Uncertainty-Assisted Image-Processing for Human-Robot Close Collaboration*

Seyedomid Sajedi, Wansong Liu, Kareem Eltouny, Sara Behdad, Minghui Zheng, *Member, IEEE*, and Xiao Liang

***Abstract*—The safety of human workers has been the main concern in human-robot close collaboration. Along with rapidly developed artificial intelligence techniques, deep learning models using two-dimensional images have become feasible solutions for human motion detection. These models serve as "sensors" in the closed-loop system that involve humans and robots. Most existing methods that detect human motion using images do not consider the uncertainty from the deep learning model itself. The mappings established by deep learning models should not be taken blindly, and thus uncertainty should be a natural part of this type of sensor. In particular, model uncertainty should be explicitly quantified and incorporated into robot motion control to guarantee safety. With this motivation, to rigorously quantify the uncertainty of these "sensors", this paper proposes a probabilistic interpretation method and automatically provides a framework to benefit from a deep model's uncertainty. Experimental data from human-robot collaboration has been collected and used to validate the proposed method. A training strategy is proposed to efficiently train surrogate models that learn to refine the prediction of the main Bayesian models. The proposed framework is also compared with Ego hands benchmark showing a 4.7% increase in mIoU.**

I. Introduction

Industrial manipulators have been widely employed in manufacturing factories. The inherent merits of manipulators, such as persistence and precision, enable the accomplishments of tasks that are repetitive or require specific handling [1], [2]. Recently, more sophisticated tasks need collaboration between human workers and manipulators along with the increasing interest in flexible manufacturing. Moving manipulators outside of cages to collaborate with humans poses significant challenges to human workers' safety, which needs to be guaranteed as the top priority in human-robot collaboration (HRC).

To enhance the safety of human workers that collaborate with robots, various methods have been developed to enable robots to take reactions immediately once collision is predicted or detected [3]-[5]. Existing workplace safet[1]y mechanisms can be classified into two categories: passive and active ones. Passive protection approaches usually use joint impedance and motion velocity control to prevent collisions (i.e., [6], [7]). The active protection approaches rely on supervisory systems in which human motion is detected and robot motion is planned in real-time to prevent collisions. Therefore, accurately detecting and recognizing human motion is the prerequisite of active safety mechanism and thus the safe HRC.

A considerable amount of studies has been conducted to detect human motion based on wearable or attachable sensors, including sensor-glove [8], surface electromyography [9], triaxial accelerometer [10], and surface markers [11]. Image-based detection using cameras has emerged as an effective alternative to detect human workers' motion due to their low cost, comfortability to human workers, and recently increased computational power. To list a few, Mapari et al. [12] proposed a method to detect hand gestures by using the infrared camera from a leap motion sensor. Zhang et al. [13] proposed a dark channel before segmenting human hands from complex backgrounds based on the variation of color channels. Wang et al. [14] used the skeleton tracking function of the Kinect software development kit to track in-depth hand images and distinguish human hands from the background by setting depth-based thresholds. Dardas et al. [15] used the scale-invariant feature transform to extract key points from images and trained multiple support vector machines to classify human hand pose. Chen et al. [16] applied the hidden Markov model to recognize human hand gestures based on the feature vector extracted from images.

Along with rapidly developed artificial intelligence techniques, deep learning models using two-dimensional images have become feasible solutions recently for human motion detection in HRC [17]. For example, Nuzzi et al. [18] trained a Faster Region Proposal Convolutional neural network to understand operators' commands based on hand gestures. Gao et al. [19] proposed a parallel convolutional neural network model that enables robots' capability of interacting with humans. Rajnathsing and Li [20] developed a network-based safety monitoring system in shared workspace HRC, and relied on networks to determine if the operator exceeds the minimum distance to the planned robot path. Piyathilaka and Kodagoda [21] utilized a dynamic Bayesian network to infer human activities based on three-dimensional skeleton joints. These motion-detection models serve as

Manuscript received: September 9, 2021; Revised: January 20, 2022; Accepted: February 1, 2022. This paper was recommended for publication by Editor Gentiane Venture upon evaluation of the Associate Editor and Reviewers comments. Research supported by National Science Foundation EFMA-1928595, CMMI-2026533 and CMMI-2026276. Corresponding authors: X. Liang and M. Zheng

S. Sajedi, K. Eltouny and X. Liang are with the Civil, Structural and Environmental Engineering Department, University at Buffalo, Buffalo, NY, 14260 USA (e-mail: ssajedi@buffalo.edu, keltouny@buffalo.edu, liangx@buffalo.edu).

W. Liu and M. Zheng are with the Mechanical and Aerospace Engineering Department, University at Buffalo, Buffalo, NY, 14260 USA (e-mail: wansongl@buffalo.edu, mhzheng@buffalo.edu).

S. Behdad is with the Environmental Engineering Sciences Department, University of Florida, Gainesville, FL 32611 USA (email: sarabehdad@ufl.edu)

Digital Object Identifier (DOI): see top of this page.

"sensors" in the closed-loop system that involves humans and robots.

Since these image-based "sensors" play an important role in the safety of human workers, the mappings established by deep learning models should not be taken blindly and a deep neural network model's uncertainty should be a natural part of this type of sensor. Though the output scores (e.g., softmax probabilities) have been considered in the HRC literature to consider the notion of uncertainty [22], [23], it is critical to quantify a deep model's uncertainty and further utilize it for reliable predictions. This is particularly critical when human operators are working with robots in close proximity, such as collaborative assembly and disassembly [24]-[27].

This paper proposes a probabilistic interpretation method and automatically provides a framework to benefit from a deep model's uncertainty. The quantified uncertainty is expected to make human motion detection in HRC more reliable; it can also be incorporated into the motion planning [28] of the robot to prevent collision with human workers. Despite the uncertainty quantification method presented in this paper being applicable to general HRC cases, we build the dataset and validate our method using the collaborative assembly and disassembly scenarios in which uncertainty is particularly critical for human safety.

II. Deep Bayesian Neural Networks

Image processing has significantly benefitted from the enormous progress of artificial intelligence and computer vision in the past decade [29]. Depending on the problem at hand, various types of neural networks have been developed for computer vision (e.g., [30]-[32]), while the search for better algorithms is still in progress [33]. The convolutional neural networks (CNNs) are among the most successful deep learning architectures to obtain valuable information from raw images [34]. Each convolution layer may contain thousands of learnable parameters ($\mathbf{W}$) that characterize the sliding filters. As a result, a deep CNN can have millions of weights corresponding to the convolution layers (e.g. [35]). After calibrating the weights in the training process, the CNN architecture will automatically generate a prediction output ($\mathbf{Y}$) based on raw image data ($\mathbf{X}$).

Despite the ever-growing robustness and accuracy of deep learning algorithms in computer vision, the fact that such autonomous models can make mistakes might raise concerns regarding their reliable industrial applications. The chances of error could be small, but achieving a mistake-free data-driven model may not be possible. In other words, there is a level of uncertainty in the predictions of any statistical model. For a reliable and practical industrial implementation, it is beneficial to measure such model uncertainty.

We will provide a brief description of approximate Bayesian inference in what follows. Further, we elaborate on a method to automatically benefit from the model's uncertainty output without requiring human intervention.

A. Approximate Bayesian Inference

Machine learning research has made several efforts to develop methods that treat a data-driven model and its parameters stochastically [36], [37]. Based on Bayesian probability theory, such approaches provide the mathematical foundation to reason about the uncertainty in predictions of a data-driven model. However, they might be associated with a prohibitive cost of computation when dealing with deep neural networks having a vast number of parameters. Regarding these restrictions, Gal and Ghahramani proposed a novel and efficient method to capture the uncertainty output of deep neural networks [38].

Srivastava et al. [39] initially proposed the standard dropout to alleviate overfitting in neural networks. This operator will randomly set a fraction of units as zero during training. Based on [38], dropout in neural networks is equivalent to the Bayesian approximation of a Gaussian process model [40] over the network weights. While the posterior probability distribution of $\mathbf{W}$ given the data ($p(\mathbf{W}\,|\,\mathbf{X},\mathbf{Y})$) is intractable, it can be estimated with an approximate $q(\mathbf{W})$ using variational inference [41]. This approximate distribution can be learned by minimizing the Kullback-Leibler divergence between the posterior and $q(\mathbf{W})$. During the training process, optimizing the regularized loss function (L) will also encourage learning $q(\mathbf{W})$ [42]. In this paper, we perform semantic segmentation for hand recognition, where each pixel is assigned a label for a binary classification as hands or background. Hence, the loss function is:

$$L=\frac{1}{N_x}\sum_{i=1}^{N_x}\Big[y_i\log p(y_i)+(1-y_i)\log(1-y_i)\Big], \qquad (1)$$

where N_x is the total number of pixel observations in the dataset. y_i is the binary ground truth label (0 or 1), and $p(y_i)$ is the model's output probability for the hands class in the binary cross-entropy function.

The model weights are treated as random variables in the Bayesian approach. Hence, elements in the output tensor will follow certain probability distribution. Given that $q(\mathbf{W})$ is an approximation of $p(\mathbf{W}\,|\,\mathbf{X},\mathbf{Y})$, the predicted labels can be expressed as the following integration [38]:

$$q(\mathbf{y}^*\,|\,\mathbf{x}^*)=\int p(y_i\,|\,\mathbf{x}^*,\mathbf{W})q(\mathbf{W})\mathrm{d}\mathbf{W}\,, \qquad (2)$$

where $\mathbf{x}^*$ and $\mathbf{y}^*$ respectively denote the raw input image and the labels for the observation being evaluated. Using Monte Carlo dropout sampling (MCDS), the mean value of softmax probability of a pixel representing the class hands (S_h) can be expressed as:

$$\mathrm{E}\big[S_h\big]=\frac{1}{T}\sum_{n=1}^{T}S_h^n\,, \qquad (3)$$

where T is the number of samples obtained by random dropout at the evaluation phase.

Furthermore, the entropy of the probability vector (as a measure of epistemic uncertainty [43]) and the standard deviation of S_h samples (SDSS) can serve as indicators of a model's lack of confidence. For this binary problem, the entropy can be expressed as:

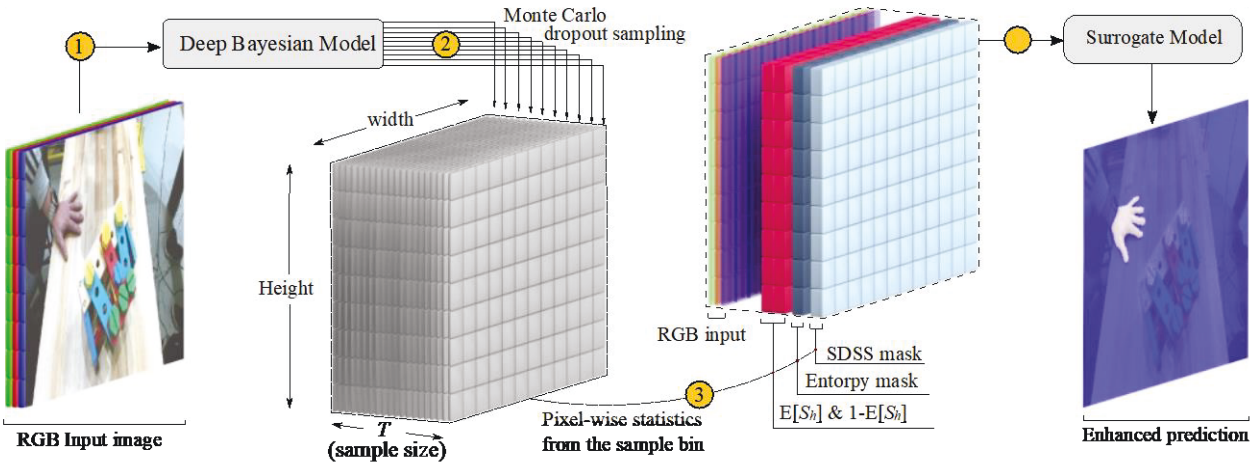

Figure. 1. The proposed refined hand segmentation framework.

$$H = \mathrm{E}\left[S_h\right]\log\frac{1-\mathrm{E}\left[S_h\right]}{\mathrm{E}\left[S_h\right]} - \log(1-\mathrm{E}\left[S_h\right]) \qquad (4).$$

In hand segmentation, both metrics will capture model uncertainty in all pixels. Therefore, the two can be represented as 2D masks for each image.

B. *Interpretation of Uncertainty*

The brief description in section A explains how Bayesian inference quantifies the model uncertainty. The uncertainty masks described earlier can be used by a human agent to monitor an automated system's performance. However, the primary purpose of using AI is to assist with automation. Manual inspection of uncertainty masks for all images contradicts this goal. In addition, such manual inspections can be intellectually challenging. The model tends to show less confidence (i.e., high values in Entropy and SDSS) in predicting visually challenging pixels (e.g., object boundaries) with a higher chance of misclassification. It is noted that the values of H and SDSS depend on both the model hyperparameters and the data itself, and thus, modifying which pixels should be relabeled as hand or background is complex and laborious.

This paper proposes a surrogate model to refine the hand segmentations based on the uncertainty output. To better explain this idea, the process is illustrated in Figure. 1. The deep Bayesian model will take raw RGB images as input (step 1). Next, a bin of T Monte Carlo samples is generated where each sample includes a mask of hand class probability (S_h) for individual pixels (step 2). The bin is then processed to obtain informative statistics, including the expected probability of the two classes, SDSS, and entropy (step 3). In the final step (4), these statistics are stacked with the initial RGB image data and fed to the proposed surrogate model for which another CNN architecture is selected. However, this choice may vary depending on the use case (e.g., sending out a warning signal based on the uncertainty output [44]). It should be noted that work environments could substantially differ. For example, human agents could be wearing gloves or special work attire. In such cases, surrogate models can be fine-tuned on a smaller dataset tailored to each work condition for optimal performance.

The surrogate model is calibrated similarly to the deep Bayesian model using supervised learning. The difference lies in how the training set is utilized for the two. Two possible approaches can be used within this framework. The simple one is to calibrate both models using the same training datasets. In this case, the deep Bayesian model can almost perfectly fit the training set, and pixels with high uncertainty (likely to be misclassified) are rare. Nevertheless, the objective of the surrogate model is to learn from mistakes and correct the initial prediction based on the uncertainty tensor. We observed minor improvement following this strategy by feeding the deep Bayesian model's input to the surrogate model.

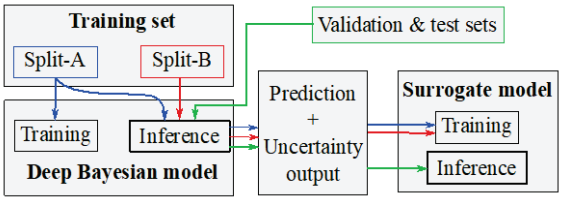

Figure. 2. Training and inference strategies for the deep Bayesian and surrogate models.

In the second approach, the Bayesian model is calibrated on a portion of the training set (split-A), and the rest is held out (split-B). The surrogate model is subsequently calibrated on the complete training set that includes pairs of images and the generated uncertainty output of the Bayesian model from the two splits. The benefit of this strategy is that the surrogate model learns to correct the mistakes from split-B, which is not used in training the original model. The process is shown in Figure.2.

C. *Risk Sensitive Loss function*

The previously explained strategy focuses on providing quality data for training the surrogate model. Nevertheless, this dataset is imbalanced because the deep Bayesian model

correctly classifies most pixels, and the ones with high uncertainty metrics are much less frequent. Moreover, the surrogate input already includes an E[S_h] channel from the Bayesian model, which is accurate for most pixels (e.g., background class). The pixel weights in the surrogate loss function are adjusted such that the model pays more attention to the regions of high uncertainty. The following equation is proposed to determine the weight of each pixel (C_i):

$$C_i = 1 + (SDSS_i^a + H_i) \times \Omega, \quad (5)$$

where $SDSS_i^a$ is the average SDSS of all classes for a pixel, and Ω is a hyperparameter to amplify the emphasis on the uncertainty. In this formulation, the weight of each pixel depends on the SDSS and entropy metrics obtained from the Bayesian model.

III. Deep Learning Architecture

This section elaborates on the deep learning architectures that will serve as the two models shown in Figure. 1. The neural network models developed in this paper are inspired by Fully Convolutional (FC) DenseNets [45]. Similar to many segmentation models, FC-DenseNet comprises a downsampling and an upsampling path where a bottleneck block connects the two. However, the sophistication of the model is due to several interlayer connections. The output feature maps from the previous layers are stacked with the new layers' output feature maps. It is best to break it down into smaller building blocks to understand the algorithm better. Table I lists all the operators required to assemble the deep Bayesian and the surrogate model that we use for hand segmentation.

As illustrated in Figure. 3, concatenation (C) and M_1, the most frequently used modules, construct a dense block (DB). Each layer's input is concatenated with extracted feature maps, where the final output is a tensor of stacked feature maps obtained from each M_1 module. After each M_1 convolution, the number of feature maps (known as growth rate) is 16 in both models' DBs.

TABLE I
Description of Modules in the Deep Learning Architecture

Module	Definition
C	Concatenation
M_0	The first convolution layer
M_1	Stack of batch normalization, *ReLU* activation, 3×3 Convolution, and dropout layers
M_2	Convolution followed by Softmax activation
DB-1	Dense block with 2 stacked layers
DB-2	Dense block with 5 stacked layers (bottleneck)
DB-3	Dense block with 3 stacked layers (bottleneck)
DB-4	Dense block with 4 stacked layers (bottleneck)
TD	Transition down with a stack of batch normalization, *ReLU* activation, 1×1 convolution, dropout layer, and 2×2 max pooling
TU	Transition up with 3×3 transposed convolution

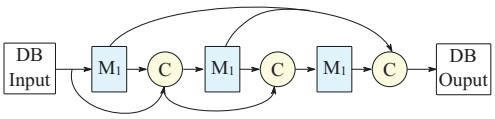

Figure. 3. An example of a dense block with three M_1 modules.

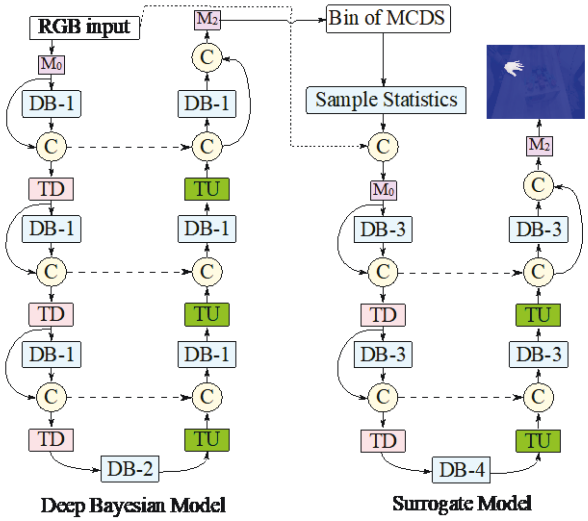

Figure. 4. Deep learning architecture for the Bayesian and surrogate models (dashed arrows represent skip-connections between the up and downsampling paths).

Transition down (TD) modules follow DBs in the downsampling path while a transition up (TU) module is placed before them in the upsampling path. The original architecture has up to 103 layers. The dataset and number of output classes are substantially different in our vision task, and thus we redesign both models, as shown in Figure. 4. The first convolution layer (M_0) will respectively output 16 and 32 channels of feature maps for the deep Bayesian and the surrogate model, and the first model is deeper by having three pairs of TD-TU modules.

Nadam optimizer with a learning rate of 1.0e-3 and schedule decay of 0.004 are selected to train both models. Furthermore, we initially consider 20 Monte Carlo samples (T=20) for each model with a dropout probability of 20%. A sensitivity analysis on the selection of T is discussed later in the paper. We utilize Keras [46] deep learning library on a workstation equipped with a 12 GB NVIDIA Titan GPU. The maximum number of epochs is set to 60, with a batch size of 1 image and early stopping criteria of 10 epochs by monitoring the validation accuracy.

IV. Validation Case Studies

This section provides two validation case studies for the proposed uncertainty-assisted technique. The performance boost using the surrogate models is investigated using two different datasets. The first case study is dedicated to binary hand segmentation and the second one includes 5 classes (4 distinct human hands and background).

A. HRC dataset

The HRC dataset is obtained from various interactions between the human agent and the robot. The platform of the data collection is illustrated in Figure. 5. A UR-5e Co-robot arm was mounted on a table in the Control and Automation Laboratory at the University at Buffalo. Three human agents were involved in the data collection experiment after

providing institutionally-approved informed consent reviewed by the University at Buffalo Institutional Review Board. The first agent was responsible for controlling the Co-robot arm and does not appear in the videos. As described in Table II, each of the other two agents (referred to as O and K) alternatively performed different tasks involving the tools. The other one held the camera and recorded the HRC image data. Agents individually worked on two tasks of disassembling a wooden toy box or a hard disk. The robot arm assisted the human agent by handing over tools, moving objects, untightening bolts, etc. A total of 13 videos were recorded. The agent holding the camera constantly revolved around and changed his distance to the setup table to capture the other agent's interaction with the robot arm. This helps us populate the dataset with variable distances and angles.

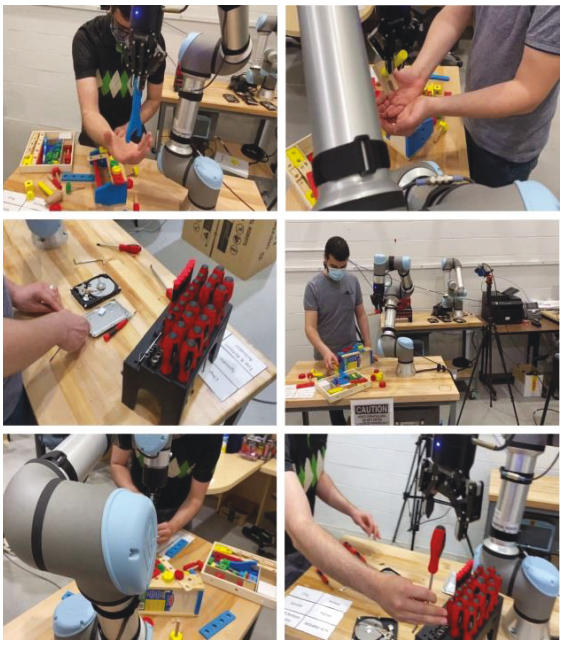

Figure. 5. Data collection in human-robot collaboration experiment

The videos were captured using an iPhone 11 pro with an original resolution of 1920×1080 pixels. 598 static images were extracted by taking one frame per second and later resizing them to 320×180 resolution. Considering the relative distance between the camera and the HRC scene, downsizing images will not result in a significant loss of information but helps boost the inference times. It is generally recommended to limit such relative distance to control the hand/background pixel imbalance. If not possible, the use of higher resolution images might be necessary for exceptional circumstances to minimize the loss of information due to downsizing.

The training, validation, and test split in this case study are obtained in 4 different ways. Training and test sets are obtained by splitting the videos based on the tools, agent, setup, or random video selection to ensure adjacent frames from the same activity are not present in both training and testing sets. For each video in the training set, a 20% validation window with a random location is held out for evaluating the models during training.

TABLE II
HRC DATASET DESCRIPTION

ID	# Frames	Agent	Tool	Activity
1	35	K	Hammer	Wooden box disassembly
2	40	K	Screwdriver	Wooden box disassembly
3	42	K	Wrench	Wooden box disassembly
4	38	O	Screwdriver	Wooden box disassembly
5	40	O	Hammer	Wooden box disassembly
6	38	O	Wrench	Wooden box disassembly
7	36	O	Screwdriver	Wooden box disassembly
8	46	O	End effector	Wooden box disassembly
9	48	O	End effector	Wooden box disassembly
10	90	K	Screwdriver	Wooden box disassembly
11	34	K	End effector	Wooden box disassembly
12	48	K	Screwdriver	Hard disk disassembly
13	63	O	Screwdriver	Hard disk disassembly

TABLE III
SPLIT TYPES FOR TRAINING AND INFERENCE

	Tool	Agent	Setup	Random
Training & validation	Screwdriver Effector	O	Toy	2,3,4,5,6,7, 8,9,10,11,13*
Testing	Hammer Wrench	K	Hard disk	1,12

*Numbers indicate the video ID

TABLE IV
TESTING PERFORMANCE METRICS FOR THE HAND CLASS

Model	IoU			
	Tool	Agent	Setup	Random
Standard dropout	0.709	0.680	0.711	0.648
Bayesian trained on split-A	0.715	0.664	0.639	0.594
Bayesian trained on splits A&B	0.719	0.674	0.670	0.689
Surrogate	**0.744**	**0.718**	**0.731**	**0.713**
	F1-score			
	Tool	Agent	Setup	Random
Standard dropout	0.829	0.809	0.831	0.787
Bayesian trained on split-A	0.833	0.798	0.780	0.746
Bayesian trained on splits A&B	0.837	0.805	0.803	0.816
Surrogate	**0.853**	**0.836**	**0.844**	**0.832**

Training and inference are conducted considering four different models in each split type. The first model is trained on the HRC dataset utilizing the standard dropout and does not perform MCDS at inference time. The second and the third models are Bayesian, where the second one only utilizes half of the training set (Split-A) and the third on the complete training set (Splits A&B). The surrogate model is calibrated on generated uncertainty masks from the complete training set as the output of the second Bayesian model. The validation and test sets are similar for a given split type. A summary of Intersection over Union (IoU) and F1-score for the hand class is given in Table IV. The performance metrics indicate that using a surrogate model yields significant improvements over the other methods using each split type in this case study.

The uncertainty output masks (SDSS and entropy) are also given in Figure. 6. These masks are obtained from the Bayesian model trained on split-A. There is relatively high uncertainty around the object boundaries, as we mentioned earlier. Moreover, the model often has less confidence in physically challenging pixels that might be misclassified. One of these challenging regions is the human forearm, more specifically, the regions around the wrists. Given the relatively uniform color of skin and highly variable view angles, the mentioned areas are often misclassified in the benchmark models. The examples in Figure. 6 indicate that

the surrogate model can learn from the mistakes of the previous model and enhance the overall prediction accuracy.

B. Ego Hands dataset

Ego hands [47] dataset is used as the other benchmark to evaluate the performance of the proposed framework in this paper. We use the portion of this dataset that deals with the segmentation of hands. Unlike the previous case study, since images were captured in an egocentric view, there are distinct labels for the left and right hands of the two players (self and other). We perform three different experiments and compare IoU for the hand classes. These experiments change the ratio between the splits A&B in the training set and also the value of hyperparameter Ω. A summary of the results is presented in Table V.

TABLE V
MULTI-CLASS HAND SEGMENTATION ON EGO HANDS' MAIN TEST SPLIT, IOU

	Self/ Left	Self Right	Other Left	Other right	Avg.
Exp. 1[(a)]	**0.583**	0.555	0.536	0.499	0.524
Exp. 2[(b)]	0.526	0.602	0.612	**0.604**	0.586
Exp. 3[(c)]	0.574	**0.622**	**0.618**	0.599	**0.603**
Bambach et al. [47]	0.515	0.579	0.560	0.569	0.556

[(a)] Surrogate model with Ω=100 and using 2/3 of training dataset for Split A
[(b)] Surrogate model with Ω=200 and using 2/3 of training dataset for Split A
[(c)] Surrogate model with Ω=100 and using 5/6 of training dataset for Split A

It can be observed that the surrogate models in two of the three experiments outperform the segmentation in [47] in terms of average IoU. This study also indicates a few points. First, the value of Ω can significantly affect the accuracy since the benchmark outperforms the surrogate model in the first experiment. The other point is that the overall performance can be affected by changing the ratio of the two splits in the training sets. For example, in the third experiment, the main Bayesian model that is trained on 5/6 of the training set performs better than the one where split-A is comprised of 2/3 of the training set. A sensitivity analysis is recommended to achieve optimal performance for the combination of two models.

V. COMPUTATIONAL COSTS

Compared to the original FC-DenseNet architecture, the proposed architecture, while with significantly reduced computational costs, still yields robust segmentation results. This efficiency is evident by comparing the total number of learnable parameters (elements of **W**) in each network compared with the FC-DenseNet103 in [45] (see Table VI).

TABLE VI
COMPARISON OF ARCHITECTURE SIZES

Architecture	Trainable parameters
FC-DenseNet103[a]	9,319,778
Deep Bayesian model	372,402
Surrogate model	390,082

[a)] The architecture is modified to be consistent with the input and output tensor shapes used in hand segmentation

The inference time of the proposed Bayesian framework is dominated by MCDS rather than the depth of the networks. A sensitivity analysis is conducted on the HRC dataset by changing *T*. The results (Table VII) indicate that the reduction in sample size substantially boosts the inference time while the decay in IoU metrics is negligible. It is noted that the achieved 16 fps is sufficiently fast for the robot to respond and prevent the collision [28].

TABLE VII
HAND CLASS IOU VS. INFERENCE SPEED

	Split				Inference time
T	Tool	Agent	Setup	Random	(fps)
3	0.715	0.709	0.731	0.708	16.1
5	0.742	0.713	0.731	0.711	11.4
10	0.742	0.717	0.732	0.712	6.5
20	0.744	0.718	0.731	0.713	3.5

VI. CONCLUSIONS AND FUTURE WORK

Human motion detection is a critical step to guarantee human workers' safety when collaborating with robots closely. Since data-driven models are not mistake-free, their reliability is of crucial importance. This paper develops deep learning models that use two-dimensional images to detect hands in close collaboration with human-robot hands. Instead of blindly treating these sensors as accurate, this paper proposes a probabilistic interpretation method by inferring distribution over the networks' weights to quantify these sensors' uncertainty rigorously and benefit from this uncertainty to enhance prediction accuracies. Bayesian FC-DenseNets are designed as the deep learning architecture concerning computational efficiency to segment pixels of a human hand. Model uncertainty is then quantified in terms of SDSS and entropy using Monte Carlo dropout sampling. A second surrogate model is then developed to benefit from the uncertainty output automatically and refine the Bayesian model's initial predictions. A novel training strategy is proposed to improve the learning capability of the surrogate model in refining the predictions from the Bayesian model.

This paper presents two case studies to validate the benefits of using surrogate models. The first case study involves gathering experimental HRC data and validating the proposed method in different activities. Moreover, the proposed surrogate models are evaluated on the benchmark Ego hands dataset to highlight its superiority. The presented method can be powerful tools to enhance the reliability of vision-based sensors for HRC tasks, and thus has the potential to reduce the need for wearable sensors which have been heavily relied on in human motion monitoring.

Future studies can focus on integrating the proposed Bayesian framework for tasks involving uncertain human behavior such as physiology or body motion detections using videos and recurrent-based deep learning architectures. This framework can also benefit robotic motion planning to guarantee collision-free collaboration between humans and robots by providing quantified uncertainties of human motion prediction. While surrogate models improve overall performance, the chances of mistakes are not eliminated. Another area for future research is to design surrogate models that generate warning signals for human intervention based on the uncertainty output. Finally, additional studies such as training with quantization and pruning networks are necessary

to improve the inference time of the Bayesian framework where real-time inference is required.

ACKNOWLEDGMENT

The authors would like to thank NVIDIA for donating a Titan V GPU that was used to train the deep learning models in this paper.

REFERENCES

[1] Iqbal, J., Islam, R. U., Abbas, S. Z., Khan, A. A., & Ajwad, S. A. (2016). Automating industrial tasks through mechatronic systems–A review of robotics in industrial perspective. Tehnički vjesnik, 23(3), 917-924.

[2] Roveda, L., Pedrocchi, N., Beschi, M., & Tosatti, L. M. (2018). High-accuracy robotized industrial assembly task control schema with force overshoots avoidance. Control Engineering Practice, 71, 142-153.

[3] Haddadin, S., Albu-Schaffer, A., De Luca, A., & Hirzinger, G. (2008, September). Collision detection and reaction: A contribution to safe physical human-robot interaction. In 2008 IEEE/RSJ International Conference on Intelligent Robots and Systems (pp. 3356-3363). IEEE.

[4] De Luca, A., Albu-Schaffer, A., Haddadin, S., & Hirzinger, G. (2006, October). Collision detection and safe reaction with the DLR-III lightweight manipulator arm. In 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems (pp. 1623-1630). IEEE.

[5] Morinaga, S., & Kosuge, K. (2003, September). Collision detection system for manipulator based on adaptive impedance control law. In 2003 IEEE International Conference on Robotics and Automation (Cat. No. 03CH37422) (Vol. 1, pp. 1080-1085). IEEE.

[6] Bicchi, A., & Tonietti, G. (2004). Dealing with the safety-performance tradeoff in robot arms design and control. IEEE Robotics and Automation Magazine, 11(2).

[7] Tonietti, G., Schiavi, R., & Bicchi, A. (2005, April). Design and control of a variable stiffness actuator for safe and fast physical human/robot interaction. In Proceedings of the 2005 IEEE international conference on robotics and automation (pp. 526-531). IEEE.

[8] Luzanin, O., & Plancak, M. (2014). Hand gesture recognition using low-budget data glove and cluster-trained probabilistic neural network. Assembly Automation.

[9] Hu, Y., Li, Z., Li, G., Yuan, P., Yang, C., & Song, R. (2016). Development of sensory-motor fusion-based manipulation and grasping control for a robotic hand-eye system. IEEE Transactions on Systems, Man, and Cybernetics: Systems, 47(7), 1169-1180.

[10] Carfi, A., Motolese, C., Bruno, B., & Mastrogiovanni, F. (2018, August). Online human gesture recognition using recurrent neural networks and wearable sensors. In 2018 27th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN) (pp. 188-195). IEEE.

[11] Metcalf, C. D., Notley, S. V., Chappell, P. H., Burridge, J. H., & Yule, V. T. (2008). Validation and application of a computational model for wrist and hand movements using surface markers. IEEE Transactions on Biomedical Engineering, 55(3), 1199-1210.

[12] Mapari, R. B., & Kharat, G. (2015, November). Real time human pose recognition using leap motion sensor. In 2015 IEEE International Conference on Research in Computational Intelligence and Communication Networks (ICRCICN) (pp. 323-328). IEEE.

[13] Zhang, Q., Yang, M., Zheng, Q., & Zhang, X. (2017, October). Segmentation of hand gesture based on dark channel prior in projector-camera system. In 2017 IEEE/CIC International Conference on Communications in China (ICCC) (pp. 1-6). IEEE.

[14] Wang, Y., & Yang, R. (2013, July). Real-time hand posture recognition based on hand dominant line using kinect. In 2013

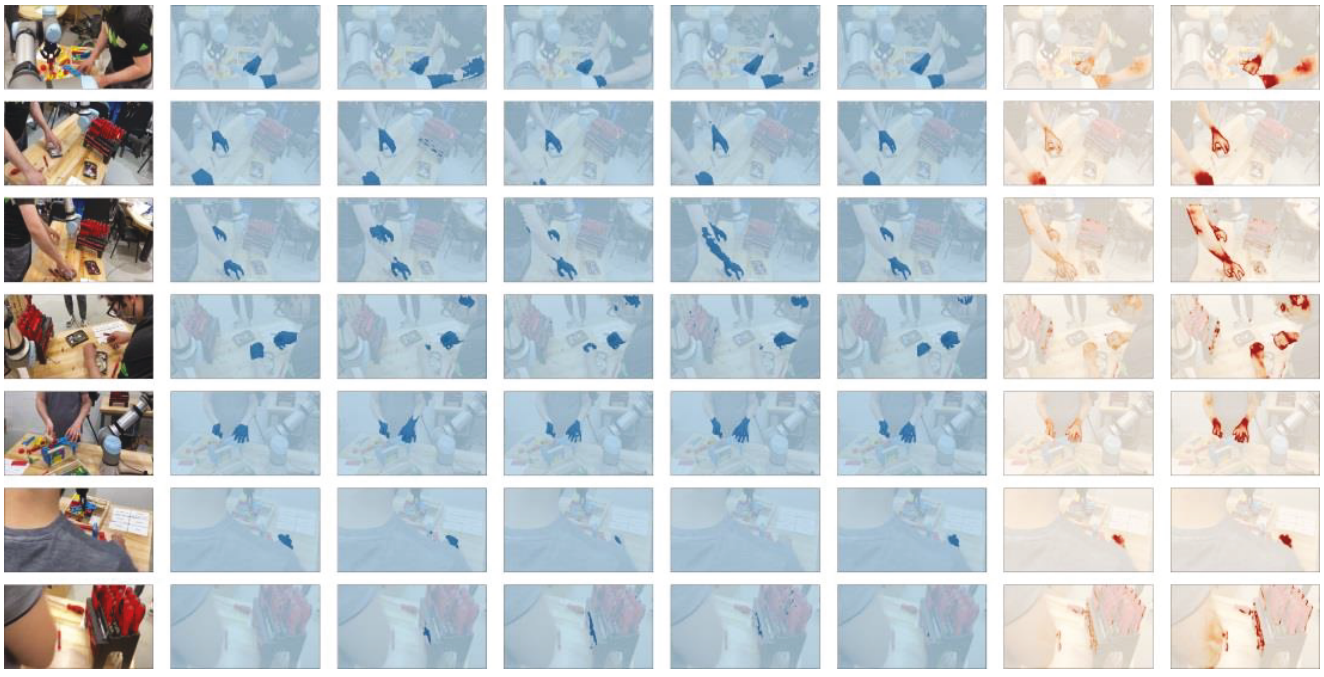

Figure. 6. Sample hand segmentation test results (SDSS and entropy are from the Bayesian model trained on split A)

IEEE International Conference on Multimedia and Expo Workshops (ICMEW) (pp. 1-4). IEEE.

[15] Dardas, N. H., & Georganas, N. D. (2011). Real-time hand gesture detection and recognition using bag-of-features and support vector machine techniques. IEEE Transactions on Instrumentation and measurement, 60(11), 3592-3607.

[16] Chen, F. S., Fu, C. M., & Huang, C. L. (2003). Hand gesture recognition using a real-time tracking method and hidden Markov models. Image and vision computing, 21(8), 745-758.

[17] Ji, Y., Yang, Y., Shen, F., Shen, H. T., & Li, X. (2019). A survey of human action analysis in HRI applications. IEEE Transactions on Circuits and Systems for Video Technology, 30(7), 2114-2128.

[18] Nuzzi, C., Pasinetti, S., Lancini, M., Docchio, F., & Sansoni, G. (2019). Deep learning-based hand gesture recognition for collaborative robots. IEEE Instrumentation & Measurement Magazine, 22(2), 44-51.

[19] Gao, Q., Liu, J., Ju, Z., Li, Y., Zhang, T., & Zhang, L. (2017, August). Static hand gesture recognition with parallel CNNs for space human-robot interaction. In International Conference on Intelligent Robotics and Applications (pp. 462-473). Springer, Cham.

[20] Rajnathsing, H., & Li, C. (2018). A neural network based monitoring system for safety in shared work-space human-robot collaboration. Industrial Robot: An International Journal..

[21] Piyathilaka, L., & Kodagoda, S. (2015). Human activity recognition for domestic robots. In Field and Service Robotics (pp. 395-408). Springer, Cham.

[22] Abu Farha, Y., & Gall, J. (2019). Uncertainty-aware anticipation of activities. In Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops (pp. 0-0).

[23] Furnari, A., Battiato, S., & Maria Farinella, G. (2018). Leveraging uncertainty to rethink loss functions and evaluation measures for egocentric action anticipation. In Proceedings of the European Conference on Computer Vision (ECCV) Workshops (pp. 0-0).

[24] Liu, L., Liu, Y., & Zhang, J. (2018). Learning-based hand motion capture and understanding in assembly process. IEEE Transactions on Industrial Electronics, 66(12), 9703-9712.

[25] Lim, G. H., Pedrosa, E., Amaral, F., Lau, N., Pereira, A., Dias, P., ... & Reis, L. P. (2017, April). Rich and robust human-robot interaction on gesture recognition for assembly tasks. In 2017 IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC) (pp. 159-164). IEEE.

[26] Lee, M., Behdad, S., Liang, X., & Zheng, M. (2022). Task allocation and planning for product disassembly with human-robot collaboration. *Robotics and Computer-Integrated Manufacturing,* 76, 102306.

[27] Lee, M. L., Behdad, S., Liang, X., & Zheng, M. (2020, July). Disassembly sequence planning considering human-robot collaboration. In *2020 American Control Conference* (ACC) (pp. 2438-2443). IEEE.

[28] Wei, K., & Ren, B. (2018). A method on dynamic path planning for robotic manipulator autonomous obstacle avoidance based on an improved RRT algorithm. Sensors, 18(2), 571.

[29] LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. nature, 521(7553), 436-444.

[30] Xingjian, S. H. I., Chen, Z., Wang, H., Yeung, D. Y., Wong, W. K., & Woo, W. C. (2015). Convolutional LSTM network: A machine learning approach for precipitation nowcasting. In Advances in neural information processing systems (pp. 802-810).

[31] Ren, S., He, K., Girshick, R., & Sun, J. (2016). Faster R-CNN: towards real-time object detection with region proposal networks. IEEE transactions on pattern analysis and machine intelligence, 39(6), 1137-1149.

[32] Brock, A., Donahue, J., & Simonyan, K. (2018). Large scale GAN training for high fidelity natural image synthesis. arXiv preprint arXiv:1809.11096.

[33] Chen, X., Girshick, R., He, K., & Dollár, P. (2019). Tensormask: A foundation for dense object segmentation. In Proceedings of the IEEE/CVF International Conference on Computer Vision (pp. 2061-2069).

[34] Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. Advances in neural information processing systems, 25, 1097-1105.

[35] Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556.

[36] Koller, D., & Friedman, N. (2009). Probabilistic graphical models: principles and techniques. MIT press.

[37] Neal, R. M. (2012). Bayesian learning for neural networks (Vol. 118). Springer Science & Business Media.

[38] Gal, Y., & Ghahramani, Z. (2016, June). Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In international conference on machine learning (pp. 1050-1059). PMLR.

[39] Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. The journal of machine learning research, 15(1), 1929-1958.

[40] Rasmussen, C. E. (2003, February). Gaussian processes in machine learning. In Summer school on machine learning (pp. 63-71). Springer, Berlin, Heidelberg.

[41] Graves, A. (2011). Practical variational inference for neural networks. Advances in neural information processing systems, 24.

[42] Gal, Y., & Ghahramani, Z. (2015). Bayesian convolutional neural networks with Bernoulli approximate variational inference. arXiv preprint arXiv:1506.02158.

[43] Kendall, A., & Gal, Y. (2017). What uncertainties do we need in bayesian deep learning for computer vision?. arXiv preprint arXiv:1703.04977.

[44] Sajedi, S. O., & Liang, X. (2021). Uncertainty-assisted deep vision structural health monitoring. Computer-Aided Civil and Infrastructure Engineering, 36(2), 126-142.

[45] Jégou, S., Drozdzal, M., Vazquez, D., Romero, A., & Bengio, Y. (2017). The one hundred layers tiramisu: Fully convolutional densenets for semantic segmentation. In Proceedings of the IEEE conference on computer vision and pattern recognition workshops (pp. 11-19).

[46] Chollet, F. (2015). Keras.

[47] Bambach, S., Lee, S., Crandall, D. J., & Yu, C. (2015). Lending a hand: Detecting hands and recognizing activities in complex egocentric interactions. In Proceedings of the IEEE International Conference on Computer Vision (pp. 1949-1957).