2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 25-29, 2020, Las Vegas, NV, USA (Virtual)

Using Diverse Neural Networks for Safer Human Pose Estimation: Towards Making Neural Networks Know When They Don't Know

Patrick Schlosser[1] and Christoph Ledermann[1]

***Abstract*— In recent years, human pose estimation has seen great improvements by the use of neural networks. However, these approaches are unsuitable for safety-critical applications such as human-robot interaction (HRI), as no guarantees are given whether a produced detection is correct or not and false detections with high confidence scores are produced on a regular basis.**

In this work, we propose a method to identify and eliminate false detections by comparing keypoint detections from different neural networks and assigning a 'Don't know' label in the case of a mismatch. Our approach is driven by the principle of software diversity, a technique recommended by the safety standard IEC 61508-7 [1] for dealing with software implementation faults. We evaluate our general concept on the MPII human pose dataset [2] using available ground truth data to calculate a suitable threshold for our keypoint comparison, reducing the number of false detections by approx. 61%. For the application at runtime, where no ground truth data is available, we introduce a method to calculate the needed threshold directly from keypoint detections. In further experiments, it was possible to reduce the number of false detections by approx. 75%. Eliminating keypoints by comparison also lowers the correct detection rate, which we maintained above 75% in all experiments. As this effect is limited and non-critical regarding safety we believe that the proposed approach can lead the way to a safe use of neural networks for human pose estimation in the future.

I. INTRODUCTION

Being able to recognize human body poses in the wild as illustrated in Fig. 1 opens up for a variety of practical applications, including collision avoidance in human-robot interaction (HRI) [3]. These applications need human poses of sufficient quality and reliability, with safety-critical applications like HRI being especially demanding. In recent years, neural networks have shown great potential for supplying high-quality human body poses. The advancements become clear when looking at the large-scale MPII human pose dataset [2] for human pose estimation. According to the official MPII website [4], the last listed approach without a neural network by Pishchulin et al. [5] detected 44.1% of the test samples correctly, while the first listed neural network approach by Tompson et al. [6] improved that number to 79.6%. At the end of 2019, the most successful listed approach by Su et al. [7] pushed the result to 93.9%.

For safety-critical applications, the reliability of the results is crucial. Thus, even Su et al. [7] seem not to be sufficient, as

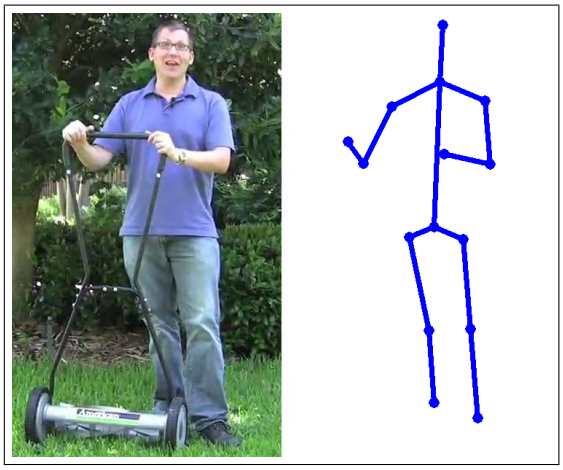


Fig. 1: Illustration of human pose estimation. In an input image (left, [2]) the position of certain human body keypoints like shoulders or ankles (right) is detected.

undiscovered false human keypoint detections are produced in 6.1% of the cases. However, wrong or missing results are not necessarily a problem in safety-critical applications, as long as the application is aware that something is wrong or missing and appropriate countermeasures are applied, e.g. falling back to more conservative safety measures or bringing the system into a safe state. The problem with recent neural networks is their unawareness of their faults - false keypoint detections are produced with high confidence scores on a regular basis. These faults are not limited to small displacement errors: even large displacements errors occur on a regular basis as observed by Ruggero et al. [8] (7.1% and 7.8% of all keypoint detections in two of their experiment were large displacement errors). In addition, Hein et al. [9] have proved that faults with high confidence scores can not be completely avoided while using the ReLU activation function, which is part of most neural networks nowadays.

In this paper, we focus on reducing the number of undetected false human keypoint detections in human pose estimation. Therefore, we transfer the principle of software diversity [1], an established method for dealing with faults in safety-critical software, to neural networks. We employ an approach that uses the keypoints of human poses produced by two or more different neural networks from the same input image and compares them with each other. For the comparison, we employ a distance-based metric based on the PCKh score [2], with the difference that it can be calculated

This work was funded by the Ministry of Economics, Work and Housing of the State of Baden-Württemberg in the research project 'RoboShield'

[1]Intelligent Process Automation and Robotics Lab, Institute of Anthropomatics and Robotics (IAR-IPR), Karlsruhe Institute of Technology (KIT), 76131 Karlsruhe, Germany. Corresponding author: Patrick Schlosser (patrick.schlosser@kit.edu)

without ground truth from the detected poses alone. For mismatches, a 'Don't know' output is introduced to allow the neural networks to express that they don't know. We use the MPII human pose dataset [2] for single person pose estimation to show that our diversity-driven approach with different neural networks and our own comparison metric is capable of reducing the remaining undetected false keypoint detections significantly while keeping the negative impact on correct detections reasonable. Furthermore, we investigate the impact of different neural network architectures on the effectiveness of the diversity-driven approach. We show that very different architectures are most effective, though even very similar architectures lead to noteworthy improvements.

II. RELATED WORK

Human Pose Estimation The field of human pose estimation has been well researched during the last decade. At the beginning of the decade, traditional approaches relying on techniques other than neural networks were dominant, for example approaches based on modifications of the pictorial structure model [5], [10], [11]. As early deep neural network approaches [6], [12] already outperformed the traditional approaches, research shifted into that direction. Today, a broad range of deep neural network approaches is available for human pose estimation, primarily being different in their architecture and detection approach:

For the detection approaches, there exists work that tries to infer the human keypoint locations directly [12], while most of the recent approaches work on so-called heatmaps [7], [13], [14], [15], [16], which indicate for each pixel location how likely it is to be a certain keypoint. These detections can be further refined, e.g. by using a spatial model [6], a separate neural network for fine-localization [13] or using intermediate results from the network in a multi-stage approach and/or for the final output [7], [15].

Having a look at the actual architecture, the use of convolutional neural networks is most common. Some approaches use them in a simple feed forward architecture [6], [12], [13], while others utilize an iterative architecture [17] to further refine detections with each pass. Among recent approaches, the employment of encoder-decoder schemes has become very popular [7], [14], [16], with the hourglass module proposed by Newell et al. [14] being especially noteworthy.

In our work we are going to utilize the neural networks of Tompson et al. [13], Newell et al. [14] and Zhang et al. [16]. Tompson et al. use a convolutional neural network approach that first calculates coarse keypoint locations and then uses a second convolutional neural network to refine these locations. Therefore it is different to Newell et al. and Zhang et al., who both employ an encoder-decoder scheme using hourglass modules. Zhang et al. can be seen as a lightweight version of Newell et al., as it uses the same general architecture, just with fewer and smaller layers. In addition, Zhang et al. perform a knowledge transfer from Newell et al., as they are using their model as a teacher model.

Comparison Metrics In human pose estimation, comparison metrics are ususally used for evaluating a detected pose with respect to an annotated ground truth pose. Among commonly used metrics are the PCP [18], PCKh [2] (used on the MPII human pose dataset [2]) and OKS [19] (used on the COCO dataset [20]). The PCP metric [18] works on body segments and calculates the deviation between the detected and the annotated endpoints of each segment. If this distance is in both cases below 50% of the segment length, then the detected body part is considered to be correct. The PCKh metric aims to have the same threshold for all keypoints of the same person by using the diagonal of the head's bounding box. The OKS metric considers the distance between a detected and an annotated keypoint with respect to the scale of the whole object, but also takes a keypoint-specific factor that indicates the deviation during a redundant annotation process into account, thus punishing deviations on keypoints less that are not completely clear for humans.

Neural Network Errors Errors produced by state of the art neural networks for human pose estimation were investigated by Ruggero et al. [8]. He identified three kinds of possible errors that apply to single persons: small displacement errors ('Jitter'), large displacement errors ('Miss') and errors confusing left and right ('Inversion'), e.g. for shoulders. All these errors occurred frequently.

The research area of adversarial attacks deals with misleading neural networks to produce incorrect results. Apart from designing such an attack, Moosavi-Dezfooli et al. [21] evaluated the errors produced by a classification network under their attack, showing that the errors were very systematic, with only a few output classes being produced from inputs of a large variety of classes.

Hein et al. [9] investigated problems with the usage of ReLU activation fucntions. They developed a method to reduce the occurrence of false results with high confidence when using ReLU activation functions, but also proved that they can not be fully avoided. This leads to the conclusion that the confidence score - at least when it comes down to safety-critical applications - should not be trusted.

The principle of having a 'Don't know' output to indicate that the result of a neural network is not reliable has been successfully employed in safety-critical applications like nuclear power plant monitoring [22] [23], showing that the principle is useful to avoid false results.

Diversity/Redundancy and Neural Networks The standard IEC 61508-7 [1] presents software diversity as a way to deal with safety-critical failures. It is a special form of redundancy, where the same functionality is implemented in different ways. For the same inputs, the produced outputs are compared to make a statement about their validity. It is crucial that the redundantly used software modules do not produce the same false output for the same input.

Redundancy/Diversity is also already used to a certain degree in different neural networks, but usually not to identify false detections but to boost their performance. For example, two-stream networks introduced by Simonyan et al. [24] use two structurally identical neural networks trained and operating on different input types for producing the same kind of output before fusing them by either averaging or

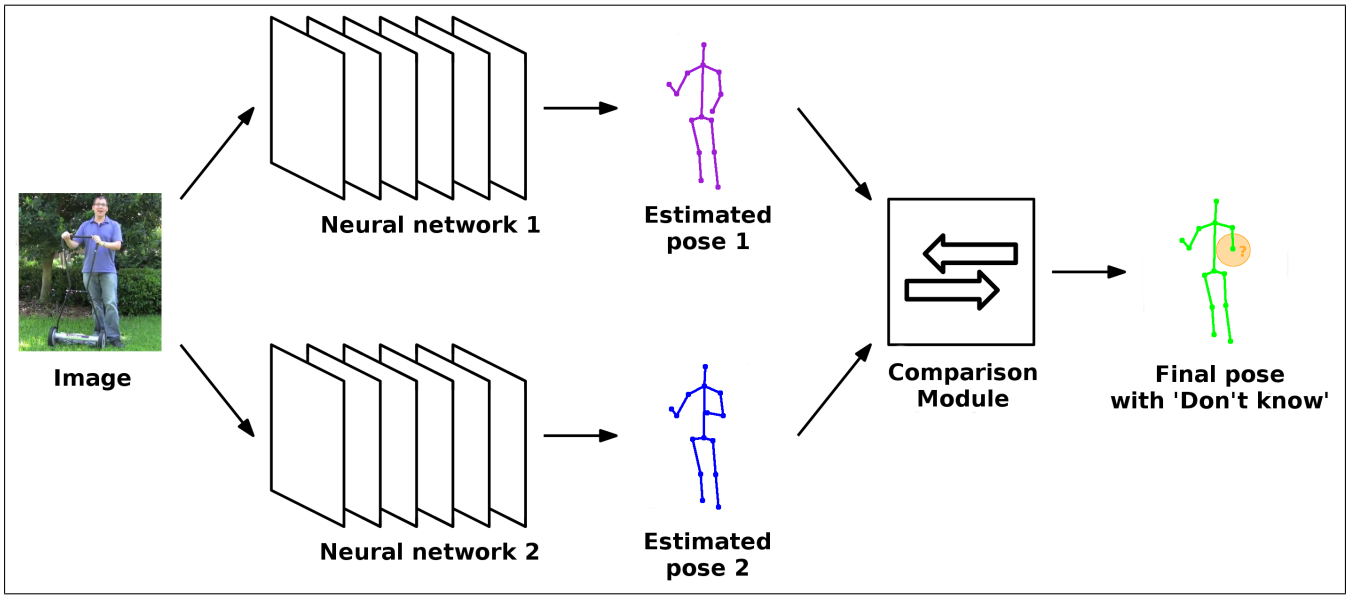


Fig. 2: Proposed diversity-based approach for detecting and eliminating faults in human pose estiamtion. Two different neural networks are used to estimate a pose from the same input image (input image from [2]). Afterwards the poses are fed into a comparison module to compare them, transforming mismatching keypoint detections to the new 'Don't know' output.

applying a linear SVM. For human pose estiamtion, Kawana et al. [25] use an ensemble of CNNs consisting of ten structurally identical networks, which were fine-tuned on separate clusters of training data. The final result is produced by an additional neural network processing the outputs of the ensemble.

To the best of our knowledge, there is no approach for human pose estimation that compares outputs of different neural networks to identify false detections.

III. PROPOSED APPROACH

Based upon the related work in section II we come to the following conclusion: Neural networks are producing faults that are potentially dangerous (e.g. large displacements errors) and the confidence score is not suitable to eliminate them. A potential solution to the problem would be to transfer the diversity-approach suggested by IEC 61508-7 [1] to a setup of different neural networks by comparing their outputs. It is crucial for that approach that the methods for calculation of the output are functionally diverse enough to not produce the same errors on the same input. We are confident that such an approach might work using different neural networks, as many different architectures and approaches exist, errors of neural networks are quite specific (at least under adversarial attacks) and using more than one neural network has already proven to be useful for boosting final scores. For now, we will assume that using different neural networks together in a diversity-based approach is suitable for eliminating a number of false detections - an assumption we are going to validate empirically later.

Based on this assumption we come to our proposed method for reducing false keypoint detections for human pose estimation, visualized in Fig. 2. An input image for which pose estimation shall be performed is used as input for two (or potentially more) neural networks, allowing the user to use his favorite neural network for pose estimation and only having to find a second one (or potentially more). Both neural networks estimate a human pose from the image. These two human poses are used as input into a comparison module, which performs a distance-based comparison between the single keypoints of both poses and produces the final output. To account for mismatches, we introduce a 'Don't know' output for keypoints produced by the comparison module, indicating that the composition of neural networks knows nothing about the specific keypoint. Let $kp_{1,i}$ and $kp_{2,i}$ denote the detections for keypoint i from the first and the second neural network. Furthermore, let $threshold_{matching}$ denote a person- and image-specific distance threshold and kp_i the final output for keypoint i produced by the comparison module. To calculate kp_i we will use the following mathematical formulation:

$$kp_i = \begin{cases} \frac{kp_{1,i}+kp_{2,i}}{2}, & \text{if } kp_{1,i}, kp_{2,i} \text{ exists } \& \\ & \|kp_{1,i} - kp_{2,i}\|_2 \leq \\ & threshold_{matching} \\ \text{'Don't know'}, & \text{otherwise} \end{cases} \quad (1)$$

In simple words, if both neural networks detected the specific keypoint (did not report it as missing) and the euclidean distance of both keypoints is below $threshold_{matching}$, the average of both keypoint locations is provided as final output. In any other case, we will put out a 'Don't know' label for the keypoint. Therefore, missing keypoints in one or both neural networks will lead to a 'Don't know' output just as mismatching detections do. Missing keypoints can have two

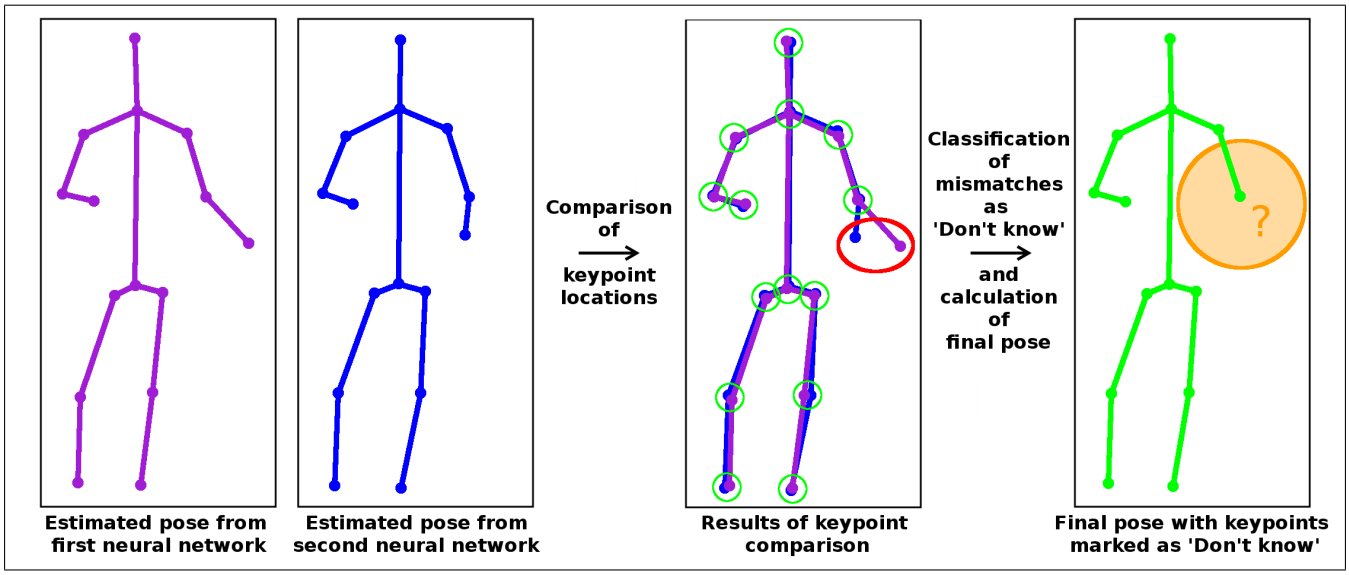


Fig. 3: Illustration of the matching process. Keypoint positions from two estimated poses are compared, with matching keypoints (green circle) being averaged and mismatching keypoints (red circle) producing the new 'Don't know' output.

reasons: 1) The keypoint location is outside of the image or 2) The keypoint location is inside the image and the neural network(s) failed to detect it. As case 2) is dangerous in safety-critical applications and we see no downside in treating missing keypoints as 'Don't know', we decided to handle them this way. To further facilitate understanding of the matching process, Fig. 3 illustrates it visually.

Throughout the following sections of the paper, we are going to show that the proposed approach is useful for eliminating false detections and how a suitable threshold can be calculated from data available at runtime.

IV. PROOF OF CONCEPT

In this section, we are going to show the capability of our diversity-based approach to eliminate a certain amount of false detections. To show that the general approach of comparing poses produced by different neural networks is a useful idea, we are going to simplify the overall problem. Instead of calculating the necessary thresholds for matching the poses by ourselves, we are going to use threshold values annotated in the ground truth data. This way we omit a potential additional error source. The calculation of suitable thresholds from data being available at runtime will be subject to section V.

To evaluate our approach we are going to take advantage of the MPII human pose dataset [2] by using the samples and annotations for single person detection provided by it. MPII has the upside of being a challenging dataset with more than 40.000 annotated persons in approximately 25.000 images. Images are taken from 410 different activities in the wild, thus delivering a broad and representative collection of human poses. On the downside, the test set of MPII is kept private by the developers, thus reducing the publicly available data to the training set with around 29.000 annotated poses and prohibiting self-evaluation on the proposed test set. From the training set, Tompson et al. [13] extracted almost 3000 samples to form a separate validation set, which was also adopted by Newell et al. [14] and Zhang et al. [16]. Our evaluation will be performed on this validation set.

To determine if a detected body keypoint matches the ground truth, MPII uses the so-called PCKh score [2]. To calculate this score, an annotated bounding box around the head is necessary in addition to the annotated ground truth locations of the body keypoints. For keypoint i, the distance between the detected keypoint $kp_{det,i}$ and the ground truth keypoint $kp_{gt,i}$ is calculated, as well as the length of the diagonal len_{diag_head} of the annotated head bounding box. To obtain the PCKh score for $kp_{det,i}$ with respect to $kp_{gt,i}$, the calculated distance between the keypoints is divided by len_{diag_head}. Therefore the PCKh score is a normalized distance measure. The actual evaluation script provided for MPII by Andriluka et al. [2] applies an additional scaling factor of 0.6 to len_{diag_head} before performing the division, resulting in the following formula:

$$\text{PCKh}(kp_{det,i}, kp_{gt,i}) = \frac{\|kp_{det,i} - kp_{gt,i}\|_2}{0.6 \cdot len_{diag\_head}} \quad (2)$$

Using the PCKh score, $kp_{det,i}$ is considered to be detected correctly with respect to the corresponding $kp_{gt,i}$, if the PCKh score is below or equal to a certain normalized distance threshold, with 0.5 being the standard value used.

To prove that our general concept for eliminating false keypoint detections works, we are going to adopt this measure for determining if the keypoint detections of both neural networks match. Looking at (1) the condition for two keypoint detections to match is, despite their existence, that the following equation holds true:

$$\|kp_{1,i} - kp_{2,i}\|_2 \leq threshold_{matching} \quad (3)$$

First we are going to rewrite $threshold_{matching}$ as a product of a normalized distance threshold $norm\_thres_{matching}$ and a distance for normalization $norm\_dist$:

$$threshold_{matching} = norm\_thres_{matching} \cdot norm\_dist \tag{4}$$

If we now use (4) in (3) and divide both sides by $norm\_dist$ (assuming $norm\_dist > 0$) we get a new formulation for our keypoint matching condition:

$$\frac{\|kp_{1,i} - kp_{2,i}\|_2}{norm\_dist} \leq norm\_thres_{matching} \tag{5}$$

Using $norm\_dist = 0.6 \cdot len_{diag\_head}$, the left side of (5) equals the calculation of the PCKh score in (2). This way, we can easily use the PCKh score for matching our poses one with another just as we match a detected pose with the ground truth, with $norm\_thres_{matching}$ controlling how restrictive the matching process is. Throughout all of our experiments, we are going to use $norm\_thres_{matching} = 0.5$ unless stated otherwise.

For our experiments in this section we are going to use the work of Tompson et al. [13] and Newell et al. [14], as their architectures are very different and we expect better results from more diverse networks. To measure the impact of our approach on the number of false detections, we introduced a new measure on the MPII dataset: The percentage of false detections in relation to the normalized distance threshold used for matching the estimated pose with the ground truth. To achieve similar evaluation conditions, missing detections when evaluating single networks will not be counted as false detections, as our approach with multiple networks treats them as 'Don't know'. While we observe different normalized distance thresholds for matching the final detection with the ground truth, our normalized distance threshold for matching the detections of two neural networks before producing the final output remains static at 0.5 as mentioned before. We are also going to have a look at the percentage of correct detections in the same way. Fig. 4 shows the results using the two mentioned neural networks on their own and our redundant usage of them. At the normalized distance threshold of 0.5 it can be seen that the number of false detections was reduced from 11.6% to 4.5% in relation to the better single neural network, thus leading to an elimination of about 61% of the previously undetected false detections. However, this result comes also at the cost of correct detections when formerly correct detections are shifted to the 'Don't know' label, for example when one neural network outputs a correct result but it does not match with the other ones result. As seen in Fig. 4, the amount of correct detections for our approach at 0.5 is reduced from 82.3% to 79.9% compared to the worse neural network. In this case, we compare to the worse neural network, as it imposes a soft limit on the performance of our approach because of the performed keypoint comparison. The limit is soft as in some cases a correct average keypoint can emerge from matching keypoint detections with one

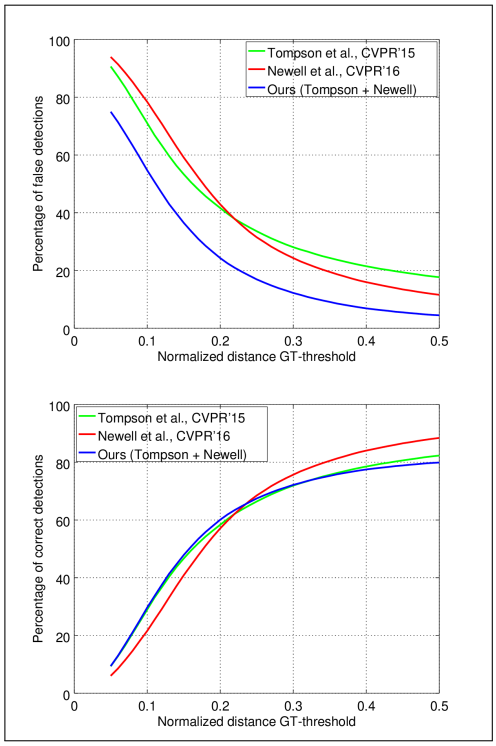


Fig. 4: Results of our approach with the networks of Tompson et al. [13] and Newell et al. [14] compared to the performance of the single neural networks. For matching the poses of both networks in our approach, data from MPII ground truth annotations and a steady normalized distance threshold of 0.5 was used, while different thresholds for matching detections to the ground truth were evaluated.

(or more) detections not being correct with respect to the ground truth. Examples can be found in Fig. 5. Looking at the relation of lost correct detections and eliminated false detections, our approach loses one correct detection per 2.94 eliminated false detections.

The results show, that our approach has the potential of eliminating a high amount of false detections, while being able to keep the negative impact on correct detections small. For application at runtime, however, a method for calculating a suitable matching threshold from data being available at that point needs to be used, which will be the subject of the following section V.

V. DISTANCE THRESHOLD DESIGN

As the missing piece for application at runtime, we need a way to realize the comparison of different poses with data being available at that point, thus not using ground truth annotations. As we are evaluating on the MPII dataset, we decided to stick with our formulation from (5) for thresholding, breaking down the problem to calculating $norm\_dist$. The most straightforward approach with respect to the MPII dataset would be to employ an additional neural network to calculate head bounding boxes to extract their diagonal, which we decided against as another neural network would

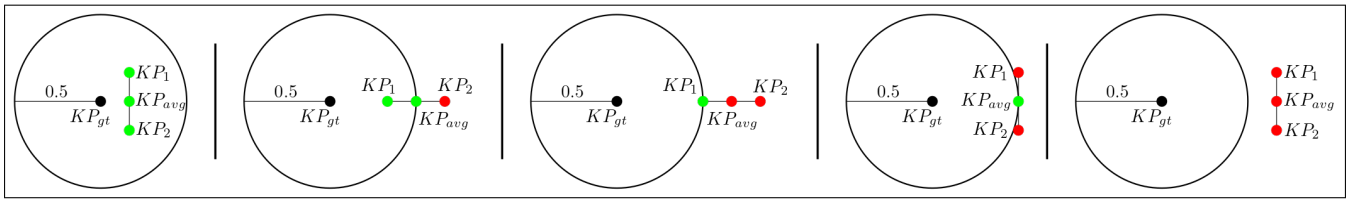


Fig. 5: Examples for possible results for two keypoints KP_1 and KP_2 that match according to the comparison module, thus producing the average keypoint KP_{avg}. Green dots mark keypoints that would be evaluated as correct with respect to the ground truth keypoint KP_{gt} and the normalized distance threshold 0.5, red dots mark keypoints that would not. In special cases, it is possible that a valid average keypoint is produced from two matching keypoints despite one or both of them would not be valid with respect to the ground truth.

pose a new potential error source as well as consuming additional computational power. Rather we decided to rely on the pose data already available from the different neural networks. The human keypoints we are going to use should have two characteristics: They are detected reliably and are suitable for conservatively estimating the head diagonal. The keypoints 'top head' and 'upper neck' have both characteristics, being the most reliable keypoints (see Table I) and can be used to estimate the height of the head len_{head}, which is a conservative (smaller) estimation for the head diagonal. Using only two keypoints is not enough, as their distance can vanish in a two-dimensional image when forming a line with the camera. We choose the shoulder keypoints as an additional pair of keypoints to calculate the shoulder width $len_{shoulder}$. They have high correct detection rates as well and can not vanish at the same time as the head keypoints due to their relative position to the camera because of anatomic constraints. We can furthermore estimate the height of the head with the distance between both shoulders. Using the work of Winter [26], who formulated the length of human body parts as a function of the body height, the relation of shoulder width to head height is approximately 2:1, which we are going to use. After calculating len_{head} and $len_{shoulder}$ we are applying a 0.6 scaling factor to both lengths just as the original MPII evaluation script does to the head diagonal, and an additional 0.5 scaling factor to the shoulder length to account for the 2:1 ratio. To take potential false detections of keypoints into account, we are performing this process for each estimated pose. Afterwards we are going to calculate the average lengths avg_len_{head} and $avg\_len_{shoulder}$. If the deviation between the single values for len_{head} respectively $len_{shoulder}$ from the different poses is larger than 50% or one of them is missing, the corresponding average value is set to zero as a protection against possible wrong length values. The final value we are going to use for $norm\_dist$ is the maximum of the average lengths and an additional minimal length len_min:

$$norm\_dist = max(avg\_len_{head}, avg\_len_{shoulder}, len\_min) \quad (6)$$

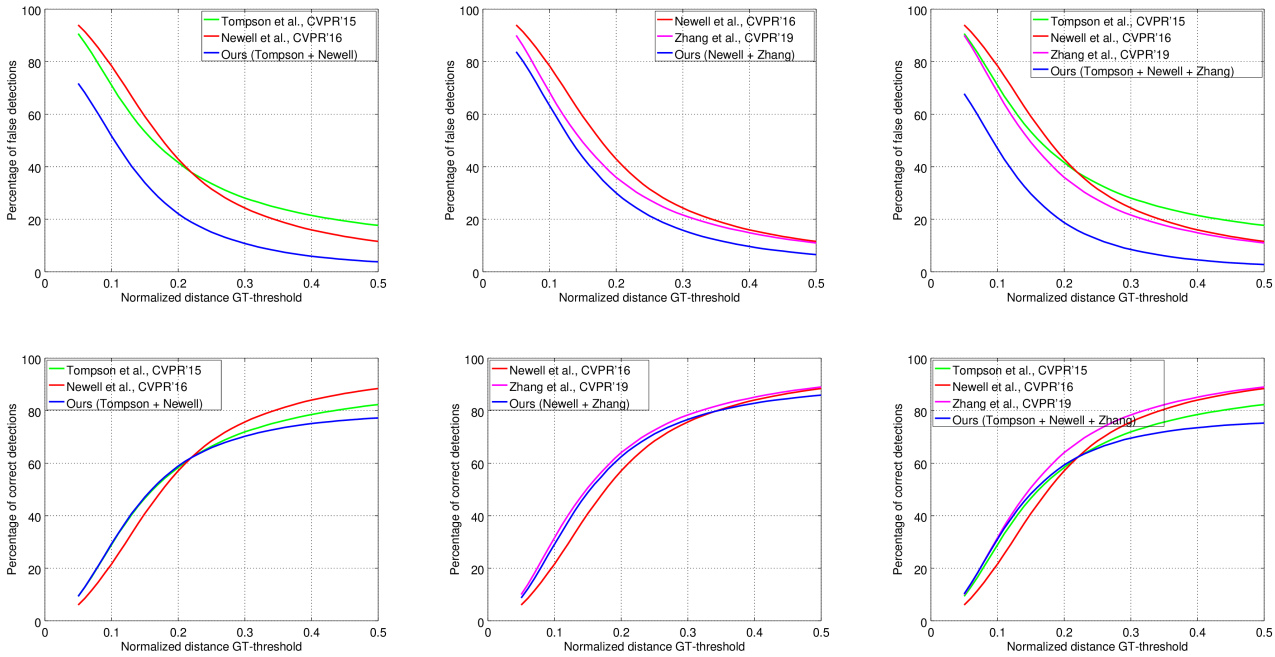
The additional len_min is seen as a protection against missing average lengths and is calculated as the average over the smallest 5% of annotated head diagonals from the training set after scaling them with 0.6.

We tested our methodology on the training and validation set to see if we usually achieve a smaller or equal estimation of the head diagonal length after scaling. On the training set, we achieved this for 98.1% of the cases and on the validation set for 97.4%. Being smaller or equal to a 50% increase of the scaled head diagonal length was achieved for 99.93% of the cases on the training set and for 99.86% of the cases on the validation set. Given the results, we have now a conservative method of estimating $norm\_dist$ at runtime.

VI. FURTHER EVALUATIONS

First, we are going to show that our approach works at runtime by repeating our experiment from section IV using our own calculation for $norm\_dist$ from section V instead of relying on ground truth data for comparing the keypoints of different poses. The results of the experiment can be seen in Fig. 6a. The number of false detections was reduced from 11.6% to 3.8%, eliminating about 67% of the remaining false detections, which is 6% more compared to the ground truth usage. With our own calculation of $norm\_dist$ being designed more conservatively than using the ground truth, this result is not surprising. However, this improvement comes at the cost of losing more correct detections, with the percentage being decreased from 82.3% to 77.2%. This corresponds to one lost correct detection per 1.52 eliminated false detections, which is worse than one lost correct detection per 2.94 eliminated false detections in the experiments using ground truth data. In total, these results show that our approach is applicable at runtime without using ground truth data.

Until now we have only used neural networks with a very different architecture. Therefore we want to investigate the behaviour of our approach with very similar architectures, using the neural networks of Newell et al. [14] and Zhang et al. [16]. We repeat the previous experiment using these networks, with the results being shown in Fig. 6b. Despite Zhang performing better than Tompson on its own, the combination of Zhang and Newell is not capable of eliminating more false detections, only reducing the percentage of false detections from 11.0% (Zhang) to 6.5% and thus eliminating only about 40% of the remaining false detections. Correct



(a) Results with Tompson et al. [13] and Newell et al. [14]

(b) Results with Newell et al. [14] and Zhang et al. [16]

(c) Results with Tompson et al. [13], Newell et al. [14] and Zhang et al. [16]

Fig. 6: Experimental results for our approach with different setups of neural networks on the MPII dataset [2]. The percentage of false detections and correct detections are examined for each setup in relation to the threshold for matching detections with the ground truth (GT). During all experiments, the normalized distance threshold $norm\_thres_{matching}$ for matching the different poses in our approach before producing the final output is kept fixed at 0.5.

detections are reduced from 88.4% to 85.9%, resulting in one lost correct detection per 1.77 eliminated false detections.

The previous experiments support our initial assumption, that architectures with high difference yield better results for eliminating false detections in our diversity-based approach. In an additional experiment, we are going to use all three architectures together. We want to investigate if there is any value of adding a neural network to a setup that already contains a similar and a different one. The experiment will be performed similar to the previous ones in this section, but with all calculations of our approach now being performed on 3 instead of 2 detected poses. This includes the need for all 3 detections to match. The results can be seen in Fig. 6c. The number of false detections has been reduced from 11.0% to 2.8%, being 1% lower than the 3.8% from Tompson and Newell alone. The reduction corresponds to an elimination of about 75% of the remaining false detections. The amount of correct detections was reduced from 82.3% to 75.3%, resulting in one lost correct detection per 1.17 eliminated false detections. Based on the results, using Zhang together with Newell and Tompson despite it's similarity to Newell added additional information useful for eliminating false detections, as the best results among all experiments were achieved, with only 2.8% false detections remaining. For further information, a detailed summary of all results from the experiments performed in this section can be found in Table I.

VII. CONCLUSION

In this paper, we have introduce a new approach to deal with false detections of human poses from neural networks, which is especially important for safety-critical applications like HRI. Our approach is inspired by software diversity. By comparing the human poses produced by different neural networks, we introduced a 'Don't know' label for mismatching human body keypoint detections, which enables us to eliminate formerly undiscovered false detections at the cost of correct detections. Furthermore, we introduced a method to compare our poses with data being available at runtime, based on the PCKh score. In our best experiment, we were able to reduce the number of false detections from 11.0% to 2.8%, while achieving correct detections in 75.3% of the cases, despite including a neural network with only 82.3% correct detections on its own in this setup. From a functional safety perspective it should be pointed out that losing some correct detections in the process of eliminating false detections is not a safety problem, as one can always fall back to more conservative safety strategies when a 'Don't know' output occurs. We also investigated different combinations of neural networks in our experiments, showing that

Method	Head		Shoulder		Elbow		Wrist		Hip		Knee		Ankle		Mean		
	C	F	C	F	C	F	C	F	C	F	C	F	C	F	C	F	DK
Tompson et al. [13]	96.6	3.4	92.7	7.3	83.4	16.6	76.7	23.3	81.9	18.1	73.3	26.7	65.4	34.6	82.3	17.7	0.0
Newell et al. [14]	96.8	3.2	95.2	4.8	89.1	10.9	84.2	15.8	87.0	13.0	83.2	16.8	80.4	19.6	88.4	11.6	0.0
Zhang et al. [16]	97.5	2.5	95.5	4.5	89.0	11.0	84.3	15.7	88.9	11.1	84.1	15.9	80.7	19.3	89.0	11.0	0.0
Ours (Tompson + Newell)	93.7	0.9	89.4	1.9	78.7	3.9	71.1	5.0	75.2	5.9	66.7	4.9	58.9	4.6	77.2	3.8	19.0
Ours (Tompson + Zhang)	95.0	0.9	90.4	2.3	79.8	4.7	72.4	6.0	78.0	6.2	68.3	6.1	60.6	5.6	78.8	4.5	16.7
Ours (Newell + Zhang)	96.5	1.9	94.2	2.9	86.4	6.6	80.0	8.3	85.5	8.8	79.4	8.8	75.4	9.8	85.9	6.5	7.6
Ours (Tompson + Newell + Zhang)	93.3	0.6	88.2	1.6	76.7	2.9	68.9	3.4	72.3	4.6	64.1	3.4	56.4	3.0	75.3	2.8	21.9

TABLE I: Results on the MPII dataset [2] for single neural networks and our diversity-based approaches using a normalized distance threshold of 0.5 to the ground truth. Our approaches used the distance threshold design from section V with $norm\_thres_{matching} = 0.5$ for matching different poses before the evaluation on the ground truth. Legend: C - percentage of correct detections; F - percentage of false detections; DK - percentage of 'Don't know' outputs.

our approach profits more from different architectures being employed than similar ones.

Further research needs to be done to perform a detailed analysis of the impact of training in a diversity-based approach, up to potentially introducing higher diversity for multiple neural networks by joint training. For practical application, we will usually have to calculate 3D poses from our 2D poses, thus it would be useful to develop a 2D keypoint-specific matching threshold that corresponds to a fixed safety distance in 3D space.

ACKNOWLEDGMENT

We want to thank Aiden Nibali for adjusting the official MPII evaluation script to the validation set used by Tompson et al. [13] and sharing his work on GitHub[1]. We used his work as a basic foundation for our own evaluation.

REFERENCES

[1] *Functional safety of electrical/electronic/programmable electronic safety-related systems – Part 7: Overview of techniques and measures*, International Electrotechnical Commission (IEC) Std. IEC 61 508-7:2010, 2010.

[2] M. Andriluka, L. Pishchulin, P. Gehler, and B. Schiele, "2d human pose estimation: New benchmark and state of the art analysis," in *Proceedings of the IEEE Conference on computer Vision and Pattern Recognition*, 2014, pp. 3686–3693.

[3] D. H. P. Nguyen, M. Hoffmann, A. Roncone, U. Pattacini, and G. Metta, "Compact real-time avoidance on a humanoid robot for human-robot interaction," in *Proceedings of the 2018 ACM/IEEE International Conference on Human-Robot Interaction*, 2018, pp. 416–424.

[4] (2019) Mpii human pose dataset results. Acessed: 2019-12-11. [Online]. Available: http://human-pose.mpi-inf.mpg.de/#results

[5] L. Pishchulin, M. Andriluka, P. Gehler, and B. Schiele, "Strong appearance and expressive spatial models for human pose estimation," in *Proceedings of the IEEE international conference on Computer Vision*, 2013, pp. 3487–3494.

[6] J. J. Tompson, A. Jain, Y. LeCun, and C. Bregler, "Joint training of a convolutional network and a graphical model for human pose estimation," in *Advances in neural information processing systems*, 2014, pp. 1799–1807.

[7] Z. Su, M. Ye, G. Zhang, L. Dai, and J. Sheng, "Cascade feature aggregation for human pose estimation," 2019.

[8] M. Ruggero Ronchi and P. Perona, "Benchmarking and error diagnosis in multi-instance pose estimation," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 369–378.

[9] M. Hein, M. Andriushchenko, and J. Bitterwolf, "Why relu networks yield high-confidence predictions far away from the training data and how to mitigate the problem," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 41–50.

[10] S. Johnson and M. Everingham, "Learning effective human pose estimation from inaccurate annotation," in *CVPR 2011*. IEEE, 2011, pp. 1465–1472.

[11] Y. Yang and D. Ramanan, "Articulated human detection with flexible mixtures of parts," *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 12, pp. 2878–2890, 2012.

[12] A. Toshev and C. Szegedy, "Deeppose: Human pose estimation via deep neural networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 1653–1660.

[13] J. Tompson, R. Goroshin, A. Jain, Y. LeCun, and C. Bregler, "Efficient object localization using convolutional networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 648–656.

[14] A. Newell, K. Yang, and J. Deng, "Stacked hourglass networks for human pose estimation," in *European conference on computer vision*. Springer, 2016, pp. 483–499.

[15] Z. Cao, G. Hidalgo, T. Simon, S. Wei, and Y. Sheikh, "Openpose: Realtime multi-person 2d pose estimation using part affinity fields," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019.

[16] F. Zhang, X. Zhu, and M. Ye, "Fast human pose estimation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 3517–3526.

[17] J. Carreira, P. Agrawal, K. Fragkiadaki, and J. Malik, "Human pose estimation with iterative error feedback," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 4733–4742.

[18] V. Ferrari, M. Marin-Jimenez, and A. Zisserman, "Progressive search space reduction for human pose estimation," in *2008 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2008, pp. 1–8.

[19] (2020) Coco keypoints evaluation. Accessed: 2020-02-25. [Online]. Available: http://cocodataset.org/#keypoints-eval

[20] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *European conference on computer vision*. Springer, 2014, pp. 740–755.

[21] S.-M. Moosavi-Dezfooli, A. Fawzi, O. Fawzi, and P. Frossard, "Universal adversarial perturbations," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1765–1773.

[22] C. d. A. Antônio, A. S. Martinez, and R. Schirru, "A neural model for transient identification in dynamic processes with "don't know" response," *Annals of Nuclear Energy*, vol. 30, no. 13, pp. 1365–1381, 2003.

[23] V. H. C. Pinheiro, M. C. dos Santos, F. S. M. do Desterro, R. Schirru, and C. M. d. N. A. Pereira, "Nuclear power plant accident identification system with "don't know" response capability: Novel deep learning-based approaches," *Annals of Nuclear Energy*, vol. 137, p. 107111, 2020.

[24] K. Simonyan and A. Zisserman, "Two-stream convolutional networks for action recognition in videos," in *Advances in neural information processing systems*, 2014, pp. 568–576.

[25] Y. Kawana, N. Ukita, J.-B. Huang, and M.-H. Yang, "Ensemble convolutional neural networks for pose estimation," *Computer Vision and Image Understanding*, vol. 169, pp. 62–74, 2018.

[26] D. A. Winter, *Biomechanics and motor control of human movement*. John Wiley & Sons, 2009.

[1] https://github.com/anibali/eval-mpii-pose