., 2020) as an annex task for face detection.

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

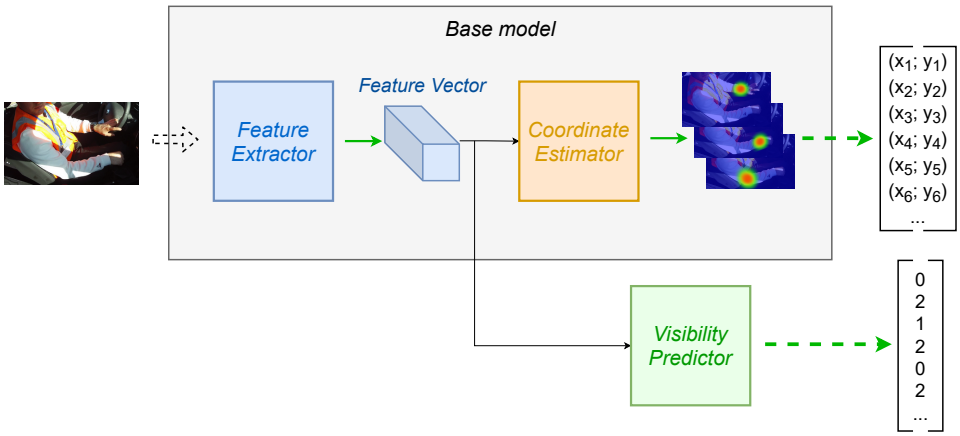


Figure 2: Architecture of our multitask metamodel for keypoint and visibility estimations.

However, prior works only predict binary visibility and do not take advantage of the three visibility labels provided by the current datasets (visible, non-visible, non-labeled). Furthermore, the authors provide few quantified results on the actual performance of the visibility predictions. Finally, these works propose a fixed network where the visibility prediction part is mostly ancillary. In this context, we propose a metamodel that allows HPE methods to predict both keypoints coordinates and ternary visibility.

# 3 PROPOSED METAMODEL

This section presents the architecture of the proposed HPE visibility metamodel. First, we describe the overall architecture. Then, we provide a more detailed description of our visibility module.

## 3.1 Metamodel

The proposed architecture is split into three parts: the feature extraction, the coordinate estimation, and the visibility prediction modules. First, the feature extraction module processes the input image to generate a feature vector. Examples of feature extractor are encoder architectures (Newell et al., 2016; Tang and Wu, 2019; Artacho and Savakis, 2020; Li et al., 2019), or image recognition backbones such as ResNet (He et al., 2016) or EfficientNet (Tan and Le, 2019). Then, the generated vector serves as the input of the two other modules. Coordinate estimation can be performed by modules such as decoder or deconvolution stages, usually followed by a convolution layer which generates the final heatmaps (Newell et al., 2016; Tang and Wu, 2019; Artacho and Savakis, 2020; Li et al., 2019). Final coordinate predictions are computed as the local maximum of each heatmap. The majority of the HPE networks can be split into a feature extraction and a heatmap generation modules, which allows most of the architectures to be compatible with our metamodel.

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

## 3.2 Visibility branch

We model the visibility prediction problem as a classification task. We follow the COCO dataset formalism and define the visibility using integer labels: 0 when the keypoint is not labeled, 1 when it is labeled but not visible, and 2 when it is fully visible. Therefore, we associate to each keypoint one of the three labels. The visibility module takes as input the feature vector computed by the feature extraction module. It is composed of a convolutional module, followed by a fully connected network (FCN) that generates the final visibility predictions.

More precisely, a residual block (He et al., 2016) first processes the input features. This block is composed of three successive convolution layers with respective kernel sizes of 3x3, 1x1, and 3x3, which form a bottleneck. An additional skip connection enables the features to be directly propagated to the next layer. We use this block in our branch since it has shown good results in feature computation for HPE ((Newell et al., 2016; Tang and Wu, 2019)). Then, a convolutional layer of kernel size 1x1 with BatchNorm and 2x2 max pooling reduces the size and the number of channels of the features. Finally, features are flattened and a fully connected network with three

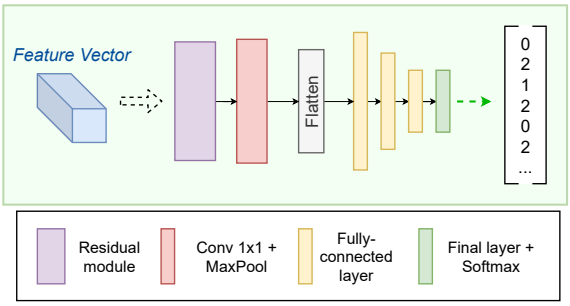


Figure 3: Architecture of our visibility predictor module.

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:

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

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, $\alpha$ 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 (SBl) 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 $\alpha$ 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 $\alpha$ 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-

cluded but where we have enough information in the image to deduce the location of the keypoint. Because the assessment of the "enough information" is left to the annotator, it leads to inconsistency in the annotations. Secondly, the keypoints labeled as non-visible represent only 7% of the COCO keypoints (Figure 5). Even if this distribution gap is taken into consideration in the computation of the weighted cross-entropy cost function $L_v$, it still has a negative impact on the learning process.

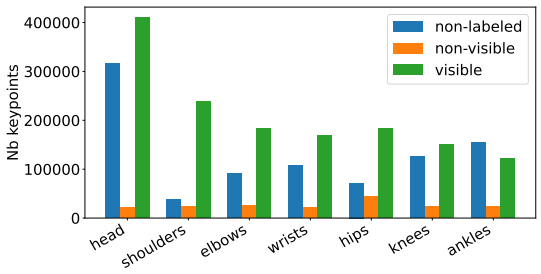


Figure 5: Distribution of the keypoint visibility labels in the COCO dataset.

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

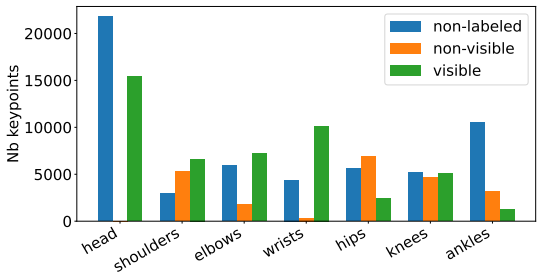


Figure 6: Distribution of the keypoint visibility labels in the DriPE dataset.

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** |

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

## 5.2 Keypoint estimation

We now study the impact of the addition of the visibility module on the performance of the keypoint detection. We use for this study the mAPK metric (Guesdon et al., 2021), which provides a more keypoint-centered performance measurement than AP OKS (Lin et al., 2015). Similar to AP OKS, mAPK measures both average precision (AP) and average recall (AR). We provide results for both COCO (Table 3) and DriPE (Table 4) datasets. The "SBl + visibility" network refers to the implementation of our metamodel with the Simple Baseline network. The "non-0" term defines the experiment where all keypoint coordinates predicted by the visibility module as "non-labeled" are considered as not predicted for the computation of the mAPK metric. This strategy aims to improve the precision on scenes where some keypoints are outside the image or strongly occluded of the keypoint prediction module, which is classically designed to predict coordinates for each type of keypoint during inference.

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

| | configuration | Head | Sho. | Elb. | Wri. | Hip | Knee | Ank. | Mean |
|---|---|---|---|---|---|---|---|---|---|
| AP | SBl | 0.66 | 0.76 | 0.73 | 0.70 | 0.74 | 0.74 | 0.74 | 0.72 |
| | SBl + visibility | 0.66 | 0.76 | 0.72 | 0.70 | 0.73 | 0.73 | 0.73 | 0.72 |
| | SBl + visibility + non-0 | 0.71 | 0.78 | 0.77 | 0.73 | 0.73 | 0.76 | 0.74 | 0.75 |
| AR | SBl | 0.73 | 0.77 | 0.73 | 0.70 | 0.70 | 0.72 | 0.72 | 0.72 |
| | SBl + visibility | 0.73 | 0.76 | 0.73 | 0.69 | 0.70 | 0.72 | 0.72 | 0.72 |
| | SBl + 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 | SBl | 0.85 | 0.90 | 0.94 | 0.96 | 0.98 | 0.95 | 0.68 | 0.89 |
| | SBl + visibility | 0.84 | 0.90 | 0.94 | 0.96 | 0.98 | 0.95 | 0.68 | 0.89 |
| | SBl + visibility + non-0 | 0.86 | 0.90 | 0.94 | 0.97 | 0.98 | 0.96 | 0.72 | 0.90 |
| AR | SBl | 0.87 | 0.96 | 0.96 | 0.97 | 0.98 | 0.95 | 0.80 | 0.93 |
| | SBl + visibility | 0.87 | 0.96 | 0.96 | 0.97 | 0.98 | 0.95 | 0.80 | 0.93 |
| | SBl + visibility + non-0 | 0.44 | 0.96 | 0.85 | 0.97 | 0.98 | 0.93 | 0.77 | 0.84 |

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

Secondly, the non-0 strategy slightly improves the average precision of the keypoint detection, which denotes a decrease in the number of false positives. However, this precision increase comes with a negative trade-off regarding the average recall, caused by an increase of the false negatives. The decrease of the recall is significant for the keypoints on the head, elbow, and ankles. Prediction of the visibility on the face is a delicate task since almost none of these keypoints are labeled as non-visible due to the COCO annotation 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 | $AR^{50}$ | $AR^{75}$ | $AR^{M}$ | $AR^{L}$ |
|---|---|---|---|---|---|---|---|---|---|---|---|
| SBl | 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 | $AR^{50}$ | $AR^{75}$ | $AR^{M}$ | $AR^{L}$ |
|---|---|---|---|---|---|---|---|---|---|---|---|
| SBl | 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 |

predict in a general context, even if it is less observable in the DriPE dataset due to the lower number of labeled ankles. In the end, an increase of precision can be useful in applications that require high confidence 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 |
|---|---|---|---|---|---|
| SBl | 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 |

We present qualitative results in Figure 7. As we observed in Tables 3 and 4, the gain in precision comes mostly from face keypoints. This is illustrated by face keypoints which were predicted even with the strong occlusion and the lack of information (Figure 7-A,B). However, the precision of other parts prediction has also been improved, such as knees (Figure 7-C). Finally, the negative trade-off regarding the recall is caused by keypoints that were correctly predicted by the coordinate estimator but predicted as non-labeled by the visibility predictor (Figure 7-D).

## 5.3 Other base models

We evaluated our metamodel with different HPE architectures: EfficientNet B0 and B6, and MSPN. The performance of these implementations can be found in Tables 5 and 7. The two tasks were trained successively while freezing the feature extractor during the visibility task training.

As we can observe, the models achieve good performance on pose estimation while reaching performance on visibility prediction similar to the one presented in Table 1. These results intend to demonstrate that our metamodel can be deployed with networks of varied sizes and architectures while preserving the performance on both tasks. Please note that we trained each network only once except SBl which is used as the baseline for our study. Therefore, these results 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 |
|---|---|---|---|---|---|
| SBl | 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 estimation. 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 improve performance, especially considering the size of the network. Also, because of the multiscale and multistage architecture of MSPN, concatenating several scale levels to extract the feature vector from the network could improve the results.

# 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 prediction while preserving the performance of the keypoint 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.

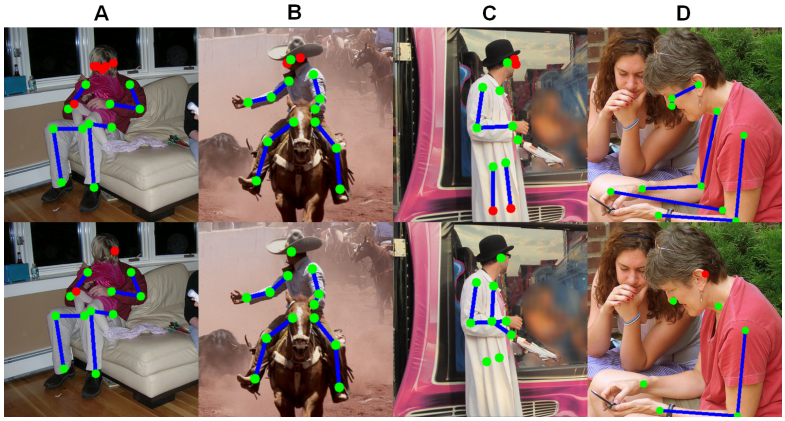


Figure 7: Qualitative comparison of keypoints prediction filtered with a confidence threshold (top row) and with the visibility predicted by our metamodel (bottom row). Red dots represent the false-positive keypoints.

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

Future work will investigate strategies to improve the precision of keypoint coordinates estimation using visibility prediction with a lesser negative trade-off on recall. For instance, we could combine the predicted confidence of the two tasks for a final prediction. Furthermore, it would be interesting to study the integration of the proposed metamodel to multi-scale architectures, like MSPN architecture. These architectures present a higher performance on keypoint estimation, but the proposed integration still does not take full advantage of the multiscale features available. Finally, it would be interesting to study the influence of the gain of keypoint estimation accuracy in practical applications, such as action recognition or posture analysis in car-safety applications.

## Acknowledgements

This work was supported by the Pack Ambition Recherche 2019 funding of the French AURA Region in the context of the AutoBehave project.

## REFERENCES

Andriluka, M., Pishchulin, L., Gehler, P., and Schiele, B. (2014). 2d human pose estimation: New benchmark and state of the art analysis. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3686–3693.

Artacho, B. and Savakis, A. (2020). Unipose: Unified human pose estimation in single images and videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7035–7044.

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.

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.

Cao, Z., Simon, T., Wei, S.-E., and Sheikh, Y. (2017). Realtime multi-person 2d pose estimation using part affinity fields. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1302–1310.

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.

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.

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.

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.

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*

*IEEE/CVF International Conference on Computer Vision (ICCV) Workshops*, pages 2865–2874.

He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Hidalgo, G., Raaj, Y., Idrees, H., Xiang, D., Joo, H., Simon, T., and Sheikh, Y. (2019). Single-network whole-body pose estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6982–6991.

Ke, L., Chang, M.-C., Qi, H., and Lyu, S. (2018). Multi-scale structure-aware network for human pose estimation. In Ferrari, V., Hebert, M., Sminchisescu, C., and Weiss, Y., editors, *Computer Vision – ECCV 2018*, pages 731–746, Cham. Springer International Publishing.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In Pereira, F., Burges, C. J. C., Bottou, L., and Weinberger, K. Q., editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc.

Kumar, A., Alavi, A., and Chellappa, R. (2017). Kepler: Keypoint and pose estimation of unconstrained faces by learning efficient h-cnn regressors. In *2017 12th IEEE International Conference on Automatic Face Gesture Recognition (FG 2017)*, pages 258–265.

Kumar, A., Marks, T. K., Mou, W., Wang, Y., Jones, M., Cherian, A., Koike-Akino, T., Liu, X., and Feng, C. (2020). Luvli face alignment: Estimating landmarks' location, uncertainty, and visibility likelihood. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8236–8246.

Li, W., Wang, Z., Yin, B., Peng, Q., Du, Y., Xiao, T., Yu, G., Lu, H., Wei, Y., and Sun, J. (2019). Rethinking on multi-stage networks for human pose estimation.

Lin, T.-Y., Dollár, P., Girshick, R., He, K., Hariharan, B., and Belongie, S. (2017). Feature pyramid networks for object detection. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 936–944.

Lin, T.-Y., Maire, M., Belongie, S., Bourdev, L., Girshick, R., Hays, J., Perona, P., Ramanan, D., Zitnick, C. L., and Dollár, P. (2015). Microsoft coco: Common objects in context.

Newell, A., Huang, Z., and Deng, J. (2017). Associative embedding: End-to-end learning for joint detection and grouping. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R., editors, *Advances in Neural Information Processing Systems 30*, pages 2277–2287. Curran Associates, Inc.

Newell, A., Yang, K., and Deng, J. (2016). Stacked Hourglass Networks for Human Pose Estimation. In Leibe, B., Matas, J., Sebe, N., and Welling, M., editors, *Computer Vision – ECCV 2016*, pages 483–499, Cham. Springer International Publishing.

Nie, X., Feng, J., Xing, J., and Yan, S. (2018). Pose partition networks for multi-person pose estimation. In *Computer Vision – ECCV 2018*, pages 684–699, Cham. Springer International Publishing.

Stoffl, L., Vidal, M., and Mathis, A. (2021). End-to-end trainable multi-instance pose estimation with transformers.

Sun, K., Xiao, B., Liu, D., and Wang, J. (2019). Deep high-resolution representation learning for human pose estimation. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5686–5696.

Sun, X., Shang, J., Liang, S., and Wei, Y. (2017). Compositional human pose regression. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2621–2630.

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

Tan, M. and Le, Q. (2019). Efficientnet: Rethinking model scaling for convolutional neural networks. In *International Conference on Machine Learning*, pages 6105–6114. PMLR.

Tang, W. and Wu, Y. (2019). Does learning specific features for related parts help human pose estimation? In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1107–1116.

Tang, W., Yu, P., and Wu, Y. (2018). Deeply learned compositional models for human pose estimation. In Ferrari, V., Hebert, M., Sminchisescu, C., and Weiss, Y., editors, *Computer Vision – ECCV 2018*, pages 197–214, Cham. Springer International Publishing.

Toshev, A. and Szegedy, C. (2014). Deeppose: Human pose estimation via deep neural networks. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1653–1660.

Xiao, B., Wu, H., and Wei, Y. (2018). Simple baselines for human pose estimation and tracking. In Ferrari, V., Hebert, M., Sminchisescu, C., and Weiss, Y., editors, *Computer Vision – ECCV 2018*, pages 472–487, Cham. Springer International Publishing.

Zhao, L., Xu, J., Zhang, S., Gong, C., Yang, J., and Gao, X. (2020a). Perceiving heavily occluded human poses by assigning unbiased score. *Information Sciences*, 537:284–301.

Zhao, M., Beurier, G., Wang, H., and Wang, X. (2020b). A pipeline for creating in-vehicle posture database for developing driver posture monitoring systems. In *DHM2020: Proceedings of the 6th International Digital Human Modeling Symposium, August 31-September 2, 2020*, volume 11, pages 187–196. IOS Press.