2023 IEEE International Conference on Robotics and Automation (ICRA 2023)
May 29 - June 2, 2023. London, UK

Robust Human Pose Estimation under Gaussian Noise

Patrick Schlosser[1] and Christoph Ledermann[1]

Abstract— **Robustness against specific kinds of noise is of high importance for safety-critical components in industrial robot applications, as legal and normative regulations demand the identification and handling of all unacceptable risks. This includes risks from environmental conditions, like noisy data.**

One such component is human pose estimation, which is needed and crucial for human-robot collaboration tasks and applications. However, little research on human pose estimation under specific noise types has been performed. In our work, we focus on extensively evaluating human pose estimation under specific noise and propose potential countermeasures. We leverage Gaussian noise as specific noise type and the hourglass model as human pose estimator. We show that human pose estimation is already vulnerable to small amounts of Gaussian noise. As countermeasures we propose either denoising images upfront or training the hourglass model to be robust against Gaussian noise. All methods achieve a significantly higher robustness against Gaussian noise, typically at the cost of slightly worse performance on clean data. Three of our methods also achieved slight improvements on clean data.

I. INTRODUCTION

A reliable detection of the human is the foundation for a variety of tasks and applications in human robot collaboration (HRC) [1], [2]. One possibility to obtain such detections is to employ human pose estimation, which detects the location of important body keypoints, like the wrists. The currently leading approaches [3], [4] for single person human pose estimation on the MPII Human Pose Dataset [5] achieve 94.1% correct results, which seems pretty reliable.

But for safety-critical applications, this is still not sufficient to fulfill European safety regulations. These regulations are formalized in safety standards, like the general ISO 13849 [6] and the robotic-specific ISO 10218 [7]. Demands for safety-relevant parts in HRC include an upper limit of 10^{-6} for dangerous errors per hour [6], [7]. This must be achieved within a predefined range of environmental conditions [8], including potential data corruption by noise.

The impact and handling of noise in human pose estimation is rarely addressed in current research: typically, evaluation is just performed on 'clean' data of large scale human pose estimation datasets [5], [9]. Only recently, Wang et al. [10] showed the vulnerability of several state-of-the-art networks [11], [12], [13], [14] to different kinds of noise (e.g. Gaussian, defocus, fog). They propose an adversarial learning approach to achieve higher robustness against noise types unseen during training. In a short ablation study they show that training with a subset of noise types outperforms their approach when testing against the same subset, while their method is superior for most unseen noise types.

Fig. 1: An image from the MPII Human Pose Dataset [5], once clean (left) and once affected by Gaussian noise (right).

Hence, an approach like the one proposed in Wang et al. [10] is best when the noise types that can be encountered are not previously know. For safety-critical industrial settings however, the predefined environment and environmental conditions limit the possible noise types. Furthermore, safety-regulations make it necessary to identify hazards originating from the environment and to counteract them [8], [7]. Hence, it can be assumed that all dangerous noise types that can be encountered in such a setting are previously known. For more detailed reasoning on this statement, see section II.

Thus we argue that - for human pose estimation aimed at use in safety-critical applications - it is important to investigate noise-specific countermeasures in greater detail. To be able to provide an extensive evaluation, we limit our investigation to one specific noise type, the frequently occurring Gaussian noise (see Fig. 1), in combination with a lightweight version of the popular hourglass model [15] for human pose estimation. First, we investigate how Gaussian noise of different severity affects human pose estimation, showing an early and significant performance decrease. Next, we investigate two general strategies to handle Gaussian noise - a) applying training strategies including noise and b) using denoisers [16], [17] before human pose estimation with/without retraining. Furthermore, we investigate the impact of slight noise type changes at test time, complementing the research that already showed the negative impact of major changes. Summarized, our contributions are:

- An analysis of European safety standards, concluding in the viability of noise-specific countermeasures.
- An investigation and extensive evaluation of human pose estimation under varying degrees of Gaussian noise and the impact of countermeasures.
- An analysis of the impact of small noise type alterations between training noise and test noise.

This work was funded by the German Federal Ministry for Economic Affairs and Climate Action in the research project 'FabOS'

[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)

II. BACKGROUND AND MOTIVATION

A. Introduction to Safety Regulations

The Machinery Directive (Directive 2006/42/EC of the European Parliament and of the Council) [18] is the central directive that regulates - among others - safety-relevant aspects of machinery, partly completed machinery and safety components within the European Union. Regulations apply to manufacturers of such goods. This term explicitly includes everyone who puts machinery into service (see article 2(i)) [18]. Therefore, when assembling a new robot cell and putting it into service, the Machinery Directive applies. Its regulations must be obeyed and conformity must be proven. To do so, harmonized standards can be employed (see Article 7(2) [18]). For industrial robots, the relevant harmonised standard is ISO 10218 [7]. It covers - among other things - the safety requirements for HRC applications and references ISO 12100 [8] for hazard identification and risk reduction.

B. Why is resistance against specific noise types important?

The detailed regulations of ISO 10218 [7] and ISO 12100 [8] set a robotic application running in an industrial robot cell clearly apart from a robotic application running in the wild. First, ISO 10218-2 section 4.3.2 requires the definition of the 'limits of the robot system' [19]. These limits consider a variety of factors, e.g. intended use and foreseeable misuse as well as the operation environment and environmental conditions. A simple example would be a required ambient temperature, e.g. between 10° and 40° Celsius. Hence, the environmental conditions that can be encountered during operation are well defined.

ISO 10218-2 [19] also requires the identification of potential hazards and associated risks for the human. Therefor the standard is largely based on and frequently refers to ISO 12100 [8] and its contents. According to ISO 10218-2 [19] and ISO 12100 [8], all reasonably probable hazards must be identified and suitable safety measures must be applied. This includes the identification of hazards arising from not properly functioning components, e.g. because of environmental conditions, as mentioned in section 5.4.b.2 of ISO 12100 [8]. Section B.2 of ISO 12100 [8] details, that the origin and/or the consequences of the hazards should be documented if it is useful for the choice of safety measures. As an example, assume a robot cell that uses a human pose estimator to monitor the distance between a robot and a human. Then, Gaussian noise caused by bad illumination could lead to a malfunction of the human pose estimator (origin), resulting in not detecting a human and continued robot operation. Then a collision with the robot can occur (consequence).

For a specific robot cell, these requirements mean that the robot's environment is well defined and that all noise types which can lead to dangerous malfunctions within this specific environment are identified. Knowing all noise types makes general countermeasures against noise unnecessary - instead of generalization ability, high efficiency against the specific noise types is necessary. Therefore, countermeasures tailored towards these noise types can and should be employed.

III. RELATED WORK

A. Human Pose Estimation

Human pose estimation aims at localizing important human keypoints (e.g. shoulder, elbow, wrist), typically from image data. One of the most fundamental tasks is hereby the localization of keypoints from a single human in a 2D image. Current state of the art approaches for this task [3], [4], [13], [15], [20], [21], [22] typically predict so-called heatmaps. They contain per-pixel scores which indicate how likely it is that a specific keypoint is located at the pixel-location (one heatmap per keypoint). Among these approaches is the stacked hourglass model of Newell et al. [15] that uses several consecutive encoder-decoder blocks (hourglass modules). Each block predicts and iteratively improves heatmaps. The model is a particularly important milestone, as it serves as a foundation for many state-of-the-art networks [3], [20], [21], [22]. Thus, a lightweight version of the stacked hourglass model is used in our work.

More complex human pose estimation tasks include predicting the poses of several humans in the same image. This can either be done by first localizing each human and then applying a human pose estimation method for single persons [23], [24] or by predicting all keypoints in the image and associating them with individual entities [11], [25]. Another advanced task is the prediction of 3D human poses. Some approaches [26], [27] consist of two steps, where the 2D pose is predicted before the final 3D pose. Therefore, they depend on an accurate and robust 2D pose estimator.

B. Human Pose Estimation under Noise

Dealing with noise in human pose estimation sees limited research. Some two-step 3D human pose estimation approaches [27], [28], [29] consider noise in form of inaccurate 2D keypoint locations from the first step. Dealing with artificial image noise in form of adversarial attacks in human pose estimation also sees some research, like the work of Shah et al. [30]. For natural image noise, not even the latest comprehensive human pose estimation survey by Zheng et al. [31] mentions a single approach. Only recently, Wang et al. [10] brought attention to this topic.

In their work, Wang et al. [10] showed the vulnerability of different state-of-the-art networks for multi-person 2D human pose estimation to various natural kinds of noise, like Gaussian noise, snow or fog. They proposed a training procedure AdvMix [10] based on noise augmentation in adversarial fashion as well as knowledge distillation to improve the robustness against unseen types of noise. They improved the results of human pose estimation under unseen noise, but a considerable negative impact remained. In a short ablation study they showed that utilizing a subset of test noise types for training can improve the robustness against those types, outperforming their approach on these noise types. On the downside, this procedure impacted the performance on clean data negatively and did not necessarily improve the performance for other noise types.

C. Image Denoising/Robustness against Noise

A large research area in computer vision is dedicated to denoising images, investigating deep-learning and traditional, handcrafted approaches alike. A traditional example is the BM3D algorithm [16], originally aimed at denoising grayscale images, which was extended to color images [16], [32] and has recently seen faster GPU-based implementations [33], [34]. Deep-learning approaches include e.g. the rather simple yet efficient DnCNN [35] and FFDNet [17]. The latter uses a feed-forward convolutional architecture and takes a noisy image as well as a noise level map as input to calculate a denoised image. The noise level map is hereby used to control the denoising strength for different image areas. So, one solution to deal with noisy images in arbitrary tasks would be to deploy such image denoisers upfront.

An alternative solution - for neural networks - is the use of training strategies against noise. Such approaches have been successfully employed for image classification: Geirhos et al. [36] added noise to images during training to achieve higher robustness. Zheng et al. [37] proposed to alter the loss function used during training. They add the distance between the result for the clean image and the result for the same image altered with Gaussian noise to the task-specific loss. Xi et al. [38] proposed to train a student model exposed to model noise and input noise based on annotated labels as well as outputs from a teacher model. The process is repeated several times, always using the latest student model as new teacher. Apart from image classification, utilizing noise during training is also a common defense against adversarial attacks [39], [40]. These attacks apply nearly imperceptible artificial noise to images to make neural networks malfunction. Typically, higher robustness is achieved against the same attack that is used during training, but transferability to other attacks is not guaranteed [40].

IV. Methods

Given our argumentation in section II, we assume that all relevant noise types in safety-critical applications are previously identified. Therefore, noise-specific countermeasures can be used. The noise types will be application-specific; for our study, we use Gaussian noise as exemplary noise type due to its frequent occurrence, e.g. through bad illumination. To negate its impact, we investigate two general strategies:

1) Employing a denoiser before a human pose estimator
2) Adapting training for robustness against noise

A. Human Pose Estimator

As human pose estimator, we utilize a lightweight version of the original stacked hourglass model by Newell et al. [15], as it is the common foundation for numerous human pose estimation networks (see section III-A). Our lightweight version uses two instead of eight consecutive hourglass modules. While it does not have state-of-the-art performance, it only consists of the parts typically shared by many of the newer networks. Thus, our final results will originate from this common structure and not from specialised improvements like the soft-gated skip connections of Bulat et al. [3].

B. Denoiser Against Noise

Our first general strategy against Gaussian noise in human pose estimation is the usage of a dedicated image denoiser to denoise noisy images before they serve as input for the human pose estimator. We investigate two different image denoisers: FFDNet [17] and BM3D [16]. For FFDNet, we follow the suggestions of the author for color images by using a version with 12 convolutional layers and 96 filters per convolutional layer except for the last one which has 12 filters [17]. For BM3D, we use the publicly available [41], GPU-accelerated version by Honzátko and Kruliš [33].

C. Training Against Noise

The second general strategy we investigate is training the human pose estimator to be robust against Gaussian noise. To achieve this, we augment each sample of the training data with a probability of $X\%$ with Gaussian noise. An individual standard deviation σ_{train} is drawn from a value range σ_{range} for each augmented sample. Using $X < 100\%$ during experiments aims at retaining clear data performance while achieving higher robustness. The loss is not modified.

D. Gaussian Noise Generation

We investigate two types of Gaussian noise (see Fig. 2), which we denote and define as follows:

1) *Channel-diverse Gaussian noise (G-div):* Each color channel of the same pixel gets an own noise value.
2) *Channel-identical Gaussian noise (G-id):* Each color channel of the same pixel gets the same noise value, which constitutes a special case of G-div.

We also use them for additional investigations on the impact of slight noise type changes between training and inference. This supplements existing research showing weak performance for large changes [10], [36]. To generate G-div with standard deviation σ for a 2D color image x of size $H \times W \times 3$ and values ranging from 0.0 to 1.0, we sample Gaussian noise g_{c3} of size $H \times W \times 3$ by drawing each value from the normal distribution $N(0, \sigma^2)$ individually. Afterwards, the noisy image x' is created by adding g_{c3} to the original image and clipping the values to the valid value range $[0.0, 1.0]$:

$$x' = clip(x + g_{c3}) \tag{1}$$

For G-id we follow almost the same procedure, except that we sample Gaussian noise of size $H \times W \times 1$ and subsequently repeat it 3 times to obtain noise of size $H \times W \times 3$.

V. Experiments

A. Evaluation Dataset and Metric

We utilize the MPII Human Pose Dataset [5] together with its associated PCKh score (percentage of correct detections) for training and evaluation. Using PCKh, a keypoint detection kp_{det} is correct, if the l_2-distance to its associated ground truth kp_{gt} normalized by $norm_{head}$ (a normalization value based on the head size) is below a threshold T:

$$\frac{||kp_{det} - kp_{gt}||}{norm_{head}} \leq T \tag{2}$$

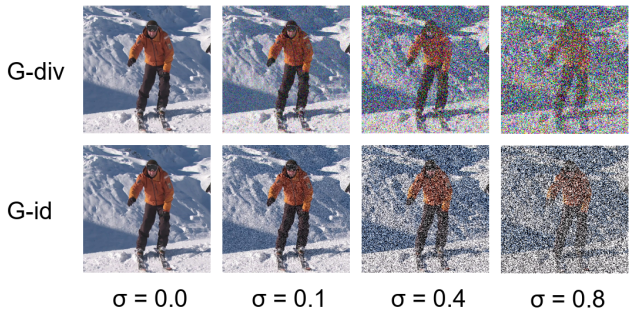


Fig. 2: An image of the MPII Human Pose Dataset [5] augmented with either G-div or G-id of increasing severity.

A typical value for T is 0.5 [5], [15], [20], [21], the PCKh score for $T = 0.5$ is called PCKh@0.5.

Because the test annotations are not publicly available, we use the training/validation split from Newell et al. [15] as training/test split and utilize 1000 samples from the training split as new validation split. Furthermore, we follow the standard procedure of using the annotated center and scale data about people to crop the images. Then, we re-scale them to the size of 256×256 needed for our networks. Pixel values for all three color channels will range from 0.0 to 1.0.

For evaluation under noise, we augment all samples from the test set with either only G-div or only G-id with a fixed σ per test. Test σ values start from 0.0 up to 0.8 in steps of 0.05, resulting in 17 different noise levels for evaluation.

B. Training and Inference Details

FFDNet: For this denoiser, we use a strongly simplified and altered training procedure compared to the original [17] one: We utilize 256×256 images from the MPII Human Pose Dataset [5] with a batch size of 16, augmented with either only G-div or only G-id (depending on experiment), using σ_{train} values randomly drawn from $\sigma_{range} = [0.0, 0.75]$. For each image, we set all values of the corresponding noise level map to the drawn σ_{train} value. As loss we use the mean squared error between the unaugmented image and the denoised output image from FFDNet. As optimizer, we use RMSprop [42] with a learning rate of 0.0001. Training is always performed for 50 epochs. During tests, we also use 256×256 images and investigate the settings $\sigma_{set} = 0.15/0.3/0.7$ for the noise level maps.

BM3D: BM3D does not require training. For denoising, our used version [33] accepts a σ parameter as input to account for the expected variance σ^2 of the noise. Similar to FFDNet, we call this setting σ_{set} and investigate the values $0.15/0.3/0.7$ for it (in practice we use upscaled equivalents for the 0 to 255 range the implementation [41] works on). Also, we apply BM3D to the full images of the MPII Human Pose Dataset [5] before cropping and rescaling.

Hourglass model: Our lightweight hourglass model is trained similar to Newell. et al [15]. We use the same input size of 256 x 256 for color images and output size of 64 x 64 for heatmaps. We employ intermediate supervision and use the mean squared error between predicted and ground truth heatmaps as loss function. The ground truth heatmaps are calculated like Newell et al. [15] did. We augment the data during training using rotation, scaling and flipping. RMSprop [42] is used as optimizer, with learning rate 0.001, gradient clipping and a batch size of 16. For some experiments, we further apply either only G-div or only G-id to 50%/100% of the training images, using randomly drawn σ_{train} values from $\sigma_{range} = [0.0, 0.75]$. For some other experiments, the clean training images are first 'denoised' by either FFDNet or BM3D. Training is always performed for 200 epochs. During tests, we pass each test image through the network to predict a heatmap for each keypoint. The final keypoint locations are calculated by backprojecting the maximum location of each heatmap into the full, uncropped image.

C. Experimental Settings

The following experimental settings are investigated:

- Hourglass model trained on clean images
- Hourglass model trained on 50%/100% noisy images
- FFDNet or BM3D upfront with $\sigma_{set} = 0.15/0.3/0.7$ + hourglass model trained on clean images
- FFDNet or BM3D upfront with $\sigma_{set} = 0.15/0.3/0.7$ + hourglass model trained on denoised clean images

Either only G-div or G-id is used for each training or test. For tests, fixed σ values are used for the noise. Tests with different noise types for training and evaluation are also performed. Each training and each test is performed exactly once. Table I shows all experimental setups with their results.

VI. Results and Discussion

The experimental results from table I show clearly, that Gaussian noise is a severe problem for human pose estimation if not handled: The hourglass model trained on clean data (Nr. 1) has 83.4% correct results on clean test data, while already dropping down to 58.1% for moderate G-div with $\sigma_{test} = 0.15$ and as far below as 10.1% for heavy G-div with $\sigma_{test} = 0.8$. G-id (Nr. 16) has a similar effect. In the following we will assume that G-div is used during training (if necessary) as well as during tests unless stated otherwise.

Circumventing the performance loss under noise without retraining the human pose estimator would be desirable. This could be achieved by deploying an additional denoising module. Therefore we combine the lightweight hourglass model trained on clean data with either FFDNet (Nr. 2-4) or BM3D (Nr. 8-10) for preprocessing input images. Most of the tested combinations outperform the sole hourglass model for σ_{test} values of 0.1 and more, with BM3D outperforming FFDNet. For a comparison, see Fig. 3a. Smaller σ_{set} values for BM3D/FFDNet were best for small σ_{test} values, while higher σ_{set} values are better for large σ_{test} values.

Next, we adjusted the neural network to the effects of image denoising (Nr. 5-7, 11-13) by training the hourglass model from scratch using clean image data, which was first 'denoised' by the respective denoiser also used during tests. In two of our experiments (Nr. 11/12), this even improved the clean data performance slightly (+0.9%/0.4% compared to Nr. 1). In comparison to using the denoisers together with an

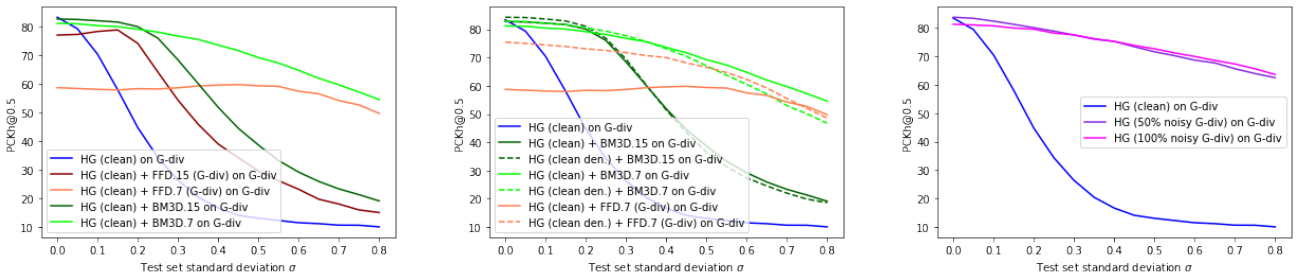
Experimental Setup					PCKh@0.5 scores for noisy test data with different σ_{test} values														
Nr.	Test noise	Train noise	HG train data	Denoiser	$\sigma =$ 0.0	$\sigma =$ 0.05	$\sigma =$ 0.1	$\sigma =$ 0.15	$\sigma =$ 0.2	$\sigma =$ 0.25	$\sigma =$ 0.3	$\sigma =$ 0.35	$\sigma =$ 0.4	$\sigma =$ 0.45	$\sigma =$ 0.5	$\sigma =$ 0.55	$\sigma =$ 0.6	$\sigma =$ 0.7	$\sigma =$ 0.8
1	G-div	-	clean	-	83.4	79.4	70.5	58.1	44.8	34.4	26.4	20.5	16.7	14.2	13.1	12.3	11.5	10.7	10.1
2		G-div	clean	FFD.15	77.1	77.4	78.4	78.9	74.2	64.2	54.4	46.0	39.0	34.2	29.6	26.2	23.2	18.1	15.1
3		G-div	clean	FFD.3	70.7	71.1	71.6	72.3	73.0	74.0	73.0	69.3	62.0	53.4	45.1	39.0	34.1	26.3	21.0
4		G-div	clean	FFD.7	58.7	58.4	58.2	58.0	58.4	58.3	58.7	59.3	59.6	59.8	59.4	59.2	57.5	54.2	49.8
5		G-div	clean-den.	FFD.15	81.7	81.4	81.2	79.1	68.6	56.2	46.4	38.8	32.4	27.9	23.2	20.8	18.4	15.2	12.7
6		G-div	clean-den.	FFD.3	80.2	80.1	80.0	79.2	78.3	76.8	71.9	63.0	50.3	39.1	31.6	26.1	22.4	17.2	14.5
7		G-div	clean-den.	FFD.7	75.5	75.0	74.5	73.9	73.1	72.5	71.8	70.7	70.0	68.1	66.6	64.7	62.3	55.5	48.6
8		-	clean	BM3D.15	82.8	82.6	82.1	81.7	80.1	76.1	68.5	60.2	51.9	44.5	38.7	33.3	29.2	23.4	19.2
9		-	clean	BM3D.3	82.2	82.0	81.5	81.0	80.1	79.2	77.7	76.7	74.3	71.7	68.6	65.8	62.0	52.9	44.1
10		-	clean	BM3D.7	81.2	81.1	80.4	80.1	79.2	78.2	76.8	75.6	73.6	71.7	69.3	67.3	64.8	59.7	54.6
11		-	clean-den.	BM3D.15	**84.3**	**84.2**	**83.6**	**82.9**	81.1	76.8	69.5	60.3	51.5	43.5	36.5	31.4	27.4	22.0	18.6
12		-	clean-den.	BM3D.3	83.8	83.5	83.2	82.7	**81.6**	**80.5**	**79.5**	**77.9**	**75.3**	72.3	68.4	64.6	60.8	52.3	43.9
13		-	clean-den.	BM3D.7	82.9	82.8	82.2	81.6	80.5	79.3	77.7	75.7	73.1	70.4	67.1	63.7	60.5	53.0	46.8
14		G-div	50% noisy	-	83.7	83.3	82.3	81.2	80.0	78.8	77.4	76.0	75.2	73.3	71.6	70.2	68.6	65.6	62.4
15		G-div	100% noisy	-	81.2	81.0	80.6	79.9	79.5	78.1	77.5	76.1	75.2	**73.8**	**72.6**	**71.1**	**69.9**	**67.3**	**63.6**
16	G-id	-	clean	-	83.4	77.2	68.5	60.0	52.1	44.4	37.0	29.9	23.9	19.2	15.3	12.5	10.4	7.3	5.6
17		G-id	clean	FFD.15	76.9	77.4	78.0	78.7	72.9	60.3	49.5	41.7	34.6	29.5	24.8	21.1	18.4	14.0	11.6
18		G-id	clean	FFD.3	72.2	72.7	73.4	73.8	74.5	75.2	75.0	72.8	66.7	57.0	47.7	38.3	31.7	22.1	16.1
19		G-id	clean	FFD.7	63.3	64.1	65.0	66.0	66.9	67.6	67.8	68.8	68.5	68.8	68.4	69.0	68.3	67.5	66.2
20		G-id	clean-den.	FFD.15	81.3	81.3	81.2	79.0	67.1	52.9	42.4	34.3	28.0	22.7	18.9	16.1	14.4	11.9	10.7
21		G-id	clean-den.	FFD.3	80.3	80.3	80.4	80.3	80.2	79.3	77.4	71.8	59.7	45.2	33.7	26.2	21.8	16.7	14.3
22		G-id	clean-den.	FFD.7	78.7	78.7	78.4	78.2	78.0	78.0	77.6	77.0	76.9	76.5	76.2	75.6	75.5	73.8	72.1
23		-	clean	BM3D.15	82.7	82.6	81.8	78.1	69.5	60.7	52.8	45.7	39.3	33.1	28.0	23.9	20.4	15.6	12.5
24		-	clean	BM3D.3	82.2	82.1	81.2	80.5	78.9	76.9	73.5	66.9	58.7	50.2	42.6	35.9	30.7	23.2	18.1
25		-	clean	BM3D.7	81.2	81.0	80.2	79.6	78.2	76.5	75.1	73.1	70.5	68.5	66.0	63.1	60.6	54.3	48.7
26		-	clean-den.	BM3D.15	**84.3**	**84.0**	**83.1**	79.3	72.1	64.4	57.3	49.9	42.5	36.8	31.8	26.5	22.8	17.5	13.8
27		-	clean-den.	BM3D.3	83.8	83.6	82.9	**81.8**	**80.4**	78.0	74.2	68.7	63.1	57.0	52.0	47.4	43.0	35.7	30.0
28		-	clean-den.	BM3D.7	82.9	82.7	81.8	80.8	79.6	77.7	75.9	73.6	71.3	69.4	66.5	63.2	60.7	55.1	48.6
29		G-id	50% noisy	-	83.4	82.6	81.5	80.8	**80.4**	**80.0**	**79.4**	**78.8**	**78.5**	**78.1**	**78.1**	**77.4**	**77.2**	**76.4**	**75.6**
30		G-id	100% noisy	-	80.1	80.1	80.1	79.6	79.4	79.2	78.9	78.5	78.1	77.8	77.2	76.9	76.7	76.2	**75.6**
1	G-div	-	clean	-	83.4	79.4	70.5	58.1	44.8	34.4	26.4	20.5	16.7	14.2	13.1	12.3	11.5	10.7	10.1
31		G-id	clean	FFD.15	76.9	76.6	**73.9**	**67.4**	**55.8**	**44.6**	35.9	28.7	23.8	20.6	17.7	15.8	14.3	12.6	11.8
32		G-id	clean	FFD.3	72.2	72.2	68.4	61.4	53.2	44.3	**37.7**	**31.1**	**25.2**	**20.8**	**18.4**	**16.1**	**14.8**	**12.9**	**11.9**
33		G-id	clean	FFD.7	63.3	61.6	58.2	50.8	43.8	37.2	31.4	26.7	22.7	19.6	17.0	15.7	13.9	12.1	10.9
34		G-id	clean-den.	FFD.15	81.3	79.2	71.8	60.3	45.0	32.2	22.6	16.2	12.5	10.4	8.9	7.9	7.7	7.0	6.7
35		G-id	clean-den.	FFD.3	80.3	77.3	66.8	52.0	36.6	24.9	17.0	12.6	10.1	8.7	7.7	7.4	6.7	6.4	6.0
36		G-id	clean-den.	FFD.7	78.7	75.8	64.8	51.3	38.8	29.0	22.7	18.7	16.2	14.3	12.9	12.0	11.4	10.8	10.4
37		G-id	50% noisy	-	**83.4**	**81.5**	73.4	58.0	39.4	23.2	13.9	9.8	7.6	6.8	6.2	6.2	5.9	5.8	5.4
38		G-id	100% noisy	-	80.1	76.4	64.7	48.2	32.7	22.3	15.8	12.3	10.8	9.8	8.7	8.2	8.0	7.4	6.9
16	G-id	-	clean	-	83.4	77.2	68.5	60.0	52.1	44.4	37.0	29.9	23.9	19.2	15.3	12.5	10.4	7.3	5.6
39		G-div	clean	FFD.15	77.1	77.9	78.7	67.6	55.8	46.1	37.9	30.5	24.6	19.8	16.1	13.7	11.8	8.9	7.4
40		G-div	clean	FFD.3	70.7	71.2	72.2	73.5	71.2	58.7	46.3	38.1	29.7	24.1	19.1	15.5	13.2	9.4	7.5
41		G-div	clean	FFD.7	58.7	58.4	58.1	57.9	58.6	59.2	58.6	55.7	51.2	44.8	37.8	30.2	23.8	13.9	9.2
42		G-div	clean-den.	FFD.15	81.7	81.2	77.3	62.1	51.5	43.5	36.5	30.4	25.1	20.7	18.0	15.5	13.5	10.7	8.8
43		G-div	clean-den.	FFD.3	80.2	80.1	79.3	76.5	65.1	46.1	33.7	26.1	20.7	16.4	13.2	11.3	10.2	9.0	8.1
44		G-div	clean-den.	FFD.7	75.5	75.1	74.3	73.3	71.7	68.7	62.6	51.6	39.1	28.5	21.1	15.5	12.1	9.6	8.6
45		G-div	50% noisy	-	**83.7**	**82.0**	79.3	75.4	71.1	66.3	60.4	55.4	49.9	44.5	38.8	33.3	28.5	20.5	15.3
46		G-div	100% noisy	-	81.2	80.8	**79.5**	**77.5**	**74.3**	**70.3**	**66.1**	**61.3**	**55.9**	**50.6**	**45.1**	**40.0**	**35.2**	**27.4**	**21.5**

TABLE I: All experimental setups and their PCKh@0.5 results for 15 of 17 tested noise levels. Results are grouped in blocks by the test noise type (G-div/G-id). Reference results are displayed in grey (from hourglass model trained on clean data), results using countermeasures against noise are displayed in green (better or same) and red (worse). The first two blocks feature experiments where the type of training noise (if used) matches the test noise, in the last two blocks training and test noise differs. Legend: "HG": hourglass model, "clean-den.": clean data denoised with "Denoiser", "X% noisy": "Train noise" on X% of training data, "FFD.X/BM3D.X": FFDNet/BM3D with 0.X as σ_{set} (FFDNet trained on "Train noise").

hourglass model trained on clean data, the approach leads to improvements for small to medium σ_{test} values and slightly worse results for high σ_{test} values (see Fig. 3b). For FFDNet, this effect started at higher σ_{test} values than for BM3D.

Training the hourglass model directly with noisy data (Nr. 14/15) also led to notable improvements (see Fig. 3c). Using 50% noisy and 50% clean data (Nr. 14) even led to slight improvements on clean test data (+0.3% compared to Nr. 1). In contrast, using 100% noisy data during training (Nr. 15) led to a performance decrease on clean data. This highlights the importance of choosing the right amount of noisy training data to retain clean data performance.

After training and evaluating all approaches using only G-div, we investigated the impact of slight noise type changes between training and evaluation for the training-based approaches. Thus, additional models were trained using G-id and evaluation was performed for all models on G-div as well as G-id. In any case, changing the noise type led to a significant performance decrease. Training on G-div and testing on G-id (Nr. 39-46) was hereby less impacted than



(a) Denoiser w/o training (b) Denoiser with training (c) Trained on noise

Fig. 3: Visualization of results from table I using the same abbreviations. (a) Results of using denoisers in front of a hourglass model trained on clean images. (b) Comparison of results from (a) (solid lines) with results from training the hourglass model on denoised clean images instead (dashed lines). (c) Results of training the hourglass model directly on noisy data.

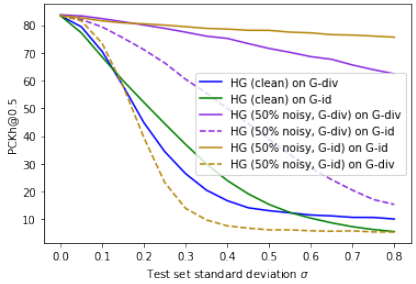


Fig. 4: Impact of changing the noise type used during training at test time (results from table I, same abbreviations.). Dashed lines are used for different train and test noise.

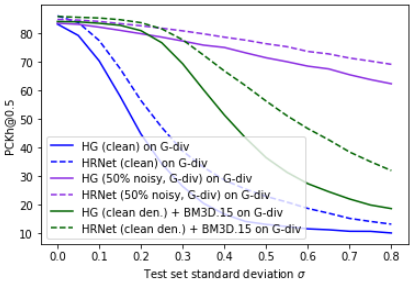


Fig. 5: Comparison of results from new experiments using HRNet (dashed) with experimental results from table I using the hourglass model (solid).

training on G-id and testing on G-div (Nr. 31-38). As an example, take the approach using 50% clean and 50% noisy data during training. This resulted in 75.2%/78.5% correct results when training and testing on G-div/G-id for $\sigma_{test} = 0.4$ (Nr. 14/29). When training on G-div and testing on G-id (Nr. 45), the correct results dropped to 49.9%, while training on G-id and testing on G-div (Nr. 37) performed even worse with 7.6% correct results. See Fig. 4 for comparison. Summarized, generalization in both directions is not good, although G-id is a special case of G-div and thus implicitly included in G-div training. In conclusion, noise types must be very precisely defined for noise-specific countermeasures.

A. Ablation Study: Transferability to other Architectures

To show that our results are not limited to the Hourglass architecture, we repeat experiment Nr. 1, 11 and 14 using the HRNet-W32 architecture [13], which is structurally different. For fair comparison, we train the HRNet similar to the hourglass model, without pretraining. The only alterations are that no intermediate supervision is used and Adam [43] is employed as optimizer, with a learning rate of 0.001 for 210 epochs, which is reduced by a factor of 10 after 170/200 epochs. During tests, the original and flipped image is processed to calculate the final heatmap. Fig. 5 shows the results: HRnet behaves comparable to the hourglass model during all experiments. It is vulnerable to Gaussian noise, and countermeasures have a similar effect.

VII. CONCLUSION

In this paper, we investigated human pose estimation under Gaussian noise and proposed potential countermeasures. Our work is motivated by safety-critical industrial robot applications, which can require proper handling of specific, dangerous noise types, as outlined in section II. We showed the strong, negative impact of Gaussian noise on human pose estimation with experiments: Using a lightweight hourglass model, the amount of correct detections was already cut in half for Gaussian noise with standard deviation $\sigma = 0.2$. To reduce the impact, we evaluated two general strategies: using an additional denoiser (BM3D [16] or FFDNet [17]) as well as training the human pose estimator with noisy data. All approaches were able to drastically improve the robustness of human pose estimation against Gaussian noise. Through an ablation study, we showed the transferability of these results to another neural network architecture. However, none of the approaches was able to fully negate the impact of Gaussian noise for noise with $\sigma > 0.1$. We also investigated the impact of slight noise type alterations between training and evaluation, showing that even such slight changes can have a strong negative impact. This highlights the need for future research to limit the impact of slight noise type changes and to fully negate the impact of Gaussian noise with higher σ. Also, further research is necessary for dealing with other predefined noise types as well as combinations thereof.

References

[1] O. Mazhar, S. Ramdani, B. Navarro, R. Passama, and A. Cherubini, "Towards real-time physical human-robot interaction using skeleton information and hand gestures," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 1–6.

[2] P. Svarny, Z. Straka, and M. Hoffmann, "Toward safe separation distance monitoring from rgb-d sensors in human-robot interaction," *arXiv preprint arXiv:1810.04953*, 2018.

[3] A. Bulat, J. Kossaifi, G. Tzimiropoulos, and M. Pantic, "Toward fast and accurate human pose estimation via soft-gated skip connections," *arXiv preprint arXiv:2002.11098*, 2020.

[4] Y. Bin, X. Cao, X. Chen, Y. Ge, Y. Tai, C. Wang, J. Li, F. Huang, C. Gao, and N. Sang, "Adversarial semantic data augmentation for human pose estimation," *arXiv preprint arXiv:2008.00697*, 2020.

[5] 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.

[6] *Safety of machinery – Safety-related parts of control systems – Part 1: General principles for design*, International Organization for Standardization (ISO) Std. ISO 13 849-1:2008, 2008.

[7] *Robots and robotic devices – Safety requirements for industrial robots – Part 1: Robots*, International Organization for Standardization (ISO) Std. ISO 10 218-1:2011, 2011.

[8] *Safety of machinery - General principles for design - Risk assessment and risk reduction*, International Organization for Standardization (ISO) Std. ISO 12 100:2010, 2010.

[9] 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.

[10] J. Wang, S. Jin, W. Liu, W. Liu, C. Qian, and P. Luo, "When human pose estimation meets robustness: Adversarial algorithms and benchmarks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 11 855–11 864.

[11] B. Cheng, B. Xiao, J. Wang, H. Shi, T. S. Huang, and L. Zhang, "Higherhrnet: Scale-aware representation learning for bottom-up human pose estimation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 5386–5395.

[12] S. Kreiss, L. Bertoni, and A. Alahi, "Pifpaf: Composite fields for human pose estimation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 11 977–11 986.

[13] K. Sun, B. Xiao, D. Liu, and J. Wang, "Deep high-resolution representation learning for human pose estimation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 5693–5703.

[14] B. Xiao, H. Wu, and Y. Wei, "Simple baselines for human pose estimation and tracking," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 466–481.

[15] 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.

[16] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, "Image denoising by sparse 3-d transform-domain collaborative filtering," *IEEE Transactions on image processing*, vol. 16, no. 8, pp. 2080–2095, 2007.

[17] K. Zhang, W. Zuo, and L. Zhang, "Ffdnet: Toward a fast and flexible solution for cnn-based image denoising," *IEEE Transactions on Image Processing*, vol. 27, no. 9, pp. 4608–4622, 2018.

[18] The European Parliament and the Council of the European Union, "Directive 2006/42/ec of the european parliament and of the council," Official Journal of the European Union, L 157, pp. 24-86, 2006.

[19] *Robots and robotic devices - Safety requirements for industrial robots - Part 2: Robot systems and integration*, International Organization for Standardization (ISO) Std. ISO 10 218-2:2011, 2011.

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

[21] L. Ke, M.-C. Chang, H. Qi, and S. Lyu, "Multi-scale structure-aware network for human pose estimation," in *Proceedings of the european conference on computer vision (ECCV)*, 2018, pp. 713–728.

[22] W. Yang, S. Li, W. Ouyang, H. Li, and X. Wang, "Learning feature pyramids for human pose estimation," in *proceedings of the IEEE international conference on computer vision*, 2017, pp. 1281–1290.

[23] W. Li, Z. Wang, B. Yin, Q. Peng, Y. Du, T. Xiao, G. Yu, H. Lu, Y. Wei, and J. Sun, "Rethinking on multi-stage networks for human pose estimation," *arXiv preprint arXiv:1901.00148*, 2019.

[24] G. Papandreou, T. Zhu, N. Kanazawa, A. Toshev, J. Tompson, C. Bregler, and K. Murphy, "Towards accurate multi-person pose estimation in the wild," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 4903–4911.

[25] 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.

[26] C. Li and G. H. Lee, "Generating multiple hypotheses for 3d human pose estimation with mixture density network," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 9887–9895.

[27] F. Moreno-Noguer, "3d human pose estimation from a single image via distance matrix regression," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 2823–2832.

[28] J. Martinez, R. Hossain, J. Romero, and J. J. Little, "A simple yet effective baseline for 3d human pose estimation," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 2640–2649.

[29] J. Xu, Z. Yu, B. Ni, J. Yang, X. Yang, and W. Zhang, "Deep kinematics analysis for monocular 3d human pose estimation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 899–908.

[30] S. Shah, A. Sharma, A. Jain, *et al.*, "On the robustness of human pose estimation," *arXiv preprint arXiv:1908.06401*, 2019.

[31] C. Zheng, W. Wu, T. Yang, S. Zhu, C. Chen, R. Liu, J. Shen, N. Kehtarnavaz, and M. Shah, "Deep learning-based human pose estimation: A survey," *arXiv preprint arXiv:2012.13392*, 2020.

[32] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, "Color image denoising via sparse 3d collaborative filtering with grouping constraint in luminance-chrominance space," in *2007 IEEE International Conference on Image Processing*, vol. 1. IEEE, 2007, pp. I–313.

[33] D. Honzátko and M. Kruliš, "Accelerating block-matching and 3d filtering method for image denoising on gpus," *Journal of Real-Time Image Processing*, vol. 16, no. 6, pp. 2273–2287, 2019.

[34] A. Davy and T. Ehret, "Gpu acceleration of nl-means, bm3d and vbm3d," *Journal of Real-Time Image Processing*, vol. 18, no. 1, pp. 57–74, 2021.

[35] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang, "Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising," *IEEE transactions on image processing*, vol. 26, no. 7, pp. 3142–3155, 2017.

[36] R. Geirhos, C. R. M. Temme, J. Rauber, H. H. Schütt, M. Bethge, and F. A. Wichmann, "Generalisation in humans and deep neural networks," *arXiv preprint arXiv:1808.08750*, 2018.

[37] S. Zheng, Y. Song, T. Leung, and I. Goodfellow, "Improving the robustness of deep neural networks via stability training," in *Proceedings of the ieee conference on computer vision and pattern recognition*, 2016, pp. 4480–4488.

[38] Q. Xie, M.-T. Luong, E. Hovy, and Q. V. Le, "Self-training with noisy student improves imagenet classification," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 10 687–10 698.

[39] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," *arXiv preprint arXiv:1412.6572*, 2014.

[40] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," *arXiv preprint arXiv:1706.06083*, 2017.

[41] D. Honzátko, "Bm3d-gpu," https://github.com/DawyD/bm3d-gpu, accessed: 2021-08-26.

[42] G. Hinton, N. Srivastava, and K. Swersky, "Rmsprop: Divide the gradient by a running average of its recent magnitude," *Neural networks for machine learning, Coursera lecture 6e*, p. 13, 2012.

[43] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.