

Continuous Wavelet Network for Efficient and Transferable Collision Detection in Collaborative Robots

Zhenwei Niu^{ID}, Taimur Hassan^{ID}, Senior Member, IEEE, Mohamed Nassim Boushaki, Naoufel Werghi^{ID}, Senior Member, IEEE, and Irfan Hussain^{ID}, Member, IEEE

Abstract—This article addresses the crucial aspect of safety in collaborative robotics by introducing a new continuous wavelet transform-convolutional neural network (CWT-CNN) for efficient robot collision detection. Unlike conventional methods, CWT-CNN exhibits superior data efficiency, requiring minimal collision data for robust training without relying on a dynamic model. The network's adaptability extends to varying internal stiffness levels, offering robustness to changes in robotic system characteristics. Through comprehensive experimental studies, we investigate the impact of input signal types, wavelet types, wavelet scale ranges, and time-moving window sizes on collision detection performance, offering critical insights for optimal CWT parameter selection. Additionally, our transferability analysis demonstrates that the CWT-CNN can seamlessly adapt from one joint to another, requiring only minimal free-motion data from the new joint. This adaptability is validated through extensive experiments on an industrial robot and the robot equipped with variable stiffness actuators. In conclusion, the CWT-CNN is highly generalizable and data-efficient, making it a reliable solution for real-time collision detection in human-robot interactions, addressing a key aspect of safety in collaborative environments.

Index Terms—Collaborative robot with variable stiffness actuators (VSAs), collision detection, continuous wavelet transform (CWT), deep learning.

I. INTRODUCTION

TO BOOST industrial efficiency and better-assist people in our daily lives, there has been a significant increase in physical collaboration between humans and robots [1]. Safety is prioritized in collaborative work to allow for more broad

Received 7 September 2024; accepted 1 December 2024. This work was supported by the Khalifa University of Science and Technology under Award RC1-2018-KUCARS. This article was recommended by Associate Editor N. Sun. (*Corresponding author: Irfan Hussain*.)

Zhenwei Niu and Mohamed Nassim Boushaki are with the Khalifa University Center for Robotics and Autonomous Systems, Khalifa University, Abu Dhabi, UAE (e-mail: zhenwei.niu@ku.ac.ae; boushakmn@yahoo.fr).

Taimur Hassan is with the Department of Electrical, Computer, and Biomedical Engineering, Abu Dhabi University, Abu Dhabi, UAE (e-mail: taimur.hassan@adu.ac.ae).

Naoufel Werghi is with the Electrical Engineering and Computer Science Department and the Khalifa University Center for Robotics and Autonomous Systems, Khalifa University, Abu Dhabi, UAE (e-mail: naoufel.werghi@ku.ac.ae).

Irfan Hussain is with the Mechanical Engineering Department and the Khalifa University Center for Robotics and Autonomous Systems, Khalifa University, Abu Dhabi, UAE (e-mail: irfan.hussain@ku.ac.ae).

Color versions of one or more figures in this article are available at <https://doi.org/10.1109/TSMC.2024.3518700>.

Digital Object Identifier 10.1109/TSMC.2024.3518700

and successful human-robot cooperation [2], [3]. It is vital to avoid collisions or respond appropriately during physical human-robot interaction (pHRI).

Unexpected collisions can be avoided by applying external sensors [4], [5], [6], [7]. However, strict collision prevention may not be sufficient or practical to ensure safety, especially considering that the relative motions between the robot and the human might be rapid or hardly foreseeable. Therefore, in addition to collision avoidance strategies, rapid and accurate collision detection and reaction are required to ensure safety. Haddadin et al. [8] proposed the collision event pipeline to reduce the risk of human injury from physical contact. As an initial stage, rapid and accurate collision detection is important in this pipeline. Currently, collision detection techniques can be categorized into three main groups: 1) artificial skin-based; 2) model-based; and 3) data-driven methods.

Artificial skins, made up of deformable material and tactile sensors, can precisely detect and localize collision events [9], [10], [11], [12]. However, manufacturing and covering the entire robot with these skins are difficult and expensive. On the other hand, model-based methods utilize the identified dynamic model to estimate external torque on robots and compare it with a user-defined threshold to detect collisions. In particular, the generalized momentum observer (MOB) method has been widely studied [8], [13], [14], [15]. However, MOB and other model-based approaches encounter challenges in accurately identifying the robot dynamic model, given nonlinearities like friction and unmodeled effects, such as backlash and elasticity [16], [17]. Tuning the user-defined threshold also demands significant effort.

Alternatively, learning-based collision detection methods address challenges in model-based approaches by training neural network models with collision and/or collision-free data from repeated trials. This enables the network to account for uncertainty and unmodeled effects in the dynamic model, eliminating the need for a user-defined threshold [16]. An end-to-end approach, trained with residuals and collected signals, such as joint position and velocity, can directly produce a binary classification [16], [17], [18], [19], [20].

Learning-based methods, widely utilized in commercial robots with rigid joints, have not been studied for collaborative robots featuring variable stiffness actuation. These robots, with potential benefits for close human-robot collaboration, pose a challenge for accurate and robust collision detection.

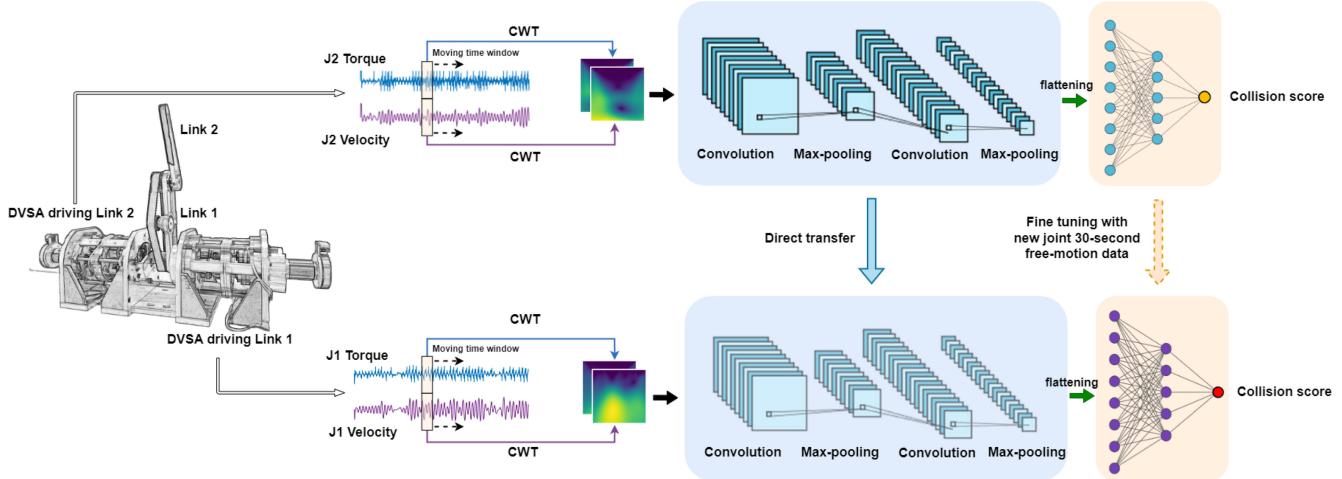


Fig. 1. Overall structure of CWT-CNN for robot collision detection. The proprioceptive signals are first transformed into time–frequency representations via CWT with a time-moving window. These representations contain informative patterns, and they are stacked together for the following collision event classification. CWT-CNN requires minimum collision data for training. It has the capability to generalize to random motion, collision location, and various intrinsic joint stiffness. The trained CWT-CNN can be transferred to a new joint by fine-tuning its last dense layers using only 30 s of free-motion data from that joint.

The varying internal characteristics of robots with variable stiffness make comprehensive data collection challenging for learning-based approaches. Training a network resilient to stiffness variations is preferable, allowing it to excel under a singular stiffness condition while maintaining performance across various stiffness levels.

Addressing the challenges of data collection risks and imbalanced datasets in robot collision detection is crucial. A data-efficient solution is necessary, especially considering the difficulty and potential risks involved in gathering sufficient collision data for both humans and robots. Kim et al. [17] introduced a versatile modularized neural network (MNN) with dynamic and bias input features for collision detection, demonstrating the ability to handle imbalanced datasets and offering greater data efficiency compared to other learning-based methods. However, training MNN still requires at least 1 h of collision data and 2 h of free-motion data.

To address the aforementioned challenges more effectively, this article focuses on enhancing sample efficiency and transferability in collision detection for robots with variable stiffness actuation. Unlike traditional learning-based methods that rely on time-dependent signals and require substantial time-series data, an alternative approach is proposed to optimize the analysis of time-domain data by using signal processing methods to extract more information even with a smaller dataset [21]. Wavelet analysis, a well-established time–frequency scheme, is particularly suitable for analyzing nonstationary signals, effectively capturing both spectral and temporal features. It has been effectively used for a wide range of applications, including but not limited to gait events identification [22], biomedical signals identification [23], [24], and fault diagnosis [25], [26], [27]. In the realm of wavelet transform families, both discrete wavelet transform (DWT) and continuous wavelet transform (CWT) are applicable for time–frequency analysis. DWT excels in decomposing signals into nonoverlapping frequency components but lacks high-scale/frequency resolution. On the contrary, CWT offers high

resolution on both scale and frequency, crucial for accurate time–frequency analysis [21], [28].

This article introduces the CWT-convolutional neural network (CWT-CNN) for collision detection in robots with variable stiffness actuation. In the proposed approach, proprioceptive signals are transformed into time–frequency representations in real time using CWT, capturing both spectral and temporal information crucial for collision identification (see Fig. 1). Subsequently, convolutional neural network (CNN) is employed to extract local and global features from these two-dimensional (2-D) representations for classification. CWT-CNN demonstrates superior sample efficiency, requiring only minimum collision data for training, making it well-suited for addressing challenges associated with imbalanced and challenging-to-collect collision conditions.

CWT-CNN not only handles imbalanced training data effectively but also demonstrates strong generalization across diverse scenarios and stiffness conditions. This versatility makes CWT-CNN a valuable tool for accurate collision detection in different applications. To enhance the generalizability of CWT-CNN and minimize the need to collect collision data across different links, we investigate the application of transfer learning (TL) techniques. The concept of TL entails the utilization of a pretrained model as an initial framework for a new task, followed by fine-tuning the model using a limited amount of new data. In our case, collision experiments are conducted exclusively on a particular link, and the CWT-CNN model is pretrained using the corresponding joint data. By fine-tuning the model with a minimal amount of free-motion data from a new joint, the pretrained network can effectively adapt to the new joint and manage varying joint stiffness conditions.

A. Main Contribution

In summary, we have made the following contributions.

- 1) We introduce a highly sample-efficient end-to-end collision detection method for robots equipped with

variable stiffness actuation, named CWT-CNN. CWT-CNN only requires minimum collision data for training. Compared with conventional learning-based techniques, CWT-CNN performs better with the same limited and inherently imbalanced training collision data.

- 2) We extensively investigate the impact of various factors on collision detection performance, including input signal types, mother wavelet types, wavelet scale range, and time-moving window size. The results are validated using quantitative experiments. To the best of our knowledge, this study represents the first work to investigate the application of time-frequency analysis for robot collision detection. Furthermore, our experimental results offer valuable practical insights into selecting optimal CWT parameters to improve collision detection performance.
- 3) CWT-CNN exhibits robust generalization capabilities, making it suitable for scenarios involving random motion and varying collision locations. Moreover, it can also handle conditions with various intrinsic joint stiffness values. This study represents the first exploration of a learning-based approach for a manipulator equipped with variable stiffness actuators (VSAs).
- 4) We introduce a TL technique that facilitates the seamless transfer of a pretrained CWT-CNN model from one joint to another. The TL process requires only a small amount of free-motion data from the new joint, thereby eliminating the need to collect collision data for the new link. We experimentally validate this TL approach on both industrial robots and robots equipped with VSAs, demonstrating the strong generalization capability of the proposed method.

B. Organization of Article

The remainder of this article is organized as follows. Section II introduces the platform used in this study. Section III outlines the CWT-CNN for robot collision detection. It begins with an overview of CWT and a discussion of choosing input variables. Subsequently, it provides an introduction to the overall network structure, the data collection method, and the network training process. In Section IV, we present the experimental findings concerning the effects of input variable types, mother wavelet types, wavelet scale ranges, and time-moving window size on collision detection. Furthermore, our results include a comparative analysis with state-of-the-art methods, demonstrating the superior data efficiency and robustness of CWT-CNN. In Section V, TL is investigated to extend CWT-CNN to different joints. Section VI presents a discussion on the experimental comparison of various time-frequency analysis methods for robot collision detection. Additionally, this section extends the TL approach on an industrial robot via extensive experiments. Section VII concludes this article.

II. PLATFORM INTRODUCTION

This study uses a robot platform equipped with discrete VSAs (DVSAs). The overall configuration of the platform

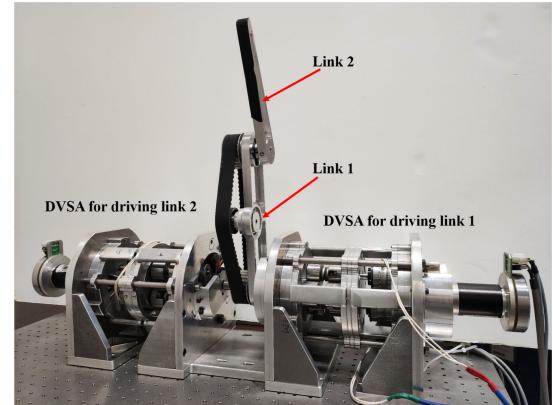
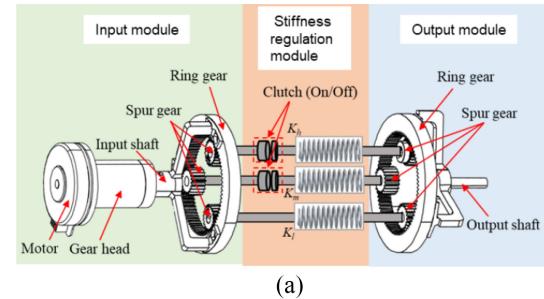


Fig. 2. Experiments platform: A robotic manipulator with discrete VSAs (DVSAs). It has four levels of intrinsic physical stiffness and the stiffness can be rapidly adjusted online by controlling the engagement of the springs.



(a)

Spring with high stiffness K_h	Spring with medium stiffness K_m	Stiffness level	Equivalent stiffness (N.m/rad)
0	0	1	6
0	1	2	50
1	0	3	160
1	1	4	246

(b)

Fig. 3. (a) Working principle of DVSA. (b) Stiffness levels of DVSA. It can adjust the stiffness by manipulating the number of active springs through the engagement or disengagement of the clutches.

is depicted in Fig. 2, while Fig. 3 illustrates the working principle of the DVSA. Each robotic joint is equipped with a system that offers four distinct levels of physical stiffness. By selectively engaging or disengaging specially designed springs, the stiffness of each joint can be rapidly and precisely adjusted to any predefined level. Fig. 3(b) presents a comprehensive overview of the specific values associated with joint stiffness levels. In our previous work [29], a novel collision handling pipeline incorporating DVSA was introduced to effectively mitigate the impact of unexpected collisions and enhance safety during human-robot interaction. Notably, one of the important reaction strategies involves rapidly switching the physical stiffness of the joint from its current level to the lowest level [stiffness level 1, as shown in Fig. 3(b)]. Experimental validation was conducted to evaluate the effectiveness of this strategy in enhancing safety. This study focuses on collision detection for collaborative robots equipped with variable stiffness actuation, and our model is proposed based on this

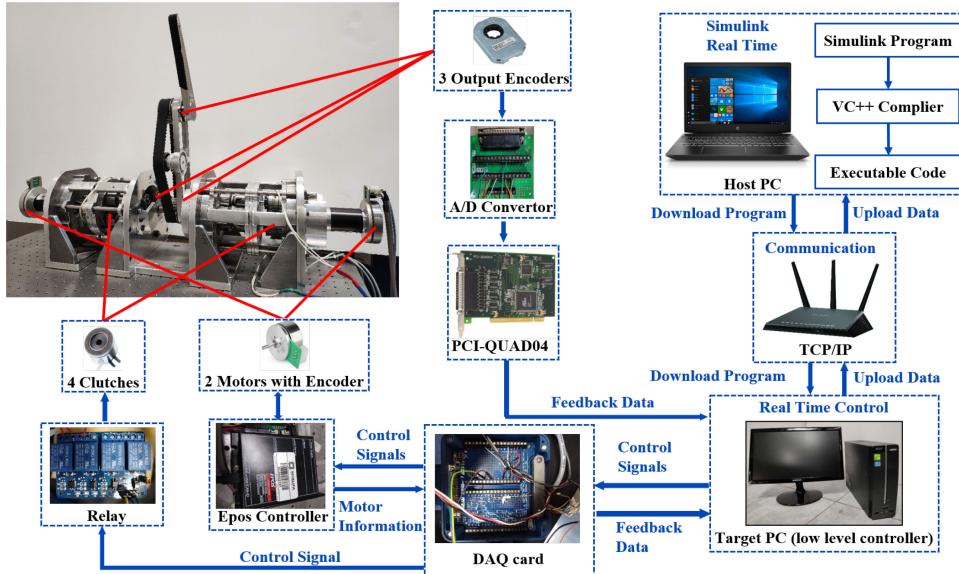


Fig. 4. Complete platform interface, including the compliant manipulator, the high-level and low-level control systems, the drivers, relays, encoders, and the communication system.

platform to examine the effectiveness of collision detection in this context.

The experimental setup, which is shown in Fig. 4, consists of several key components: the host computer, the target computer, a TCP/IP communication link, a PCI 6221M data acquisition (DAQ) device, EPOS 24/5 motor drivers, relays for clutch control, encoders, and an analog-to-digital (A/D) converter. High-level control tasks are performed in Simulink on the host computer, while real-time low-level control is handled on the target computer using the MATLAB xPC target framework. During the experiments, a proportional-integral-derivative (PID) controller is utilized. Gain scheduling is implemented to ensure precise position tracking under varying stiffness conditions. The two computers communicate via a 100 Mb/s TCP/IP connection. Initially, executable code is downloaded from the host to the target computer. The DAQ card then converts the command signals and sends them to the EPOS 24/5 drivers to control the motors. Additionally, the DAQ card coordinates the engagement and disengagement of the clutch relays. Feedback data from encoders and motors is sent back to the host computer through the DAQ system, allowing the host computer to monitor operations in real time.

III. ROBOT COLLISION DETECTION USING CWT-CNN

A. Background of Continuous Wavelet Transform

The wavelet transform is a mathematical tool that can be used to transform a signal from the time domain into a new form consisting of a sequence of wavelet coefficients. Unlike the Fourier Transform, which uses a series of sine waves with various frequencies for signal analysis, the wavelet transform uses a series of functions called wavelets, i.e., the small waves, to analyze a signal. These small waves are zero means and limited (finite) in time. It can specify which spectral components are present at any particular moment from the raw

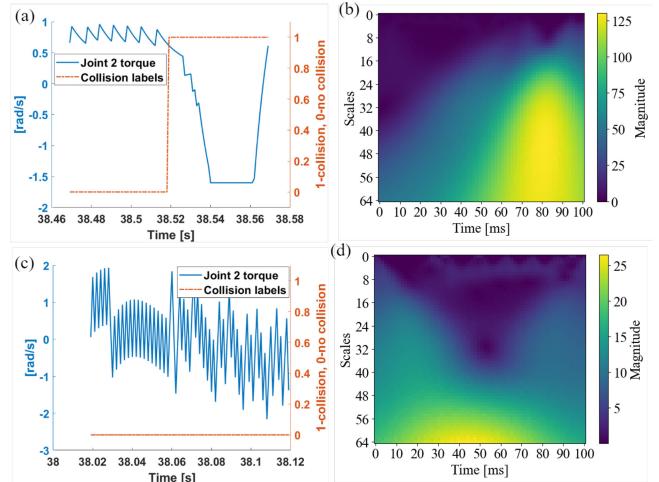


Fig. 5. Figures consist of two main components: raw torque signals and CWT scalograms. Specifically, (a) and (b) depict the torque signals recorded during a collision event and the corresponding CWT scalogram illustrating this collision. On the other hand, (c) and (d) showcase the torque signal within a collision-free scenario and the associated CWT scalogram for that specific instance.

time-series data. CWT is one of the wavelet transforms which can give high-scale resolution to reveal more intrinsic patterns.

CWT can be computed by a convolution process between the signal $x(t)$ and the complex conjugate of a family of wavelets as [30]

$$X_w(a, b) = \frac{1}{|a|^{1/2}} \int_{-\infty}^{\infty} x(t) \bar{\psi}\left(\frac{t-b}{a}\right) dt \quad (1)$$

where $\psi(\cdot)$ is the continuous mother wavelet that is translated by a factor of b and scaled by a factor of a . During the transform, $a > 0$ can effectively stretch or compress the mother wavelet in time. Hence, $X_w(a, b)$, the correlation matrix, can be obtained by comparing $x(t)$ with compressed or stretched versions of $\psi(\cdot)$. It can be visualized as a 2-D map of similarity coefficients of the signal $x(t)$ with $\psi(\cdot)$ at time t and

scale a . When a collision occurs on the robot, an abrupt change is usually observed in proprioceptive signals, such as measured force, joint torque, and velocity. This specific pattern, which is indicated by local maxima on the 2-D map, is beneficial for the identification of robot state. Fig. 5 provides an illustration of the torque signals in the time domain alongside the CWT scalograms for scenarios with and without collisions. Each of these cases comprises 100 sampling points and has been subjected to decomposition via the second-order complex Gaussian wavelet. The evident distinctions in the scalograms between these two cases suggest that CWT scalograms hold the potential for effective collision detection applications.

B. Input Variable Selection

The input variables selection is vital for the accuracy and data efficiency of the data-driven collision detection technique. Proprioceptive signals, such as motor torque τ_m , joint position θ , and joint velocity $\dot{\theta}$, are classical selections for data-driven methods to find the internal relationship between these signals and recognize the collision and collision-free cases. Park et al. [16] chose the moment observer torque τ_{MOB} , which is the low-pass filtered torque estimation as input. However, the only consideration of τ_{MOB} is not sufficient, since friction is also contained in τ_{MOB} . In [31], $\dot{\theta}$ is used to compensate for the effect of nonlinear friction. Kim et al. [17] pointed out that in addition to τ_{MOB} and $\dot{\theta}$, the joint position tracking error θ_{err} and the velocity tracking error $\dot{\theta}_{err}$ are beneficial for the distinction between collision and collision-free states. Additionally, error-related input is effective in dealing with intrinsically imbalanced datasets. This article uses CWT to transform collected raw signals into 2-D time-frequency representations. Instead of finding the dynamic relationship within the input variables, the proposed CWT-CNN recognizes collision and noncollision events based on distinct features from the CWT reconstructed representations.

The effectiveness of the CWT-CNN framework relies on creating distinguishable CWT representations for collision and collision-free scenarios. Clear and pronounced differences in these representations are crucial for accurate discrimination. The inherent imbalance in the training dataset, with fewer collision cases, requires a sample-efficient training approach. The distinct features in the CWT scalograms for both cases not only improve the classification accuracy, but also alleviate training challenges on imbalanced datasets. Pursuing sample efficiency during training enhances the overall robustness and effectiveness of the collision detection system [17].

In this study, we focus on the analysis and experimental evaluation of the following input candidates.

- 1) Var 1: τ .
- 2) Var 2: $\tau, \dot{\theta}$.
- 3) Var 3: $\tau, \dot{\theta}, \theta_{err}$.
- 4) Var 4: $\tau, \dot{\theta}, \dot{\theta}_{err}$.
- 5) Var 5: $\tau, \dot{\theta}, \theta_{err}, \dot{\theta}_{err}$.

where $\theta_{err} = \theta_d - \theta$ and $\dot{\theta}_{err} = \dot{\theta}_d - \dot{\theta}$.

The input torque τ and the velocity $\dot{\theta}$ are intuitive signals to reflect the distinction between collision-free and collision states. Kim et al. [17] pointed out that joint position

tracking error θ_{err} and joint velocity tracking error $\dot{\theta}_{err}$ show evident increases under collision compared to noncollision cases. However, the authors also mentioned that error-related features depend on tracking performance and could improve or degrade performance if different robot platforms are used. It is necessary to verify the actual performance while using these input signals. Therefore, Var 3 and Var 4 are designed to check whether performance can be improved with each error-related feature. Var 5 is designed to check the performance of a combination of all the signals considered. In Section IV-A, the performance based on the input variables is compared and discussed.

C. Network Structure

We propose to use a CNN to extract critical features from CWT scalograms and accurately classify the collision and collision-free cases. CNN provides efficient processing of 2-D data structures, significant reduction in parameter complexity, streamlined training procedures, and enhanced network performance [32]. The proposed structure is shown in Fig. 1. Before being fed to CNN, the scalograms of different signals are placed stacked on each other like the three channels red, green, and blue (RGB) of a color image. The benefit of this feeding-type data is that all the interdependencies between different signals can be taken into account. The proposed structure contains two convolutional layers (depth: 64, 32), two max-pooling layers, and two dense layers (size: 128, 54). The CNN network structure is expected to extract the most extreme features: abrupt changes and high-pixel density on different channels. Therefore, max pooling instead of average pooling is used to investigate better performance. Moreover, we adopted the rectified linear unit (ReLU) activation function instead of the hyperbolic tangent (tanh) for all convolutional layers and dense layers to overcome the problem of a vanish gradient, which can speed up the training process and improve performance. The output layer uses the sigmoid to output the likelihood of a collision. Batch normalization is applied to all layers. The whole structure requires 546 μ s to be computed on an Intel i7-12700K with twelve cores and a clock between 3.6 and 5 GHz, allowing the proposed structure for collision detection in real time.

D. Data Collection

The data collection process uses the platform described in Section II to conduct experiments. During this process, the robot performs fully random point-to-point acceleration-uniform-deceleration motions in the joint space without any collision reaction strategy. A person collides with a specific link at a random position and at a random moment with a collision tool (limit switch) to record the ground truth (1: collision; 0: no collision). The collisions vary in intensity, including both strong and weak impacts, and are induced at different speeds, with the robot moving at a maximum joint velocity of 4.3029 rad/s. Fig. 6 shows the collision tool and the process for collecting the test collision data. The joint space dataset $(\tau, \dot{\theta}, \theta_{err}, \dot{\theta}_{err})$ is collected at 1000 Hz. Two scenarios are considered during collection: 1) collision

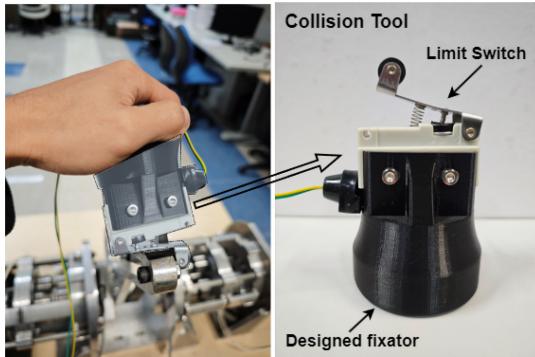


Fig. 6. Limit switch used to collide with the manipulator and record the ground truth collision labels. It outputs binary results: 1—collision and 0—no collision.

TABLE I
TRAINING AND TESTING DATASET AND THE CORRESPONDING TIME

Training	Motion collision	Joints Stiffness	Link 2
Testing	Motion	Joints Stiffness	Link 1 Link 2
		Highest level	5min 5min
	Collision	Highest level	5min 5min
		3rd level	5min 5min
		2nd level	5min 5min
Collision-free		Highest level	15min
		3rd level	15min
		2nd level	15min

and 2) collision-free. For each scenario, there are three cases for different physical stiffness (stiffness level 2, 3 and 4). Based on our previous work in [29], the lowest-stiffness level [referred to as “stiffness level 1” and detailed in Fig. 3(b)] is exclusively designated for the safety configuration. In this configuration, when a collision is detected, all joints are adjusted to the lowest-stiffness level. Consequently, the data collection process does not include the lowest-level stiffness configuration.

For training, only two-minute collision data is required. We trained the proposed network structure with only collision data from Joint 2 with the highest stiffness [stiffness level 4 in Fig. 3(b)]. The other collected data, joint 2 with other stiffness [stiffness levels 2 and 3 in Fig. 3(b)] and joint 1 with three stiffness levels [stiffness levels 2, 3, and 4 in Fig. 3(b)], are used for the validation of detection robustness and TL. Training and testing data scenarios and the corresponding time are shown in Table I. “Link 1” and “Link 2” refer to the specific links on the robot where collisions were carried out. There are a total of 30-min collision motions (516 collisions) and 45-min collision-free motions for testing.

E. Training and Implementation Details

Training data in Table I are randomly shuffled prior to training. The network structure is implemented in Python with Keras and Tensorflow (Python 3.10, Keras 2.10.0, and Tensorflow 2.10.0). The mini-batch size is 368 and the number of epoches is set to be 10. The optimizer is Adam with default parameters ($\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 1e-07$) and the learning rate of $lr = 1e-3$. We use BinaryFocalCrossentropy as the loss function to improve the network’s ability to handle

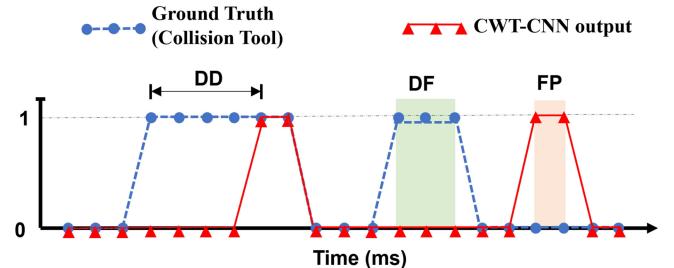


Fig. 7. Explanation of DD, DF, and FP. The dots represent the samples.

the imbalanced dataset. The focal loss [33] is defined as

$$\text{FL}(p_t) = -\alpha(1 - p_t)^\gamma \log(p_t) \quad (2)$$

where

$$p_t = \begin{cases} p, & \text{if } y = 1 \\ 1 - p, & \text{otherwise} \end{cases} \quad (3)$$

and $p \in [0, 1]$ is the output probability for the class with the label $y = 1$. The collected collision dataset is highly imbalanced, which means that the collision-free case data are more than the collision cases. In the initial stages of training, the output probability, denoted as p_t , for both collision and noncollision cases remains relatively low. Consequently, the term $(1 - p_t)^\gamma$ within (2) approaches 1, resulting in a loss function equivalent to standard cross-entropy (CE) loss. As the model performance evolves, achieving a near-perfect classification accuracy for collision-free cases (approaching 1), $(1 - p_t)^\gamma$ tends to zero. The loss for collision-free cases is down-weighted. Consequently, the model is focused more on improving the classification accuracy for the minority class of collision data. $\alpha \in [0, 1]$ is a weighting factor and $\gamma > 0$ is a tunable focusing parameter; here we set $\alpha = 0.25$, $\gamma = 2$ based on [33].

IV. EXPERIMENTAL RESULTS OF TRAINED MODEL

The proposed CWT-CNN model for collision detection is trained with two-minute collision data from Joint 2 with the highest stiffness. The testing dataset is also from Joint 2 with the highest-stiffness level. The testing dataset includes 5 min of collision data, which contains 86 random collisions, and 15 min of collision-free data.

To evaluate the proposed approach and facilitate a fair comparison with state-of-the-art methods, we utilize three criteria: 1) detection failure (DF) number (DFN); 2) detection delay (DD) in milliseconds (ms); and 3) false positive (FP) number (FPN), following the evaluation framework in [17], which is shown in Fig. 7. The collision tool generates a “1” signal upon intentional collisions, and the network predicts collisions based on a predefined threshold of 0.9. DD is the temporal gap between the actual collision moment and the network’s prediction. DFN indicates misclassified collisions, while FPN counts instances of noncollision events incorrectly classified as collisions. Both DFN and DD indicate the network’s sensitivity to collisions, while minimizing FPN is crucial for operational efficiency, particularly in scenarios requiring a prompt stop action. Consecutive FPs within the

TABLE II
TEST PERFORMANCE ACCORDING TO THE INPUT VARIABLES

	Collision (5mins)		Collision free (15mins)
	DFn	DD(ms)	FPn
Var 1 (τ)	4/86	31.6585	2
Var 2 ($\tau, \dot{\theta}$)	1/86	21.0824	8
Var 3 ($\tau, \dot{\theta}, \theta_{err}$)	2/86	25.4286	2
Var 4 ($\tau, \dot{\theta}, \dot{\theta}_{err}$)	3/86	25.7470	117
Var 5 ($\tau, \dot{\theta}, \theta_{err}, \dot{\theta}_{err}$)	2/86	20.8810	5

same time step are consolidated into a single FP event for assessment.

A. Performance According to the Input Variable

In this section, the classification results with five types of input variables listed in Section III-B are investigated. The network is the same for each input type, and the first-order complex Gaussian wavelet (cgau1) is used to transform the raw signal into a 2-D time-frequency representation (coefficient matrix). The wavelet scale range is fixed to [1, 64] and the size of the moving window is 100 sample points (100 ms). The coefficient matrix of each signal is scaled to 64×64 before feeding to the network. The comparison results are shown in Table II.

The following features are discovered by comparing the results for each input type.

- 1) Var 2 shows the best-detection accuracy for collision cases. Although it has a delay of 0.2 ms compared to Var 5, Var 2 still shows relatively good performance in detection sensitivity (DFN and DD).
- 2) *Var 1 and Var 2:* By adding velocity information, Var 2 achieves a better-detection sensitivity compared to Var 1. However, FPs increase slightly in the 15-min collision-free test dataset.
- 3) *Var 2 and Var 3 and Var 2 and Var 4:* FPs are reduced by adding joint tracking error in Var 3. However, the detection sensitivity has decreased, particularly for the performance of the DD. It is noticed that the velocity tracking error contained in Var 4 results in a dramatic increase in FPN, and compared to Var 2, Var 4 has a lower-detection sensitivity.
- 4) Var 5 performs fairly well in terms of FP and detection accuracy, with a low DD. However, the performance improvement is not distinct compared to Var 2. Moreover, less input signals are preferred, especially considering the transform time during CWT and the parameters that need to be trained in CNN. Therefore, Var 2 is selected as the input variable for the following experiments.

B. Performance According to the Wavelet Kernel Types

This section explores the effectiveness of various wavelet kernels (mother wavelets) in robot collision detection. The choice of a mother wavelet significantly influences time-frequency analysis results, yet determining the most suitable wavelet for a specific research context remains an open question in the academic community. While the Morlet wavelet

is considered effective for mechanical dynamical signal diagnosis [34], the increased complexity of robots, especially in their interactions with the environment, necessitates an experimental examination of the impacts of different wavelet kernels on collision detection performance.

We assessed 21 commonly used wavelet kernels in CWT, including Morlet, Mexican hat, Gaussian (gaus1-gaus8), Complex Gaussian (cgau1-cgau8), Complex Morlet (cmor 1-1), Shannon (shan 1-1), and B-Spline (fbsp 1-1.5-1). Each kernel was applied to a training dataset (2 min of collision data) and a testing dataset (5 min of collision data and 15 min of free-motion data). The wavelet scale range was set from 1 to 64, with a time-moving window size of 100 sample points (100 ms). The resulting time-frequency representations for collision and collision-free moments for each kernel are depicted in Fig. 8, where bright yellow indicates high-amplitude coefficients, and dark blue represents low amplitudes. The representations exhibit clear differentiation between collision and collision-free instances, facilitating straightforward identification of these scenarios for training and testing the proposed CNN structure.

Test results for detection metrics (DFN, DD for 5 min of collision data, FPN for 15 min of collision-free data) and conversion time for 21 wavelet kernels are depicted in Fig. 8. The Shannon wavelet (shan 1-1) exhibits high sensitivity with 100% detection accuracy and the second-shortest DD. Conversely, the first-order complex Gaussian wavelet (cgau1) has the lowest-FP number (FPN) at eight occurrences within 15 min of free-motion data. The fourth-order Gaussian wavelet (gaus4) demonstrates the shortest processing time at 1.4930 ms. Considering the importance of detection sensitivity and minimizing FPs for efficient task execution, the selection of wavelet kernels is crucial. Given these considerations, Shannon (shan 1-1) and the second-order complex Gaussian (cgau2) were chosen for in-depth investigations into the effects of wavelet scale range and time-moving window size on collision detection performance.

C. Performance According to the Wavelet Scale Ranges

This section discussed the effects of the scale range on collision detection performance. Selecting an appropriate scale range is significant for the application of CWT. Based on (1), in general, a smaller scale size a allows for more emphasis on sudden transitions. These abrupt changes are usually the most crucial traits. On the contrary, a wide range of scales provides more information for the slowly changing signals. We compared the collision detection performance of two types of wavelet kernel, the Shannon wavelet (shan1-1) and the second-order complex Gaussian wavelet (cgau2), within different scale ranges: [1, 8], [1, 16], [1, 32], [1, 64], and [1, 128].

Table III summarizes the results. The Shannon wavelet (shan1-1) demonstrates higher sensitivity (smaller DFN and DD) to collision cases across all scale ranges compared to the second-order complex Gaussian wavelet (cgau2). However, cgau2 exhibits a shorter conversion time than shan1-1 within all scale ranges. In the 15-min collision-free testing dataset, cgau2 has fewer FPN for all scale ranges except [1, 32] and

Wavelet kernel type	Kernel visualization	Torque CWT visualization		Velocity CWT visualization		Collision data (5-min)		Free-motion data (15-min)	Conversion time (ms)
		Collision	Free-motion	Collision	Free-motion	DFn	DD (ms)		
morl						1/86	23.6118	11	1.7264
mexh						0/86	18.9767	43	1.7480
gaus1						1/86	18.7500	27	1.4976
gasu2						1/86	17.5882	18	1.4961
gaus3						0/86	19.0349	33	1.5181
gaus4						0/86	17.3140	34	1.4949
gaus5						0/86	19.8953	48	1.5164
gaus6						0/86	18.8953	36	1.4930
gaus7						1/86	23.8471	30	1.5420
gaus8						0/86	18.1977	18	1.5405
cgau1						1/86	21.0824	8	1.8333
cgau2						1/86	16.9412	43	1.8150
cgau3						0/86	19.0698	41	1.8083
cgau4						1/86	19.8471	38	1.8440
cgau5						2/86	22.2619	22	1.9149
cgau6						0/86	19.9070	16	1.9005
cgau7						1/86	19.8235	15	1.9641
cgau8						2/86	24.9048	24	1.9753
cmor 1-1						1/86	20.2588	12	2.1945
shan 1-1						0/86	16.7647	19	3.0038
fbsp 2-1.5-1						1/86	16.7059	12	3.8082

Fig. 8. Evaluation of 21 wavelet kernels on collision detection performance. Each wavelet kernel and the visualization of the time–frequency representations for one collision moment and one collision-free moment are shown. Bright yellow colors indicate high-amplitude coefficients, while dark blue colors indicate low amplitudes. The distinction representation between collision and collision-free cases allows the network to easily identify two scenarios even with a small amount of samples. Collision detection performance and transform time for each wavelet kernel are experimentally evaluated.

[1, 64]. The smallest scale range ([1, 8]) shows the shortest DD and the most efficient conversion time for both wavelet kernels. This is attributed to capturing sharp transitions during collisions in the smaller scale range. However, it introduces a higher occurrence of FPs due to high-frequency noise. The largest scale range [1, 128] exhibits respectable detection sensitivity, but FPs increase significantly, suggesting dominance

of low-frequency features. This similarity in representations leads to more misclassifications. Additionally, the conversion time for both wavelet kernels is the longest in this scale range, making it inefficient for collision detection.

For the cgau2 wavelet at the [1, 32] scale range, no DFs occur. However, this scale range is linked to the longest DD and a relatively higher FPn compared to other selected

TABLE III
PERFORMANCE ACCORDING TO THE WAVELET SCALE RANGE

Wavelet type	Scale range	Collision (5min)		Collision-free (15min)	Conversion time (ms)
		DFn	DD (ms)		
cgau2	[1, 8]	1/86	16.1429	159	0.3199
	[1, 16]	1/86	20.4000	8	0.4704
	[1, 32]	0/86	20.9647	76	0.8609
	[1, 64]	1/86	16.9412	43	1.8150
	[1, 128]	0/86	18.9302	48	4.6152
shan1-1	[1, 8]	1/86	14.6463	209	0.3223
	[1, 16]	0/86	16.2771	82	0.5673
	[1, 32]	1/86	18.9294	7	1.3307
	[1, 64]	0/86	16.7647	19	3.0038
	[1, 128]	0/86	23.6977	62	12.0038

scales for cgau2. Scale ranges [1, 16] and [1, 64] with cgau2 yield identical DF numbers (DFNs) but differ in DD and FPNs. Specifically, the [1, 64] scale range has a shorter DD but a significantly higher FPN than the [1, 16] scale range. For the shan1-1 wavelet, the [1, 64] scale range exhibits superior detection sensitivity compared to [1, 32], but it comes with higher FPNs and longer conversion times. In practical applications, handling FPs can involve techniques like continuous filtering and ensemble methods, as discussed by Kim et al. [17], but these may affect detection sensitivity. Considering the analysis, [1, 16] and [1, 64] scale ranges are suitable for cgau2, while [1, 32] and [1, 64] are appropriate for shan1-1, depending on the chosen wavelet kernel for a specific application.

D. Performance According to the Time Moving Window Sizes

In addition to the wavelet kernels and the wavelet scale range, the size of the time-moving window is also crucial for real-time collision detection with CWT-CNN. As shown in Fig. 9, a time-moving window includes the current time t and T past points, i.e., $x_{in} = [x_{in}(t - T + 1), x_{in}(t - T + 2), x_{in}(t - T + 3), \dots, x_{in}(t - 1), x_{in}(t)]$. The size of the time-moving window T is a crucial parameter, determining the extent of historical information included in the time-frequency representation of the current moment. For collision moments, a smaller T concentrates more collision-specific features in the representations, while a larger T includes more collision-free information for comparative analysis. It's worth noting that CWT with a smaller T has a shorter computational transformation time compared to CWT with a larger T .

Based on the discussion of the above sections, Var 2 ($\tau, \dot{\theta}$) is selected as the source signals, and the scale range is set as [1, 64]. Similarly to Section IV-C, we also studied the effects of variant time-moving window size with two wavelet kernels, the Shannon wavelet (shan1-1) and the second-order complex Gaussian wavelet (cgau2). Five sizes of time-moving window are tested (50, 100, 200, 300, 500), and the results are shown in Table IV.

Among the tested time-moving window sizes, smaller sizes (50 and 100) exhibit superior detection sensitivity for both wavelet kernels. However, the smallest size (50) leads to a higher FPN during the 15-min collision-free data testing phase. This could be attributed to the small window size allowing accurate recognition of collision cases due to a high proportion of collision features in CWT representations.

TABLE IV
PERFORMANCE ACCORDING TO THE TIME-MOVING WINDOW SIZE

Wavelet type	Scale range	Collision (5min)		Collision-free (15min)	Conversion time (ms)
		DFn	DD (ms)		
cgau2	50	1/86	18.3765	40	1.5496
	100	1/86	20.4000	8	1.8150
	200	1/86	28.7500	8	2.2519
	300	3/86	36.0610	4	2.7612
	500	0/86	25.9294	88	3.6659
shan1-1	50	1/86	17.6824	85	2.9913
	100	0/86	16.7447	19	3.0038
	200	0/86	27.3810	33	5.3343
	300	0/86	49.7176	30	6.9391
	500	0/86	67.1176	58	10.6070

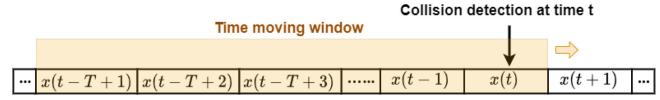


Fig. 9. Illustration of time moving window size. Data up to the window size T is transformed to a time-frequency representation via CWT.

Nevertheless, noise during collision-free motions may exhibit similar variations within the smallest window size (50), resulting in a higher FPN in the free-motion dataset testing. The largest moving window size (500) also performs poorly regarding FPN. This is because, with a longer time-moving window, the collision moment comprises a smaller portion compared to collision-free states, increasing the likelihood of misclassification in CWT representations for both states. Additionally, a longer time-moving window is associated with increased DD and longer conversion times compared to other window size options.

For cgau2, the 300-sample window size yields the smallest FPN among other sizes. However, its poor performance in detection sensitivity makes it unsuitable for collision detection applications. Window sizes 100 and 200 display identical DFN and FPN, but the 100-sample window surpasses the 200-sample window in terms of DD. Regarding Shan1-1, the 100-sample window size stands out as the optimal choice compared to sizes 200 and 300, demonstrating superior performance in terms of DD, FPN, and conversion time.

E. Performance Comparison to Existing Methods

In this section, we conduct a comparative analysis of CWT-CNN with two existing learning-based approaches as reference points: 1) MNN introduced by Kim et al. [17] and 2) the one-dimensional (1-D) CNN method presented by Park et al. [16]. MNN has demonstrated exceptional collision detection performance for the Doosan robot M0609, showcasing potential transferability to other robots of the same model through fine-tuning. The 1-D CNN structure proposed in [16] represents a prominent end-to-end learning-based technique for collision detection. These state-of-the-art methods are chosen for comparison with our proposed model in the domain of collision detection for collaborative robots equipped with VSAs.

In the study [17], the training of MNN requires over 1 h of collision data and 2 h of collision-free data. In contrast, the proposed CWT-CNN model only needs 2 min of collision data, comprising 47 collision events. To ensure a fair comparison,

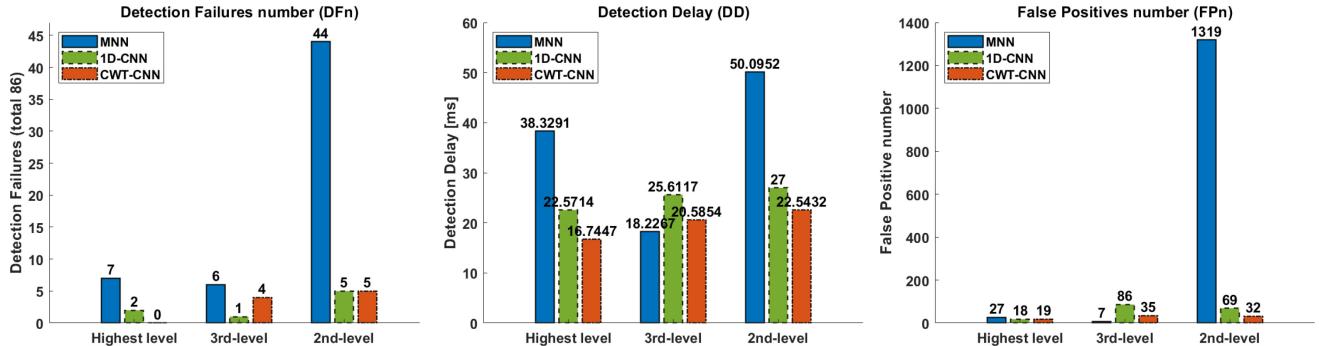


Fig. 10. Performance comparison results between MNN [17], 1-D CNN [16], and CWT-CNN for collision detection with respect to the DFN, DD in millisecond, and the number of FPs (FPN).

we use the same 2 min of collision data from joint 2 at the highest-stiffness level to train all models. In the context of collision detection for collaborative robots with VSAs, an essential characteristic is a network's ability to maintain robustness when facing changes in stiffness. This implies that testing scenarios should align not only with the stiffness level in the training dataset but also include other stiffness levels. Hence, the testing dataset, collected from the second link, comprises 15 min of collision data (5 min for each stiffness level, totaling 258 collisions) and 45 min of collision-free data (15 min for each stiffness level). The detection performance of all trained networks is evaluated across different stiffness levels.

The results are shown in Fig. 10. In the testing scenario corresponding to the highest-stiffness level, which matches the stiffness of our training dataset, the CWT-CNN model demonstrates superior performance. It successfully detects all collisions with a minimal DD of 16.7447 ms and 19 FPs. In the evaluation of the robustness of the detection under varying stiffness conditions, CWT-CNN consistently demonstrates high performance. Although there are three additional DFs compared to the 1-D-CNN model at the third stiffness level, the CWT-CNN model maintains a detection accuracy exceeding 95%. In particular, at this stiffness level, the MNN model outperforms the other models in terms of DD and FPs. However, MNN exhibits suboptimal performance at both the highest and second-level stiffness settings in all evaluation criteria. This observation suggests that, due to the constraints of limited training data, MNN faces challenges in effectively handling collision detection in robots equipped with VSAs. Taking all evaluation criteria and stiffness levels into account, CWT-CNN consistently demonstrates superior detection performance and greater robustness to stiffness variations.

V. TRANSFERABILITY ANALYSIS OF CWT-CNN

Collecting collision data for training purposes is challenging due to the risks and effort involved in intentionally colliding with the robot during its movement. In particular, to endow the neural network model with generalization, collision locations should cover all moving links. However, colliding with some links, such as links close to the base, is more difficult than colliding with other links. It is also dangerous when the robot is moving randomly. Therefore, TL can be used to take full advantage of the pretrained network from one joint data. The

new classification network for other joints can be obtained based on the pretrained network by TL with fine-tuning.

In this section, we propose to transfer the pretrained network from one joint to another to further improve data efficiency and generalization. The overall framework is illustrated in Fig. 11. Initially, CWT-CNN is trained using collision data obtained from joint 2 at the highest-stiffness level. In the TL phase, the CNN module is directly transferred to a new joint network, with the parameters in the CNN module kept frozen during fine-tuning. Fine-tuning is carried out using 30-s free-motion data from the new joint, which includes 10 s for each stiffness condition. The parameters in the last two dense layers are updated with a low-learning rate of 1e-5 and a small number of epochs (5 epochs). We conducted a comparative study involving fine-tuning and a scenario without fine-tuning [direct transfer (DT)]. Fig. 14 illustrates the comparison of collision moment output possibilities between these two cases. The network transferred without fine-tuning can identify collision cases, but it assigns relatively low-possibility scores, indicating lower sensitivity to such occurrences. On the contrary, fine-tuning increases the network's sensitivity to collisions, as evidenced by the increased output possibility scores. Fig. 12 shows the results for the entire testing dataset in joint 1. Compared with DT, fine-tuning yields higher-detection accuracy and reduces the DD. This suggests an improved sensitivity to collision cases following fine-tuning. However, fine-tuning also increases sensitivity to noise, evident in the output possibility for fine-tuning, as shown in Fig. 14. This characteristic leads to an increase in FPs after fine-tuning, as demonstrated in Fig. 12(c).

It is worth highlighting that the proposed CWT-CNN exhibits robustness to variations in stiffness. Moreover, it can be effectively transferred to different joints across various stiffness conditions through a fine-tuning process, requiring only 30 s of free-motion data from the new joints. This approach eliminates the requirement to collect collision data from these new joints, which offers a significant advantage.

VI. DISCUSSION

A. Comparison of Different Time-Frequency Analysis Methods for Robot Collision Detection

In this article, we mainly studied the CWT for robot collision detection. However, other time-frequency analysis

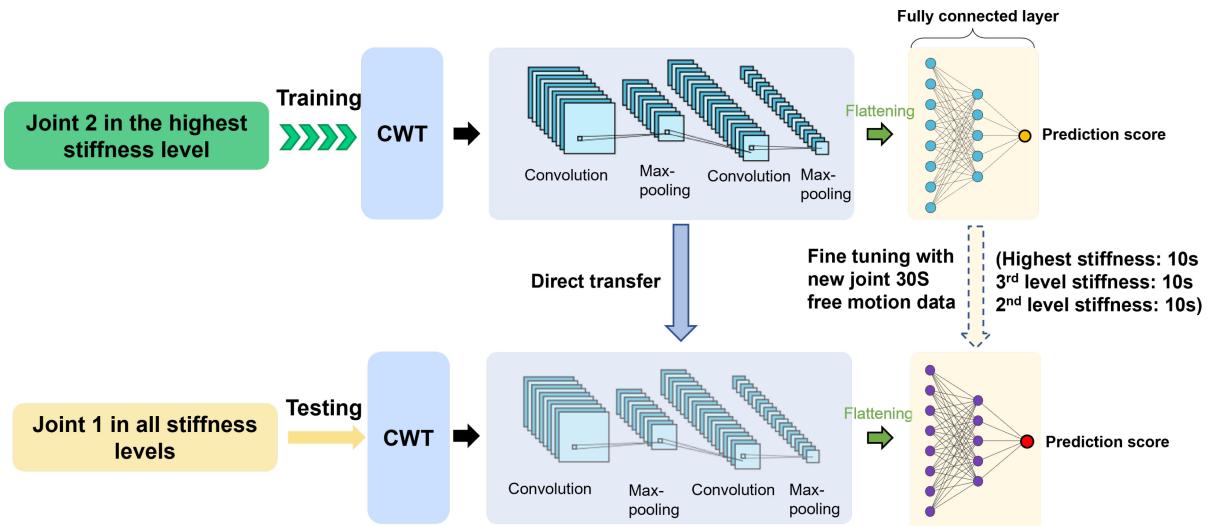


Fig. 11. TL framework is applied to distinct joints under varying stiffness levels. Specifically, CWT-CNN is initially trained using collision data from joint 2 under the highest-stiffness setting. The pretrained CWT-CNN model can then be adapted to a new joint by fine-tuning the last fully connected layers using 30 s of free-motion data from that specific joint.

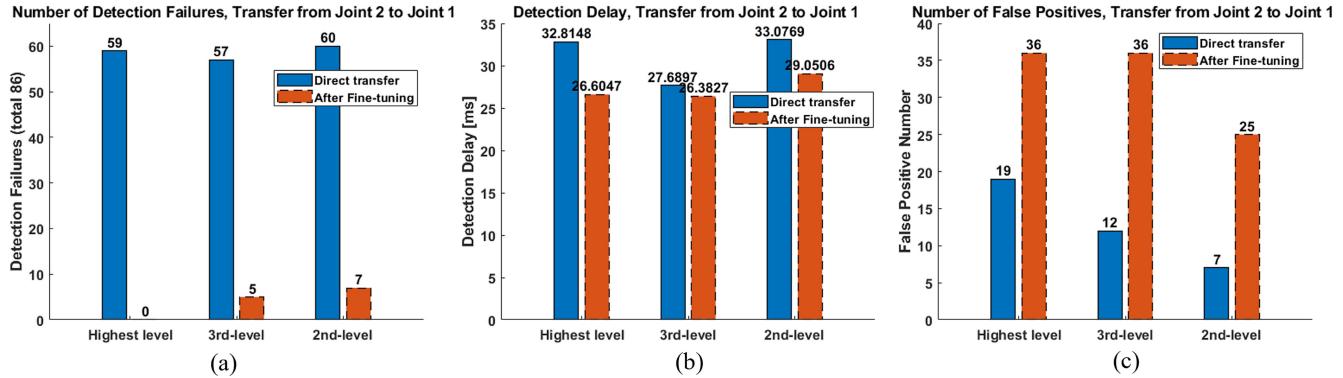


Fig. 12. Comparison of TL results at different stiffness levels, using direct transfer and fine-tuning. The performance is indicated by (a) number of DFs, (b) DD, and (c) FPN.

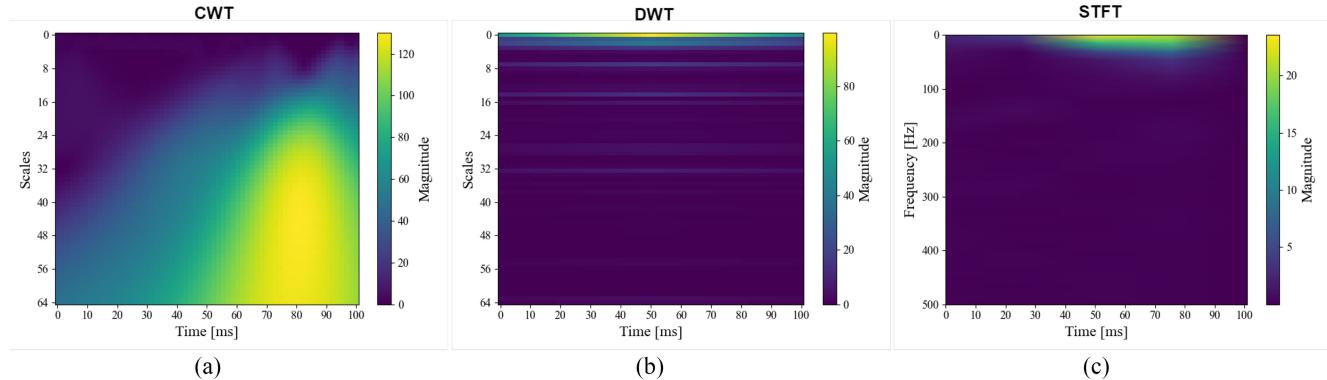


Fig. 13. Visualization of a collision case using three different time-frequency analysis methods: (a) CWT, (b) DWT, and (c) STFT. Each subplot illustrates the analysis results using the respective method, highlighting their unique approaches to capturing the temporal and frequency characteristics of the collision event.

methods also offer valuable insights. One such method is the short-time Fourier transform (STFT). STFT analyzes nonstationary signals by applying a sliding-window approach. This method divides the signal into short, often overlapping segments and performs the Fourier Transform on each segment iteratively, covering the entire signal [35]. Although STFT is effective, it has a notable limitation: the tradeoff between time

and frequency resolution. Using a longer window improves frequency resolution but reduces time resolution, whereas a shorter window enhances time resolution at the cost of frequency accuracy [36]. Another method, the DWT, reduces the computational complexity compared to CWT. However, it may not achieve sufficient resolution at very high scales and frequencies [21].

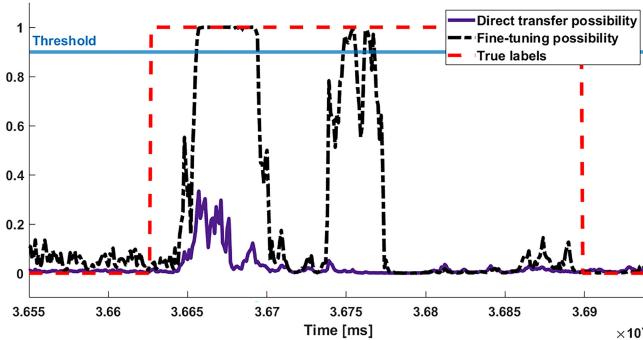


Fig. 14. Output possibility comparison for one collision moment. Compared with DT, the network after fine-tuning increases the output possibility of collision cases, improving the detection sensitivity.

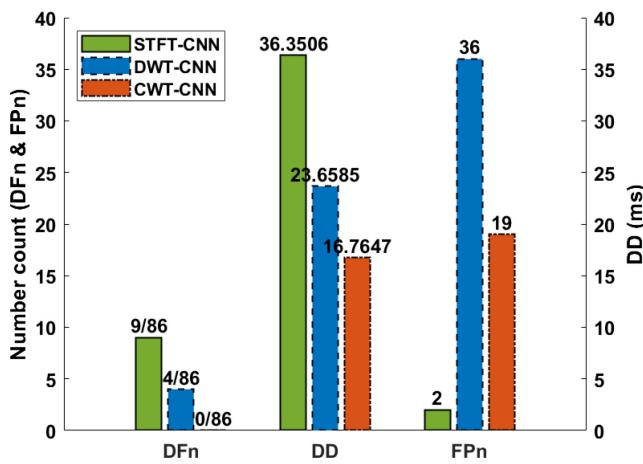


Fig. 15. Experimental comparison results for CWT-CNN, DWT-CNN, and STFT-CNN.

Fig. 13 presents visualizations of a collision case using three distinct time–frequency analysis methods: 1) CWT; 2) DWT; and 3) STFT. The plots highlight notable differences in their ability to depict abrupt changes in the signal. Notably, CWT reveals clearer and more pronounced features of abrupt changes compared to DWT and STFT. This clarity underscores CWT’s effectiveness in facilitating feature extraction crucial for collision detection using CNN.

To better illustrate the effectiveness of different time-frequency analysis methods for robot collision detection, we compared the experimental detection performance of STFT-CNN, CWT-CNN, and DWT-CNN. The CNN architecture for all three methods is identical, as detailed in Section III-C. The training and testing datasets are the same for all methods, as described in Section III-D, and the training and implementation details are consistent across methods, as outlined in Section III-E. For DWT-CNN, we utilized the Haar wavelet kernel. The Haar wavelet is known for its efficiency in multiresolution analysis, noise reduction, and edge detection [37]. We applied the same scale range [1, 64] and a time-moving window size of 100, consistent with the parameters used for CWT-CNN. For STFT-CNN, we set the segment length to 64, which matches the input array length used in CWT-CNN.

The comparison results for CWT-CNN, DWT-CNN, and STFT-CNN are presented in Fig. 15. Our findings indicate that CWT-CNN outperforms the other methods, achieving

100% detection accuracy and the fastest response time. In contrast, STFT-CNN exhibits the lowest-detection accuracy and a longer DD, attributed to the inherent tradeoff between frequency and time resolution in STFT. This tradeoff results in suboptimal collision detection sensitivity, thereby reducing FPs but also missing some collision events.

DWT-CNN demonstrates reasonable detection performance; however, it does not match the accuracy and response time of CWT-CNN. In our study, we utilized the Haar wavelet for DWT-CNN, which, while effective, may not be the optimal choice for collision detection. Future work could explore other DWT wavelets to potentially improve performance. Nonetheless, our extensive experiments highlight the superior performance, generalization, and robustness of CWT-CNN for collision detection, validating its efficacy across various scenarios.

B. Generalization of CWT-CNN for Industrial Robots

To further validate the effectiveness and practicality of the CWT-CNN for robot collision detection, we extend our TL approach to industrial robots, showcasing its potential for broader applications in real-world robotic systems, where precise and rapid collision detection is crucial for maintaining operational safety and efficiency.

The experimental setup for industrial robots is illustrated in Fig. 16. We employed an Aubo-i10 robot from Aubo Robotics, equipped with six degrees of freedom (6-DOF). The robot interface offered extensive user I/O options, which is beneficial for integrating our collision detection tool for ground truth label collection. We employed the Aubo SDK¹ to control the robot’s movement and gather joint data along with collision tool labels. The robot operating system (ROS) is used to orchestrate these processes efficiently. During the data collection process, the robot performed acceleration–uniform–deceleration motions in the joint space without any collision reaction strategy. Our collision detection tool, as detailed in Section III-D, was then applied to induce collisions at random points along the robot’s links, allowing us to record the corresponding ground truth data. To validate the TL capability of the proposed method, we specifically collected data involving collisions on different links (links 2 to 4), which are the links where collisions typically occur in real-world robotic environments. The dataset we gathered includes joint current i_m , position θ_m , and collision tool recordings, all sampled at a rate of 1000 Hz. Table V provides a summary of the dataset collected using the Aubo-i10. For training, we collected 2 min of collision data on each link, resulting in 40 collisions per link. We collected data as follows: 2 min of collision data per link for training, yielding 40 collisions per link; 10 min of collision data per link for testing, resulting in 80 collisions per link and 240 collisions in total; and 15 min of free-motion (collision-free) data for testing.

Based on the discussion of input variable selection in Section IV-A, we have chosen joint torque (τ_m) and joint velocity ($\dot{\theta}_m$) as inputs to our algorithm. Since direct measurement of joint torque is not possible with the Aubo-i10, we

¹<https://www.aubo-cobot.com/public/download4>

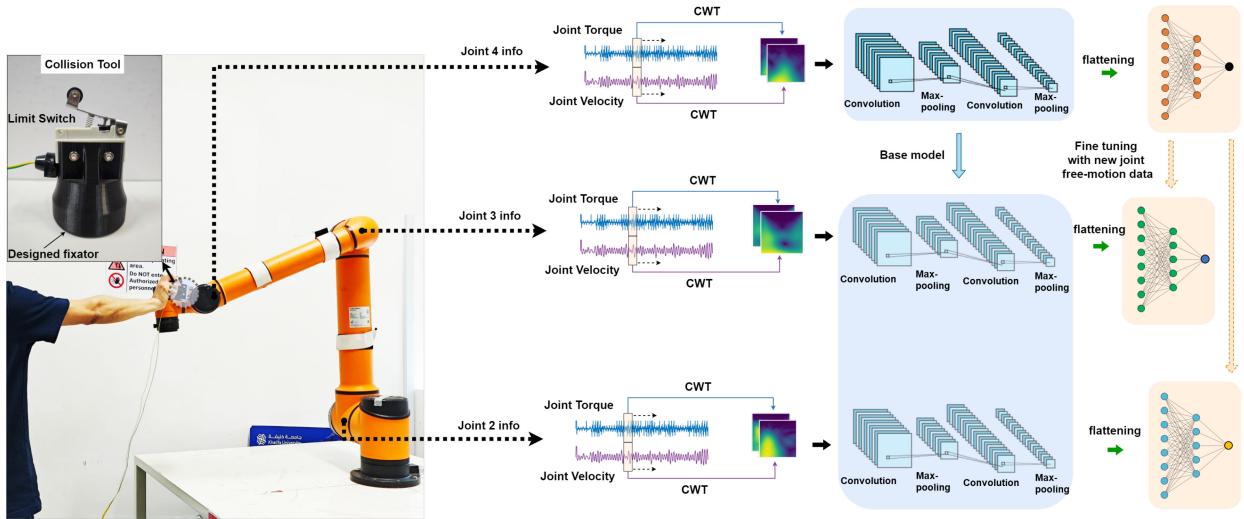


Fig. 16. Experimental setup for collision detection in an industrial robot using CWT-CNN. The proposed model is validated on the Aubo-i10 robot and can be seamlessly transferred across joints by fine-tuning with minimal free-motion data from the new joint.

TABLE V
TRAINING AND TESTING DATASET WITH AUTO ROBOT

	Motion	Link 2	Link 3	Link 4
Training	Collision	2mins	2mins	2mins
Testing	Collision	10mins	10mins	10mins
	Collision-free	15mins		

TABLE VI
RESULTS OF DETECTION FAILURE NUMBER FOR TL ON AUTO-I10
(DT: DIRECT TRANSFER AND TL: TRANSFER LEARNING)

Training\Testing	DFn (collision dataset, n/80)					
	joint 2		joint 3		joint 4	
DT	TL	DT	TL	DT	TL	
joint 2	4	nan	50	20	45	24
joint 3	2	2	5	nan	23	15
joint 4	2	4	3	3	3	nan

calculate the torque indirectly using the joint current ($k_m i_m$), where k_m represents the motor constant. The joint velocity is obtained by differentiating the joint position. We utilized the Shan 1-1 wavelet kernel with a wavelet scale range of [1, 64] and a time-moving window size of 100, as discussed in Sections IV-B–IV-D. We employed cross-validation across different joints to thoroughly evaluate the effectiveness of TL in the CWT-CNN model for industrial robotics. The model was initially trained using 2 min of collision data from one joint. We then tested the model using testing datasets from other joints, each containing 10 min of collision data and 15 min of collision-free data. Two scenarios were analyzed: 1) DT and 2) TL. In the DT scenario, the pretrained model from one joint was directly applied to the datasets of other joints without modification. In the TL scenario, the pretrained model was fine-tuned using the collision-free data (30 s) of the specific joint before being tested on that joint's dataset. The training detail is the same as Section III-E and the testing results are shown in Tables VI–VIII.

The CWT-CNN model exhibits exceptional performance when each joint is trained on its respective dataset. This is

TABLE VII
RESULTS OF DD FOR TL ON AUTO-I10. (DT: DIRECT TRANSFER AND TL: TRANSFER LEARNING)

Training\Testing	DD (ms) (collision dataset)					
	joint 2		joint 3		joint 4	
DT	TL	DT	TL	DT	TL	
joint 2	3.55	nan	4.36	10.90	4.47	5.57
joint 3	4.01	3.06	7.98	nan	5.07	6.26
joint 4	4.73	3.69	7.74	7.93	3.90	nan

TABLE VIII
RESULTS OF FPN FOR TL ON AUTO-I10. (DT: DIRECT TRANSFER AND TL: TRANSFER LEARNING)

Training\Testing	FPn (free-motion dataset)					
	joint 2		joint 3		joint 4	
DT	TL	DT	TL	DT	TL	
joint 2	2	nan	3	23	3	23
joint 3	19	33	24	nan	23	48
joint 4	148	21	62	21	49	nan

evidenced by the model's extremely high accuracy and minimum DD. Notably, the collision DD is significantly reduced compared to our earlier study on the robot with VSAs. This difference is likely due to the inherent compliance in flexible joints, which absorbs some impact energy during collisions, resulting in delayed detection. In contrast, industrial robots with nearly rigid joints enable the CWT-CNN to respond more rapidly to collision events.

The networks directly transferred from one joint to another can still distinguish between collision and collision-free cases, though their performance varies across different joint transfers. Notably, the network trained on joint 2 data exhibits the shortest DD and minimum FPs when detecting collisions on other links. However, this comes at the cost of reduced detection accuracy, resulting in more DFs. After fine-tuning with the corresponding joint's free-motion data, the detection accuracy improves, although the detection response and FPs are impacted. The network trained on joint 3 data demonstrates good detection performance when directly applied to joint 2

data. Additionally, it shows rapid detection response for joint 4 testing data, though the detection accuracy for joint 4 is somewhat degraded. After fine-tuning with the corresponding joint's free-motion data, the detection accuracy improves, but this comes at the cost of increased DD and a rise in FPs. The directly transferred network, initially trained on joint 4 data, exhibits good collision sensitivity on joint 2 and joint 3 testing data, as evidenced by minimal DFs and DD. However, this transfer also results in an increase in FPs. Fine-tuning can reduce the number of FPs, but it comes at the cost of a slight increase in DD DFs. In summary, the proposed CWT-CNN collision detection model demonstrates strong performance in industrial robots, highlighting its generalization capability.

Despite the demonstrated superiority of our proposed method over conventional learning-based approaches when using the same limited training data, there remains a scope for further improvement, particularly in addressing FPs. This can potentially be addressed by incorporating wavelet-based noise suppression techniques, as suggested in [38]. Moreover, future work will explore different network structures to further improve collision detection performance. While the current study demonstrates computational efficiency, the reliance on preprocessing may challenge deployment on resource-constrained platforms. Building on our comprehensive investigation into factors influencing collision detection, future work will optimize efficiency by integrating wavelet decomposition directly within the network architecture to reduce preprocessing overhead and improve deployment feasibility. Furthermore, while current supervised learning methods heavily rely on labeled collision data, it may be beneficial to explore unsupervised learning approaches, such as auto-encoders (AEs) [18], [39] or generative adversarial networks (GANs) [40], as a potential area of interest in future investigations. The current study focuses exclusively on hard collisions, thus not covering the full spectrum of collision types. In future work, we plan to extend our investigation to include both hard and soft collisions [16]. This expansion will allow us to evaluate the performance of our collision detection methods across a broader range of scenarios, enhancing the generalizability and robustness of our model.

VII. CONCLUSION

This article introduces CWT-CNN, a data-efficient and robust collision detection method for collaborative robots equipped with variable stiffness actuation. The proposed method uses the CWT to extract more informative features from proprioceptive signals, enhancing the recognition of collision events in the collaborative robot setting. Compared to conventional learning-based methods, the proposed CWT-CNN is much more sample-efficient and robust to variation in system stiffness. Minimum collision data are sufficient for training this model, rendering it highly suitable for handling imbalanced and hard-to-collect collision scenarios. Furthermore, this article conducts a comprehensive investigation into the influence of various variables, encompassing input signal types, mother wavelet types, wavelet scale ranges, and time-moving window size, on robot collision detection.

Our experimental findings offer valuable insight and practical recommendations for selecting the optimal CWT parameters to enhance collision detection performance.

Moreover, the proposed method demonstrates good generalization performance to random motion, collision location, and various joint stiffness conditions. To further alleviate the need to collect collision data on different robot links, we introduce a transferability analysis based on TL. By fine-tuning with a small amount of free-motion data from the new joint, the pretrained CWT-CNN model can be effectively transferred to the new joint. This transferability is validated through extensive experiments on industrial robots and the robot equipped with VSAs.

REFERENCES

- [1] X. Li, Q. Lu, J. Chen, N. Jiang, and K. Li, "A robust region control approach for simultaneous trajectory tracking and compliant physical human–robot interaction," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 53, no. 10, pp. 6388–6400, Oct. 2023.
- [2] A. Zacharaki, I. Kostavelis, A. Gasteratos, and I. Dokas, "Safety bounds in human robot interaction: A survey," *Saf. Sci.*, vol. 127, Jul. 2020, Art. no. 104667.
- [3] S. Kumar, C. Savur, and F. Sahin, "Survey of human–robot collaboration in industrial settings: Awareness, intelligence, and compliance," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 51, no. 1, pp. 280–297, Jan. 2021.
- [4] F. Flacco, T. Kröger, A. De Luca, and O. Khatib, "A depth space approach to human–robot collision avoidance," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2012, pp. 338–345.
- [5] S. M. Khansari-Zadeh and A. Billard, "A dynamical system approach to realtime obstacle avoidance," *Auton. Robots*, vol. 32, no. 4, pp. 433–454, 2012.
- [6] Z. Xu, X. Zhou, H. Wu, X. Li, and S. Li, "Motion planning of manipulators for simultaneous obstacle avoidance and target tracking: An RNN approach with guaranteed performance," *IEEE Trans. Ind. Electron.*, vol. 69, no. 4, pp. 3887–3897, Apr. 2022.
- [7] Z. Zhang et al., "A real-time 3-D visual detection-based soft wire avoidance scheme for industrial robot manipulators," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 54, no. 2, pp. 827–840, Feb. 2024.
- [8] S. Haddadin, A. De Luca, and A. Albu-Schäffer, "Robot collisions: A survey on detection, isolation, and identification," *IEEE Trans. Robot.*, vol. 33, no. 6, pp. 1292–1312, Dec. 2017.
- [9] V. Duchaine, N. Lauzier, M. Baril, M.-A. Lacasse, and C. Gosselin, "A flexible robot skin for safe physical human robot interaction," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2009, pp. 3676–3681.
- [10] G. Pugach, A. Melnyk, O. Tolochko, A. Pitti, and P. Gaussier, "Touch-based admittance control of a robotic arm using neural learning of an artificial skin," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, 2016, pp. 3374–3380.
- [11] C. Vergara, G. Borghesan, E. Aertbeliën, and J. De Schutter, "Incorporating artificial skin signals in the constraint-based reactive control of human–robot collaborative manipulation tasks," in *Proc. 3rd Int. Conf. Adv. Robot. Mechatron. (ICARM)*, 2018, pp. 280–287.
- [12] M. Teyssier, B. Parilusyan, A. Roudaut, and J. Steimle, "Human-like artificial skin sensor for physical human–robot interaction," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, 2021, pp. 3626–3633.
- [13] A. De Luca and R. Mattone, "Actuator failure detection and isolation using generalized momenta," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2003, pp. 634–639.
- [14] A. De Luca, A. Albu-Schäffer, S. Haddadin, and G. Hirzinger, "Collision detection and safe reaction with the DLR-III lightweight manipulator arm," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2006, pp. 1623–1630.
- [15] S. A. B. Birjandi, J. Kühn, and S. Haddadin, "Observer-extended direct method for collision monitoring in robot manipulators using proprioception and IMU sensing," *IEEE Robot. Autom. Lett.*, vol. 5, no. 2, pp. 954–961, Apr. 2020.
- [16] K. M. Park, J. Kim, J. Park, and F. C. Park, "Learning-based real-time detection of robot collisions without joint torque sensors," *IEEE Robot. Autom. Lett.*, vol. 6, no. 1, pp. 103–110, Jan. 2021.

- [17] D. Kim, D. Lim, and J. Park, "Transferable collision detection learning for collaborative manipulator using versatile modularized neural network," *IEEE Trans. Robot.*, vol. 38, no. 4, pp. 2426–2445, Aug. 2022.
- [18] K. M. Park, Y. Park, S. Yoon, and F. C. Park, "Collision detection for robot manipulators using unsupervised anomaly detection algorithms," *IEEE/ASME Trans. Mechatronics*, vol. 27, no. 5, pp. 2841–2851, Oct. 2022.
- [19] M. Lippi, G. Gillini, A. Marino, and F. Arribiello, "A data-driven approach for contact detection, classification and reaction in physical human-robot collaboration," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, 2021, pp. 3597–3603.
- [20] Y. J. Heo, D. Kim, W. Lee, H. Kim, J. Park, and W. K. Chung, "Collision detection for industrial collaborative robots: A deep learning approach," *IEEE Robot. Autom. Lett.*, vol. 4, no. 2, pp. 740–746, Apr. 2019.
- [21] V. Ganjalizadeh, G. G. Meena, T. A. Wall, M. A. Stott, A. R. Hawkins, and H. Schmidt, "Fast custom wavelet analysis technique for single molecule detection and identification," *Nat. Commun.*, vol. 13, no. 1, pp. 1–9, 2022.
- [22] C. Livolsi, R. Conti, F. Giovacchini, N. Vitiello, and S. Crea, "A novel wavelet-based gait segmentation method for a portable hip exoskeleton," *IEEE Trans. Robot.*, vol. 38, no. 3, pp. 1503–1517, Jun. 2022.
- [23] O. Faust, U. R. Acharya, H. Adeli, and A. Adeli, "Wavelet-based EEG processing for computer-aided seizure detection and epilepsy diagnosis," *Seizure*, vol. 26, pp. 56–64, Mar. 2015.
- [24] R. He et al., "Automatic detection of atrial fibrillation based on continuous wavelet transform and 2-D convolutional neural networks," *Front. Physiol.*, vol. 9, p. 1206, Aug. 2018.
- [25] Y. Cheng, M. Lin, J. Wu, H. Zhu, and X. Shao, "Intelligent fault diagnosis of rotating machinery based on continuous wavelet transform-local binary convolutional neural network," *Knowl.-Based Syst.*, vol. 216, Mar. 2021, Art. no. 106796.
- [26] H. Ehyi, A. Nysveen, and J. A. Antonino-Daviu, "Advanced fault detection of synchronous generators using stray magnetic field," *IEEE Trans. Ind. Electron.*, vol. 69, no. 11, pp. 11675–11685, Nov. 2022.
- [27] C. Chen, T. Wang, C. Liu, Y. Liu, and L. Cheng, "Lightweight convolutional transformers enhanced meta-learning for compound fault diagnosis of industrial robot," *IEEE Trans. Instrum. Meas.*, vol. 72, pp. 1–12, May 2023.
- [28] Y. Xu et al., "Continuous wavelet analysis of leaf reflectance improves classification accuracy of mangrove species," *Remote Sens.*, vol. 11, no. 3, p. 254, 2019.
- [29] Z. Niu et al., "Towards safe physical human-robot interaction by exploring the rapid stiffness switching feature of discrete variable stiffness actuation," *IEEE Robot. Autom. Lett.*, vol. 7, no. 3, pp. 8084–8091, Jul. 2022.
- [30] O. Rioul and M. Vetterli, "Wavelets and signal processing," *IEEE Signal Process. Mag.*, vol. 8, no. 4, pp. 14–38, Oct. 1991.
- [31] D. Lim, D. Kim, and J. Park, "Momentum observer-based collision detection using LSTM for model uncertainty learning," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, 2021, pp. 4516–4522.
- [32] L. Alzubaidi et al., "Review of deep learning: Concepts, CNN architectures, challenges, applications, future directions," *J. Big Data*, vol. 8, pp. 1–74, Dec. 2021.
- [33] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 2980–2988.
- [34] J. Lin and L. Qu, "Feature extraction based on Morlet wavelet and its application for mechanical fault diagnosis," *J. Sound Vib.*, vol. 234, no. 1, pp. 135–148, 2000.
- [35] A. Akhan and O. K. Cura, "Time-frequency signal processing: Today and future," *Digit. Signal Process.*, vol. 119, Dec. 2021, Art. no. 103216.
- [36] R. G. Stockwell, L. Mansinha, and R. P. Lowe, "Localization of the complex spectrum: The S transform," *IEEE Trans. Signal Process.*, vol. 44, no. 4, pp. 998–1001, Apr. 1996.
- [37] M. Thill, W. Konen, and T. Bäck, "Time series anomaly detection with discrete wavelet transforms and maximum likelihood estimation," in *Proc. Int. Conf. Time Ser. (ITISE)*, vol. 2, 2017, pp. 11–23.
- [38] M. P. Wachowiak, G. S. Rash, P. M. Quesada, and A. H. Desoky, "Wavelet-based noise removal for biomechanical signals: A comparative study," *IEEE Trans. Biomed. Eng.*, vol. 47, no. 3, pp. 360–368, Mar. 2000.
- [39] T. Chen, X. Liu, B. Xia, W. Wang, and Y. Lai, "Unsupervised anomaly detection of industrial robots using sliding-window convolutional variational autoencoder," *IEEE Access*, vol. 8, pp. 47072–47081, 2020.
- [40] P. Liang, C. Deng, J. Wu, G. Li, Z. Yang, and Y. Wang, "Intelligent fault diagnosis via semisupervised generative adversarial nets and wavelet transform," *IEEE Trans. Instrum. Meas.*, vol. 69, no. 7, pp. 4659–4671, Jul. 2020.



Zhenwei Niu received the bachelor's and master's degrees in mechanical engineering from the China University of Petroleum (East China), Qingdao, China, in 2016 and 2019, respectively, and the Ph.D. degree in robotics from Khalifa University, Abu Dhabi, UAE, in 2023.

He is currently a Postdoctoral Fellow with Khalifa University. His research interests include physical human-robot interaction, compliant manipulation, robot teleoperation, shared autonomy, and robot learning.



Taimur Hassan (Senior Member, IEEE) received the Ph.D. degree in computer engineering from the National University of Sciences and Technology, Islamabad, Pakistan, in 2019.

He is currently an Assistant Professor with Abu Dhabi University, Abu Dhabi, UAE. His research interests lie in the fields of computer vision, medical imaging, and robotics.

Dr. Hassan was a recipient of many national and international awards throughout his career.



Mohamed Nassim Boushaki received the M.Sc. degree in robotics and automation and the Ph.D. degree in robotics from The University of Montpellier, Montpellier, France, in 2012 and 2016, respectively.

His Ph.D. works focus on the design optimization and control of Concentric Tube Robots for surgical applications. He later worked as a Postdoctoral Fellow on the integration of new surgical tools for the dVRK robot and the force reflection slave/master in teleoperation with the Scuola Superiore Sant'Anna, Pisa, Italy, in 2017/2019. He was a Postdoctoral Fellow with Khalifa University, Abu Dhabi, UAE, from 2019 to 2023, working on the control and applications of variable stiffness actuators, design optimization of exoskeletons, and modeling of cable-driven mechanism. He is currently a Project Lead, a Senior R&D of Robotics with the Innovation Department, Expleo Group Paris, Montigny-le-Bretonneux, France, developing a new mobile manipulator for different industrial applications.



Naoufel Werghi (Senior Member, IEEE) received the Ph.D. and Habilitation degrees in computer vision from the University of Strasbourg, Strasbourg, France, in 2000 and 2018, respectively.

He is currently a Full Professor with the Computer Science Department, Khalifa University, Abu Dhabi, UAE. He is the Theme Leader of the Artificial Intelligence and Big Data Pipelines Theme with the Cyber-Physical Security System Center, Khalifa University. His research interests span computer vision and machine learning, where he has been leading several funded projects related to biometrics, medical imaging, remote sensing, surveillance, and intelligent systems.



Irfan Hussain (Member, IEEE) received the B.E. degree in mechatronics engineering from Air University, Islamabad, Pakistan, in 2008, the master's degree in automatica and control technologies from Politecnico Di Torino, Turin, Italy, in 2012, the M.S. degree in mechatronics engineering from the National University of Sciences and Technology, Islamabad, in 2011, and the Ph.D. degree in robotics from the University of Siena, Siena, Italy, in 2017.

He currently serves as an Assistant Professor of Robotics and Mechanical Engineering with Khalifa University, Abu Dhabi, UAE. He is the author of one book, more than 70 articles, and three inventions. He was a Post Doctoral Researcher with Robotics Institute, Khalifa University of Science and Technology, Abu Dhabi, UAE. He also worked as a Post Doctoral Researcher with Siena Robotics and System Lab, Siena, Italy. He worked as a Research Assistant with Gyeongsang National University, Jinju, South Korea, from 2012 to 2013. He was a Visiting Researcher with Centro Ricerche Fiat, Cascina Mellano, Italy. He worked as an Assistant Manager of Engineering with Trojans, Islamabad, Pakistan, from 2008 to 2011. His research interests include marine robotics, healthcare robotics, and industrial robotics.

Dr. Hussain is an Associate Editor of *Proceedings of the Institution of Mechanical Engineers—Part C: Journal of Mechanical Engineering Science*. He is the Research Topic Editor on *Wearable Robots and Sensorimotor Interfaces: Augmentation, Rehabilitation, Assistance or Substitution of Human Sensorimotor Function* (ID 21096) and *Frontiers in Neurorobotics*. He is the Guest Editor, Special Issue “Design and Development of Vision-Based Tactile Sensors”—a special issue of *Sensors* (ISSN 1424-8220). He is an Associate Editor of IEEE/RAS ICRA-23, ICRA-22, ICRA-21, and RAS/EMBS BioRob-18.