Modeling Human Response to Robot Errors for Timely Error Detection

Maia Stiber[1], Russell Taylor[1,2], and Chien-Ming Huang[1]

***Abstract*— In human-robot collaboration, robot errors are inevitable—damaging user trust, willingness to work together, and task performance. Prior work has shown that people naturally respond to robot errors socially and that in social interactions it is possible to use human responses to detect errors. However, there is little exploration in the domain of non-social, physical human-robot collaboration such as assembly and tool retrieval. In this work, we investigate how people's organic, social responses to robot errors may be used to enable timely automatic detection of errors in physical human-robot interactions. We conducted a data collection study to obtain facial responses to train a real-time detection algorithm and a case study to explore the generalizability of our method with different task settings and errors. Our results show that natural social responses are effective signals for timely detection and localization of robot errors even in non-social contexts and that our method is robust across a variety of task contexts, robot errors, and user responses. This work contributes to robust error detection without detailed task specifications.**

I. Introduction

Unmanaged robot errors are harmful to human-robot collaboration. These errors present a safety concern, damage task performance, and erode users' trust and willingness to continue that cooperative partnership [1], [2]. The first step towards successful error management is timely error detection, which is key to enabling error mitigation and recovery [3]. While prior research has explored various methods for error detection, these methods tend to be task dependent and are not adaptable to other contexts or unexpected errors [4]. However, robot errors can be unexpected and independent of task, situation, and user [5].

To enable automatic detection of robot errors that may occur in varying contexts, we build on prior work [6] and explore how social signals may be an indicative source for automatic error detection. Though prior work has shown that human collaborators are likely to exhibit social signals due to errors' unexpectedness [7], most research has been situated in social contexts using a social robot (e.g., [8], [9], [10]). It is, however, unclear whether this social signal-based approach to robot error detection would work for a non-social robot (e.g., a manipulator) interacting with people in a non-social setting (e.g., task demonstration).

In this work, we explore how Action Units (AUs)—individual muscular movements as defined in the Facial Action Coding System [11]—may be used for automatic, timely detection of robot errors in non-social contexts. To this end, we first conducted a data collection study in which robot errors were intentionally produced to elicit natural social

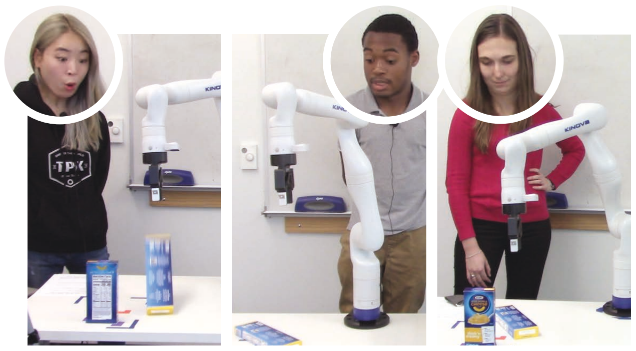

Fig. 1. In this work, we investigate the use of human response (action units) to detect unexpected robot errors in non-social interaction scenarios. Our approach works over a diverse array of tasks, errors, and responses.

responses from participants (Fig. 1). The collected data was then used to train a machine learning model for detection and temporal localization of robot errors. We additionally conducted a case study with a different set of participants experiencing different types of robot errors in various contexts to explore the generalizability of our approach. Our results indicate that AUs can be advantageously used for timely detection and temporal localization of unexpected robot errors and that our data-driven model can reasonably generalize its effectiveness to different settings and error types. Our work makes the following set of contributions:

- We show that it is possible to detect robot errors using social signals, especially AUs, in non-social, physical interaction scenarios.
- We develop a real-time human-in-the-loop error detection system, using the human collaborator as part of the detection process.
- Our approach shows its effectiveness beyond its training setting and generalizes to different contexts and errors.

II. Background and Related Work

Robotic systems often make unavoidable technical and unexpected errors. For example, Nourbakhsh *et al.* deployed three mobile robots in museums over five years and found that the mean time between failures was about 72 to 216 hours; it was difficult to increase the time between errors beyond that [12]. Robot errors harm robot performance, which impacts user trust [13] and the intensity of that negative effect is dependent on the error severity and quantity [14], [15]. Furthermore, robot errors are dependent on the individual's perception and how a robot's behavior deviates from the individual's mental model of the robot [16].

[1]Dept. of Computer Science, Johns Hopkins University, Baltimore, Maryland, USA {mstiber,rht,chienming.huang}@jhu.edu
[2] Life Fellow, IEEE

A. Social Responses to Robot Errors

People respond socially to robot errors [9], [17]. Moreover, humans exhibit more behaviors during error-occurring situations than error-free ones [17]. In particular, gaze [8], [18], [19], facial expressions [9], verbalizations [8], and body movements [7], [8], [20] have been shown to be common instinctive responses to robot errors. Much of this prior work was contextualized in social scenarios and relied on participants' existing expectations of robots during the interactions. Furthermore, the robots used in prior works typically were humanoid, which could have impacted the elicited responses [19]. Our preliminary work indicates that people exhibit social signals in response to robot errors even during physical human-robot interaction scenarios [6].

B. Error Detection Using Social Signals

To the best of our knowledge, there has been one system built utilizing social signals to detect social errors automatically. Kontogiorgos *et al.* showed that—with a collection of head, body, gaze, AU, and verbal tracking—it is possible to both automatically detect and classify certain types of social errors during conversational failures with a social robot [8]. However, little is known about how social signal responses to robot errors may enable automatic error detection in physical human-robot interactions, such as a robot manipulator committing technical task errors.

III. Modeling User Response to Robot Errors

In this work, we investigated whether it is possible to detect unexpected robot errors during physical interactions with a non-humanoid robot using facial action units (AUs). Our investigation consisted of (1) a data collection study to understand how people's AUs may manifest in response to technical robot errors in a non-social setting and (2) a data-driven model to allow for timely detection and localization of robot errors based on observed AUs.

A. Data Collection

Our data collection study was contextualized in a programming by demonstration (PbD) scenario in which participants provided task demonstrations through kinesthetic teaching using a Kinova Gen3 robot arm. This setup allowed participants to establish accurate mental models of the robot's capabilities and treated the participants as the task "experts," similar to real-world applications. Throughout the study, a real-time data collection system logged facial AUs exhibited by participants in response to robot movements and errors. In addition, the experimenter was not in the room with the participant during our study except to introduce the task and the PbD interface as the presence of an experimenter has been shown to increase implicit social signal quantity expressed [7].

1) Study Task: Before the actual data collection, participants completed a practice task, seeking to reduce any possible novelty effect associated with the robot moving, thereby leveling out participants' reactions to "normal" robot movements. The practice task, similar to the actual task, involved picking and placing wooden blocks with the robot executing the task without errors (see practice task in Fig. 2). Participants had the option to repeat the practice task until they were confident that they could program the robot and that the robot executed what they programmed.

For the actual data collection (see Fig. 2 for training and actual task workflow), participants were asked to "program" the robot to unpack two pasta boxes from a crate and place each of them in predefined locations on a table. The robot was "trained" with one of the boxes and executed the pick-and-place with both. The box the robot trained with was placed first without error before the robot "generalized" and performed a pick-and-place with the other box. During the execution of the second box's pick-and-place, we inserted a pre-programmed error—the robot dropping the box before it reached its goal—for the participant to observe and react to. Unbeknownst to the participants, their robot's training had no effect on the robot's behavior as its movements were pre-programmed by the experimenter.

2) Study Procedure: After consenting to partake in the study, participants were informed about the task and taught how to program the robot. The participants then conducted the practice task. Once done, the experimenter confirmed participants' confidence in programming the robot and introduced the actual task. The actual task included the drop error embedded in robot execution. After the task, participants were asked to fill out a questionnaire about whether they witnessed an error, its severity, and basic demographics. The experimenter then debriefed the participants and informed them of the involved deception. The study lasted about 30 minutes and was approved by our institutional review board (IRB). Participants were compensated $8 for study completion.

3) Study Systems: In support of our data collection, we developed two systems to allow the experimenter to (1) operate the robot as in a Wizard of Oz (WoZ) paradigm [21] and track the study progress from a different room and (2) collect video and AU data. The first system was a PbD interface which served as a simulated programming environment, provided a sequence of instructions to the participants, and allowed the experimenter to oversee the task status for WoZ operations. The second system logged user AUs while the robot carried out tasks. Below, we provide detailed descriptions of the two systems.

A Simulated Programming System. We created a 2D Unity application to increase the realism of our PbD scenario, hoping to lead participants to believe that they were actually programming the robot even though all of the robot's actions were actually pre-programmed. As a simulated programming system, the Unity application walked participants through a sequence of task steps. For the PbD portion of the task, the application provided the user a UI to add/delete waypoint and gripper commands as they kinethestically programmed the robot through the pick and place motions. On subsequent steps, the application informed participants about the task layout, robot movements, and then robot task execution. In addition, as the user progressed through the task steps, the

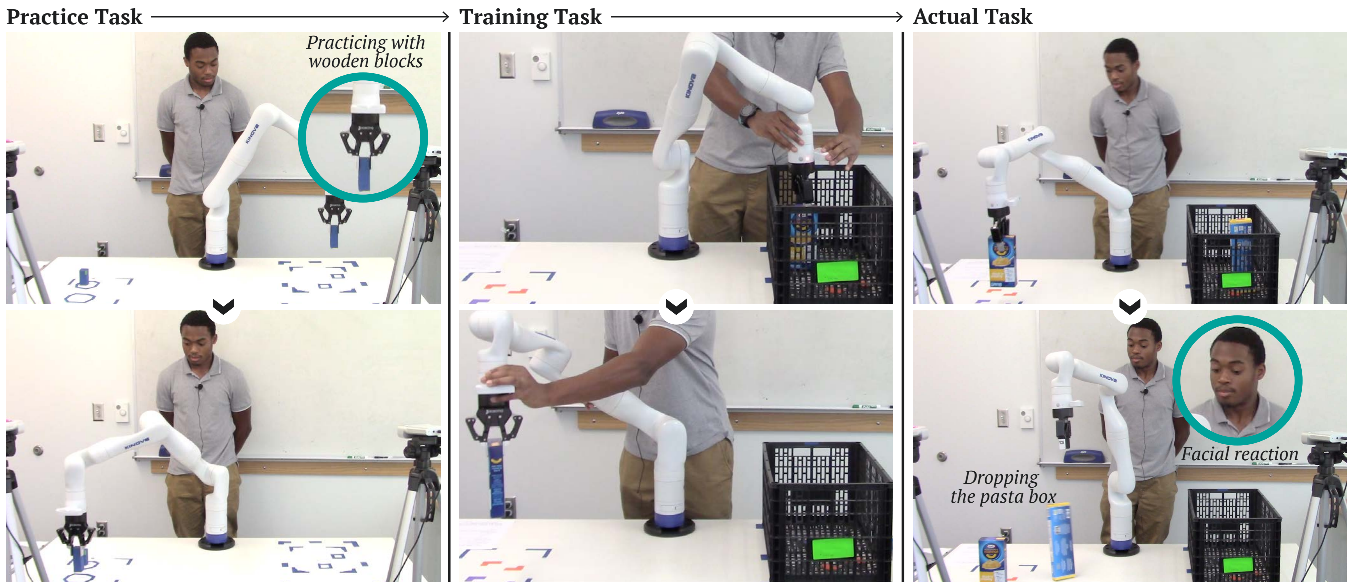

Fig. 2. Data collection study workflow. The practice task had the robot pick-and-place blocks from pre-designated positions, which was comparable to the actual task. Participants worked with the robot to train it in a pick-and-place task and then have it execute unpacking boxes from a crate. Participants kinethestically demonstrated to and programmed the robot by waypoint, using Box 1, the pick-and-place task needed to execute the actual task (Training Task). For the actual task, the robot executed a grocery unpacking task. The robot first unpacked Box 1 and then simulated the generalization to unpack Box 2, during which it dropped the box (pre-programmed error).

application sent the experimenter relevant signals for WoZ operations (e.g., when to trigger robot movements and turn on the data collection system).

A Data Collection System. We constructed a data collection system (Fig. 3) that took in live video (30fps) and processed it in real-time to log AU occurrence and intensity for each timestep, where a timestep was 1/3 second. Our system was built on top of Microsoft's Platform for Situation Intelligence (\psi) [22], which allowed us to synchronize multiple device inputs to a common time base, collect data, and add our real-time detection algorithm. We used OpenFace [23] to extract 17 AUs. To allow the participants to have full freedom of body and head movement around the robot during the study, our system took input from two cameras strategically placed such that the system could maximize facial detection confidence regardless of participant position. The system then simultaneously stored the recordings and piped the videos to two instances of our AU detection component, one for each camera. These components computed AU occurrences and intensities, along with facial detection confidence. At each time step, the data from the component with higher facial detection confidence was logged. If neither components' facial detection confidence was above 50%, then the AU calculation was considered inaccurate and zeros were logged for the AU metrics.

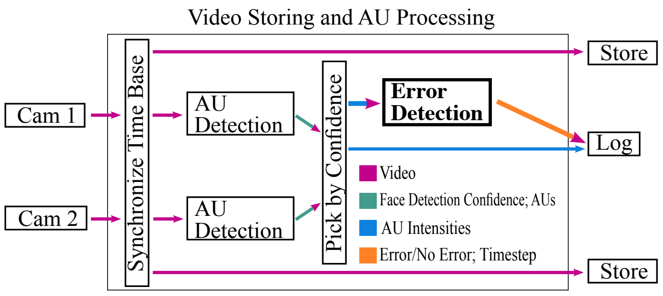

Fig. 3. Diagram of the real-time data collection platform. The bolded lines and box show how the error detection algorithm is integrated into the platform for the case study.

4) Participants: Twenty-three participants were recruited for the study. Among all the participants, two participants exhibited strong robot novelty effects despite the practice task (i.e., consistently reacting to the robot picking up an object) and two exhibited no visible reaction to the robot errors. We considered these four participants special cases and did not include them in training our detection algorithm; however, we included an evaluation of our algorithm on these cases. Consequently, the resulting training data consisted of 11 females and 8 males with ages ranging from 18 to 39 ($M = 23.4, SD = 4.6$). Participants' experience with robots and technology was assessed through three questions on a 5-point scale (Cronbach's $\alpha = 0.83$); they had low-medium prior experience, $M = 2.93, SD = 0.89$.

5) Dataset: In total, we amassed 25.05 min of video of which 3.03 min consisted of participants' responses to the drop error through this data collection study. All 23 participants said that they did witness an error; however, two of them had no visible reaction. On average, the drop error's severity was evaluated as medium-high severe, $M = 4.91, SD = 1.70$, with 1 being low and 7 is high severity. In addition, we had each participant rate to what extent they thought the error was their fault as a metric to determine participant's confidence in programming the

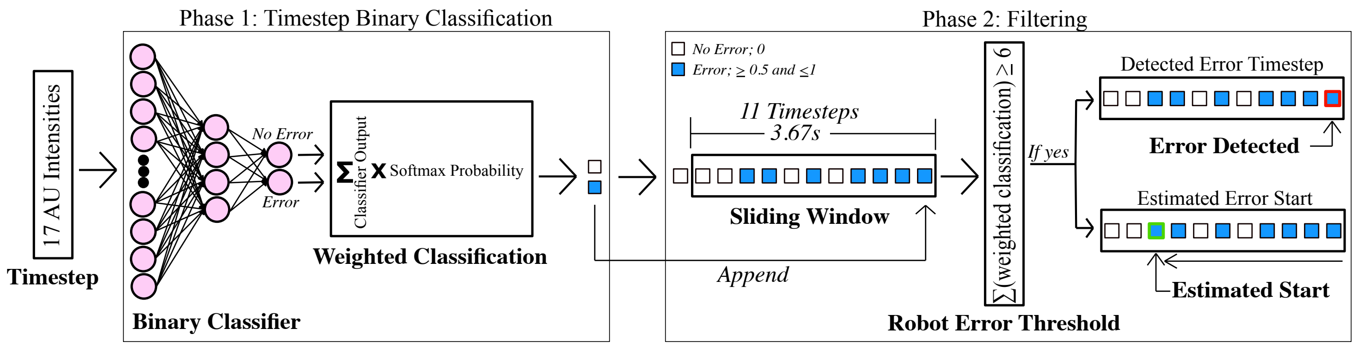

Fig. 4. Diagram of our detection algorithm. The input is the AU intensity of each timestep and the output, when the error is detected, is the timestep of error detection and the algorithm's estimated error start. Phase 1 consists of a 2-layer neural network that classifies each frame and the output of that is weighted by the softmax probability (classification confidence). The weighted classifications (no error: 0, error: ≥ 0.5 and ≤ 1) are the inputs for the sliding window of size 3.67s. If the window's convolution is greater than or equal to 6, then an error has been detected. The algorithm then traces back through the window to find the earliest timestep that is marked as an error and denoted as the estimated error start.

robot and perception of their mental model of the robot's capabilities. Across all participants, they minimally blamed themselves for the error, $M = 2.00, SD = 1.12$.

In addition, two independent coders annotated the videos frame by frame to help quantify human response to robot errors in terms of reaction time and duration (See Fig. 5 for a visual representation):

- **Human Reaction Time** (seconds). This metric quantifies how fast a participant reacted to the robot's error. It is defined as the difference between *user reaction start* and *perceived error start*. The user reaction start is defined as the time in which the first sign of any visible change in a participant's face happened. The perceived error start time is defined as when the coder was completely certain that the error was occurring, because some errors were slow to unfold. The average reaction time of the participants to the error was 0.5s ($SD = 0.68$).
- **Human Reaction Duration** (seconds). This metric quantifies how long a participant reacted to the robot's error. It is defined as the difference between *user reaction start* and *user reaction end*—when the participant's face/behavior returned to their norm as seen through video. The average reaction duration was 11.78s ($SD = 7.08$).

We further examined if AU intensities during error and no-error instances were significantly different. Since we had heavily imbalanced data for error and no-error instances and potential unequal variance in the corresponding intensities, we used Welch's t-tests to compare the AU intensities. Our analysis revealed that there was a statistically significant difference in intensities between error and no-error frames for 16 action units, except for AU_4 (brow lowerer), illustrating the discriminative potential of AUs in characterizing the manifestation of an unexpected robot error.

B. Modeling Action Units for Error Detection

We designed an algorithm that capitalizes on the discriminative potential of AUs to detect and localize unexpected robot errors. Rather than building a "complete" error detector, our goal in this work is to explore the possibility of automatic recognition of robot errors using facial cues in non-social settings. We, however, explored various modeling methods including bidirectional LSTM, anomaly detection using an autoencoder, and SVM. Ultimately, due to the small size of our dataset and large imbalance in error versus non-error instances, we chose a simple, yet sufficiently robust method. Our detection algorithm takes 17 AU intensities as inputs at each timestep and outputs the timestep number at which an error is detected. The detection algorithm consists of two phases: (1) weighted binary classification and (2) sliding window filtering (Fig. 4).

Phase 1: Weighted Binary Classification. At each time step, 17 AU intensities are fed into a two-layer neural network (input: 17, hidden: 4, output: 2) that conducts timestep-by-timestep classification of participant expressions and outputs a binary classification of error versus no error. In training this binary classifier, we accounted for the large imbalance in the number of error versus error-free timesteps by randomly undersampling the error-free timesteps every training epoch so that their counts matched those of the error timesteps. Moreover, each time step was treated as its own data point, and the temporal dimension was not preserved; this decision was made in part because of the small size of the training dataset. Two independent coders annotated ground truth for each time step. The classification output from the network was then weighted by the classification confidence (softmax probabilities) and then summed to minimize the impact of out-of-distribution samples and misclassifications; prior work has shown this weighted approach to be effective [24].

Phase 2: Sliding Window Filtering. To reduce spurious error classifications (e.g., any quick movements the participants made such as twitch), we employed a sliding window filter after the weighted binary classification. The sliding window was $3.67s$ long which translates to 11 timesteps. As every new weighted classification (ranging from 0 to 1 with 0 being no error) at each timestep was outputted, the window slid

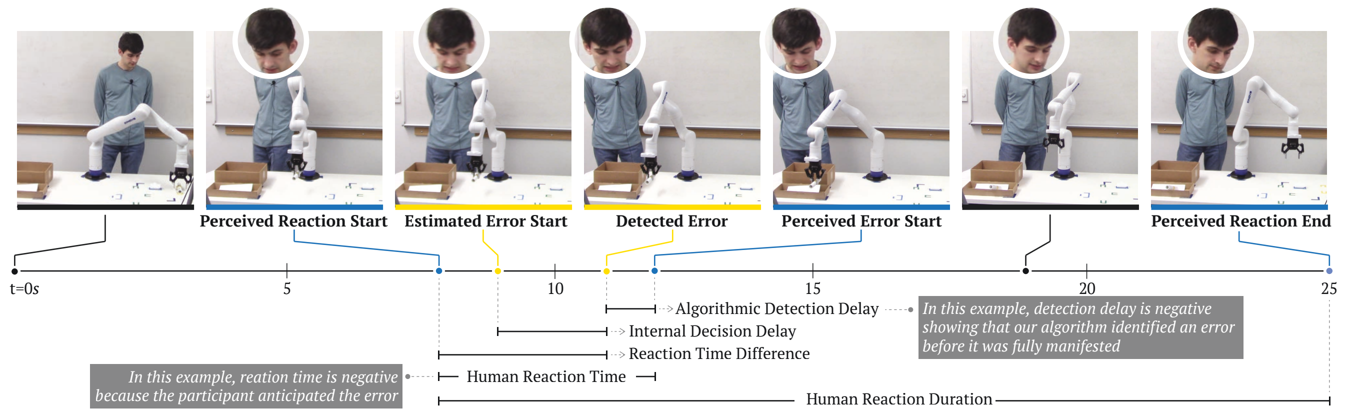
Perceived Reaction Start
Estimated Error Start
Detected Error
Perceived Error Start
Perceived Reaction End
t=0s
5
10
15
20
25
Algorithmic Detection Delay
Internal Decision Delay
Reaction Time Difference
Human Reaction Time
Human Reaction Duration
In this example, detection delay is negative showing that our algorithm identified an error before it was fully manifested
In this example, reation time is negative because the participant anticipated the error

Fig. 5. Case study example of one participant's reaction to and our algorithm's error detection of a sorting error. The participant reacted to the error before the coder's perceived error start and so our algorithm was able to detect the error before then. The blue represents the coded timesteps (left to right): start reaction, perceived error start, and end reaction. The yellow indicates (left to right): our algorithm's estimated error start and robot error detected.

over by one timestep to include the most recent output. If the convolution of the window was at least 6 (empirically tuned), then the algorithm outputted that a robot error had occurred.

When an error was detected, we considered the *estimated error start* as the earliest timestep in the window that was classified as an error and the *detected error timestep* as the newest timestep of the latest error classification. If the *estimated error start* or *detected error timestep* was within one timestep of a previously *detected error timestep*, then the present error was considered part of the former error. We integrated this two-phase detection algorithm into our data collection system (Fig. 3) with AU intensities at each timestep inputted into the algorithm in real time. In production, our detection system could be run in real-time.

C. Model Evaluation

Typical ML metrics, such as accuracy and F1-score, are not representative of this algorithm's performance for two reasons: (1) errors are sparse in our dataset and (2) the classification accuracy of individual time step classification is not as important as sequential clustering of error time steps. Therefore, in evaluating our algorithm performance, we focused on the following metrics (See Fig. 5 for a visual representation):

- **Algorithmic Detection Delay** (seconds). This metric measures the root mean squared error of the delay between *algorithmic error detection* and *perceived error start* (as annotated by independent coders).
- **Reaction Time Difference** (seconds). This metric represents the robot mean squared error of the time difference between *algorithmic error detection* and coded *user reaction start*.
- **Internal Decision Delay** (seconds). This metric computes the average difference between the algorithm's *detected error timestep* and *estimated error start*.
- **False Positive Rate**. This metric quantified the average number of false positives logged per trial. A false positive is defined as when the algorithm's output timesteps (detected error timestep and estimated error start) do not overlap with the coded participant's reaction to the robot error.

We used leave-one-out cross validation with the collected data to evaluate the family of models used (feasibility of detecting errors), which allowed for a comprehensive evaluation of the model and also tested for overfitting due to the small size of the data sets. Our error detection and localization algorithm had an average *algorithmic detection delay* of 3.25s and a *reaction time difference* of 3.10s. The system's *internal decision delay* was on average 2.37, $SD = 0.50$. In addition, the *false positive rate* was 0.61 per trial, $SD = 0.78$, and false negative rate was 0 per trial. Table I summarizes results of our system evaluation.

IV. Real-time Error Detection and Model Generalization

To explore how well our method for real-time error detection generalized to different tasks and error types, we ran a case study evaluating our algorithm against different tasks and errors than the data collection study.

A. Task and Procedure

We contextualized our case study in a sorting task, where participants demonstrated to the robot how to sort PVC pipes and joints into differently labeled bins. Similar to the data collection study, participants were given a practice task (sorting blocks by color) to familiarize themselves with the robot and the programming system. The task allowed participants to establish an understanding of the robot's capabilities to pick and place objects. In addition to demonstrating pick-and-place operations, we had the participants "teach" the robot the concept of sorting by having them show the robot's wrist cameras the object and placing it in the appropriate bin. The programming and teaching did not effect the robot's behavior; the robot errors were again pre-programmed.

For the main task, each participant experienced three different errors in three interaction rounds, where each round consisted of sorting three different objects into bins. In each

TABLE I

SUMMARY OF RESULTS FOR THE TWO COLLECTED DATASETS. THE GRAYED ROWS IN THE TABLE SHOW AN ADDITIONAL ANALYSIS OF THE CASE STUDY DATASET AS BROKEN DOWN BY ERROR TYPE.

		Human Metrics		Algorithmic Metrics				
		Human Reaction Time	**Human Reaction Duration**	**Algorithmic Detection Delay**	**Reaction Time Difference**	**Internal Decision Delay**	**False Positive Rate**	**False Negative Rate**
Dataset	Model Training Dataset	0.50 ± 0.68	11.78 ± 7.08	3.25	3.10	2.37 ± 0.50	0.61 ± 0.78	0
	Case Study Dataset	-0.02 ± 2.21	10.09 ± 5.57	5.98	5.95	2.64 ± 0.67	0.40 ± 0.51	0.27 ± 0.44
	↓ *breakdown analysis*							
Error Type	Physical	0.33 ± 2.16	7.53 ± 4.14	2.17	2.22	2.67 ± 0.94	0.50 ± 0.71	0.40 ± 0.55
	Concept	-0.73 ± 3.21	8.93 ± 4.67	3.90	3.92	2.00 ± 0.00	0.25 ± 0.50	0.20 ± 0.45
	Generalization	0.47 ± 0.99	13.80 ± 6.52	8.48	8.41	2.74 ± 0.69	0.50 ± 0.58	0.20 ± 0.45

TABLE II

THREE ERROR CATEGORIES EXPERIENCED BY PARTICIPANTS DURING THE CASE STUDIES OF WHICH EACH HAS TWO POSSIBLE ROBOT ERRORS.

Error Type	Description
Physical	Dropping the object
	Hitting the sorting bin with the object
Concept	Sorting the object incorrectly
	Misplacing an object outside the target bin
Generalization	Missing a pick by not extrapolating to object
	Missing a pick with wrong gripper orientation

round, the participant programmed the robot with only one of the objects, and the robot "generalized" to the other two objects when executed. In order to reset the participant's mental models between each round, we had them "add" onto the original robot program by programming the robot again with the object it just had made a mistake with. That object was then the first sorted by the robot in the following round to confirm successful program (i.e., reset mental model).

Participants were randomly assigned errors from three categories (*physical*, *concept*, and *generalization* errors), one from each in a random order. Table II lists the possible errors the participants could witness, the majority differing from the error seen in the data collection study. These error categories were chosen to represent the three aspects of the task: the robot was "programmed" for a pick-and-place (physical), was taught the notion of sorting into bins (concept), and extrapolated those two lessons towards new objects (generalization). All of the errors, except for the drop error, are considered to be *predictable* errors meaning that the errors initially make slow perceptible changes in trajectory before they fully unfold.

For this case study, the error detection algorithm, only trained on the data from the data collection study, was integrated into the data collection system (Fig. 3); the AUs were directly piped to the detection algorithm to allow for indication of when robot errors happened based on human reaction in real time. The overall procedure for this case study was comparable to that in the data collection study. This study lasted around one hour and participants were compensated $16.

B. Participants

Five participants (three female and two male) were recruited for this case study. Their ages ranged from 22 to 29 ($M = 25.2, SD = 3.11$) and had medium prior experience with technology and robots, $M = 3.00, SD = 0.85$.

C. Case Analysis

The case study dataset consisted of 23.47 min of robot interaction video, where 2.61 min involved facial reactions to errors. The average error severity (1 is low and 7 being high severity) for all errors rated by the participants was medium severe ($M = 4.33, SD = 1.60$) where physical ($M = 4.00, SD = 2.00$), concept ($M = 3.80, SD = 1.48$) and generalization ($M = 4.60, SD = 0.55$) errors were all considered medium severe. All participants confirmed that they saw the robot make an error and attributed low blame (1 being low and 7 is high blame) to themselves, $M = 3.00, SD = 1.85$.

On average, the *human reaction time* was 0.02s before coded *perceived error start*, $SD = 2.21$, indicating that the participants reacted before the error was fully manifested. The average *human reaction duration* was 10.09s, $SD = 5.57$. When we look at participants' reactions to the different error types (Table I), we see that, on average, participants reacted to concept errors before the *perceived error start* and the *human reaction duration* for the generalization errors were longer than that of for physical and concept errors. Furthermore, we ran the best resulting model, only trained on the data collection dataset, with the case study data as a test set (generalizability of use cases). The algorithm detected, in real-time, errors with an *algorithmic detection delay* of 5.98s and a *reaction time difference* of 5.95s. It had an *internal decision delay* of 2.64s, $SD = 0.67$. In addition, the *false positive rate* was 0.40, $SD = 0.51$, and a false negative rate of 0.27, $SD = 0.44$. If we were to separate the model's evaluation by error type, the results reveal that the algorithm was able to timely detect the physical and concept errors but was delayed about twice as long as for

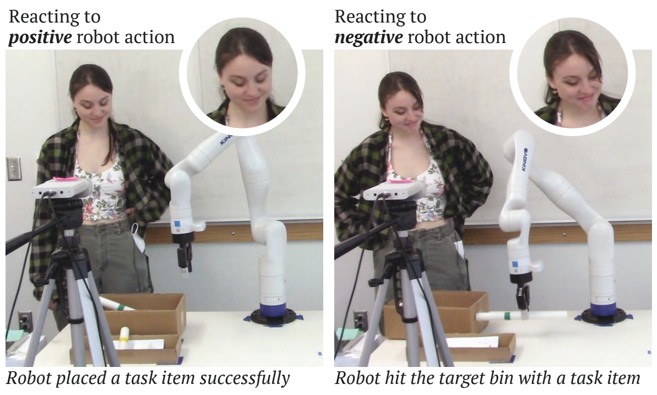
Reacting to
positive robot action
Reacting to
negative robot action
Robot placed a task item successfully
Robot hit the target bin with a task item

Fig. 6. Example of similar reaction to positive and negative robot actions. The left image (positive robot action) shows the robot moving away from a bin after it properly and correctly placed a pipe in it. The right image (negative robot action) shows the robot making an error by pushing the same bin with the same pipe.

detecting generalization errors. See Table I for a summary of the statistics for this dataset.

V. Discussion

In this paper, we demonstrate leveraging natural human response to unexpected robot errors to automatically detect them in real-time. Our results show that it is possible to detect and temporally localize errors within reasonable accuracy and timeliness using AUs and that this algorithm can be generalized to a different task and to different types of errors. More importantly, this approach is not person-specific.

A. Error Detection Through Social Responses

Robot errors likely and immediately elicit social responses. Of all 28 participants in both studies, only two did not have visible reactions to errors. In addition, participant reaction times were, on average, within a second of the perceived error start for both studies. By relying on implicit, social reactions to recognize robot errors, our approach does not lose noticeable time to "wait" for the user's explicit responses, as evidenced by *algorithmic detection delay* and *reaction time difference*, which are within 0.15s of each other. On the flip side, our approach in principle is only as fast as detectable social responses.

However, if the error is predictable by the human—namely if the error causes gradual noticeable deviations in behavior/trajectory before it fully unfolds—it is possible for the person to observe and react before the *perceived error start*. In our case study, some errors were predictable, and when examining participants' reaction times to predictable vs. non-predictable errors, we found that, on average, they reacted before the *perceived error start* for predictable errors, $M = -0.03, SD = 2.38$, and after for non-predictable errors $M = 0.33, SD = 0$. Thus, the algorithm has the ability to take advantage of the early reaction and indicate that the robot is making a mistake before the *perceived error start*. Fig. 5 illustrates an example of such a detection.

B. Reliability, Customizability, and Generalizability

All of our results were consistent despite the fact that no two participants reacted the same, even when the errors were the same. Localization and detection of the robot error for our data collection study was 3.25s delayed; for the case study, the *algorithmic detection delay* was 2.73s longer as compared to the data collection study. Nevertheless, our approach was able to reliably detect different types of errors in situations in which it was not trained.

We explored this discrepancy further by training the algorithm on one of the trials for each person in the case study, tailoring it to each person, before testing it on the remaining trials for that person. This fine-tuning improved *algorithmic detection delay* and *reaction time difference* by 0.74s and 0.60s, respectively. These results are similar to what we see in human-human interaction where people have a harder time (are less accurate) decoding meaning from facial expressions with strangers than with friends [25]. Consequently, this finding shows the potential for improving performance through fine-tuning detection, which would be useful for longer-term interactions with the same person.

Our algorithm detects large changes in a person's facial reactions from their norm and then makes the assumption that those changes are due to errors. This is, however, not always the case. Indeed, we observed an average *false positive rate* of about 0.51 per trial over both studies. All of the false positives detected were a consequence of the participants reacting to different robot actions, such as placing an object correctly or relief from correct operation after an error. The algorithm cannot tell the difference between a reaction due to a positive or negative robot action; *it needs context* (Fig. 6). In human-human interaction, facial expressions are inherently ambiguous without context [26] and the same facial expressions in different cultures could mean different things [27]. However, it is important to note that some visible reactions are not detected as errors by the algorithm.

C. Limitations

One limitation of our approach is that detection can only occur if the user reacts and within a reasonable time frame. We tested our algorithm on the two trials from the data collection study where the participants had no reaction, and the algorithm failed to detect the errors. On the opposite side, if the user reacted to everything (mostly due to the novelty effect), then the algorithm would constantly generate false positives, as shown by testing our algorithm on two trials from the data collection study where the participants exhibited strong novelty effects (false positive rate: $M = 2, SD = 0$). In addition, the algorithm could not discern differences between reactions to positive robot actions and negative robot actions (errors). Another limitation is related to reliable facial detection. For example, if the individual's face is obscured (e.g., by their hand) then our approach would not be able to detect AUs. Moreover, we found that if the person were to remove the obstruction, then there would be a jump in AU intensity, triggering a false positive. Finally, we had limited datasets; therefore, the algorithm was not necessarily trained with a fully comprehensive array of human responses to robot errors. In addition, we were not able to implement this approach using other modeling methods that require larger training sets.

D. Implications for Human-Robot Collaboration

Our findings illustrate the feasibility of detecting various types of errors in different tasks in real-time using natural responses during human-robot interaction. This work is the first step towards robust error detection, key to successful error management. While our approach can detect errors across different users, it also has the potential to be tailored to an individual for more accurate error detection and localization.

Future work will focus on fully integrating the algorithm into an autonomous robotic system that will perform error recovery after detection. In addition, we look to collect more data, improve the detection algorithm by exploring different social signals, and add context to improve the false positive rate. We should consider how this social signal-based approach may be used with other error detection methods to improve detection reliability and flexibility. We also want to explore automatic classification of error severity through social signals as that could provide information for appropriate recovery strategies and help us understand the impact of those errors.

ACKNOWLEDGMENT

We thank Kaitlynn Pineda for performing the video coding for the errors and reactions. This work was supported by the Johns Hopkins University Institute for Assured Autonomy and the National Science Foundation award #2143704.

REFERENCES

[1] D. J. Brooks, M. Begum, and H. A. Yanco, "Analysis of reactions towards failures and recovery strategies for autonomous robots," in *2016 25th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*, 2016, pp. 487–492.

[2] M. Salem, G. Lakatos, F. Amirabdollahian, and K. Dautenhahn, "Would you trust a (faulty) robot? effects of error, task type and personality on human-robot cooperation and trust," in *Proceedings of the Tenth Annual ACM/IEEE International Conference on Human-Robot Interaction*, 2015, pp. 141–148.

[3] H. Yasuda and M. Matsumoto, "Psychological impact on human when a robot makes mistakes," in *2013 IEEE/SICE International Symposium on System Integration*, 2013.

[4] S. Honig and T. Oron-Gilad, "Expect the unexpected: Leveraging the human-robot ecosystem to handle unexpected robot failures," *Frontiers in Robotics and AI*, vol. 8, 2021.

[5] R. Gideoni, S. Honig, and T. Oron-Gilad, "Is it personal? the impact of personally relevant robotic failures (perfs) on humans' trust, likeability, and willingness to use the robot," *arXiv preprint arXiv:2201.05322*, 2022.

[6] M. Stiber and C.-M. Huang, "Not all errors are created equal: Exploring human responses to robot errors with varying severity," in *Companion Publication of the 2020 International Conference on Multimodal Interaction*, 2020, p. 97–101.

[7] M. Giuliani, N. Mirnig, G. Stollnberger, S. Stadler, R. Buchner, and M. Tscheligi, "Systematic analysis of video data from different human-robot interaction studies: A categorisation of social signals during error situations," *Frontiers in Psychology*, vol. 6, 2015.

[8] D. Kontogiorgos, M. Tran, J. Gustafson, and M. Soleymani, "A systematic cross-corpus analysis of human reactions to robot conversational failures," in *Proceedings of the 2021 International Conference on Multimodal Interaction*, 2021, pp. 112–120.

[9] P. Trung, M. Giuliani, M. Miksch, G. Stollnberger, S. Stadler, N. Mirnig, and M. Tscheligi, "Head and shoulders: Automatic error detection in human-robot interaction," in *ACM International Conference on Multimodal Interaction*, 2017.

[10] N. Mirnig, G. Stollnberger, M. Miksch, S. Stadler, M. Giuliani, and M. Tscheligi, "To err is robot: How humans assess and act toward an erroneous social robot," *Frontiers in Robotics and AI*, vol. 4, 2017.

[11] P. Ekman and W. V. Friesen, *Facial action coding system: Investigator's guide*, 1978.

[12] I. R. Nourbakhsh, C. Kunz, and T. Willeke, "The mobot museum robot installations: A five year experiment," in *Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003)*, vol. 4, 2003.

[13] K. E. Schaefer, J. Y. C. Chen, J. L. Szalma, and P. A. Hancock, "A meta-analysis of factors influencing the development of trust in automation: Implications for understanding autonomy in future systems," *Human Factors*, vol. 58, no. 3, 2016.

[14] A. Rossi, K. Dautenhahn, K. L. Koay, and M. L. Walters, "How the timing and magnitude of robot errors influence peoples' trust of robots in an emergency scenario," in *International Conference on Social Robotics*. Springer, 2017, pp. 42–52.

[15] B. M. Muir and N. Moray, "Trust in automation. part ii. experimental studies of trust and human intervention in a process control simulation," *Ergonomics*, vol. 39, no. 3, 1996.

[16] A. Rossi, K. Dautenhahn, K. L. Koay, and M. L. Walters, "Human perceptions of the severity of domestic robot errors," in *Social Robotics*. Springer International Publishing, 2017, pp. 647–656.

[17] D. E. Cahya, R. Ramakrishnan, and M. Giuliani, "Static and temporal differences in social signals between error-free and erroneous situations in human-robot collaboration," in *International Conference on Social Robotics*, 2019, pp. 189–199.

[18] R. M. Aronson and H. Admoni, "Gaze for error detection during human-robot shared manipulation," in *RSS Workshop: Towards a Framework for Joint Action*, 2018.

[19] D. Kontogiorgos, A. Pereira, B. Sahindal, S. van Waveren, and J. Gustafson, "Behavioural responses to robot conversational failures," in *Proceedings of the 2020 ACM/IEEE International Conference on Human-Robot Interaction*. Association for Computing Machinery, 2020, pp. 53–62.

[20] R. Bovo, N. Binetti, D. P. Brumby, and S. Julier, "Detecting errors in pick and place procedures," in *ACM Conference on Intelligent User Interfaces*, 2020.

[21] L. D. Riek, "Wizard of oz studies in hri: a systematic review and new reporting guidelines," *Journal of Human-Robot Interaction*, vol. 1, no. 1, pp. 119–136, 2012.

[22] D. Bohus, S. Andrist, A. Feniello, N. Saw, M. Jalobeanu, P. Sweeney, A. L. Thompson, and E. Horvitz, "Platform for situated intelligence," 2021.

[23] T. Baltrušaitis, P. Robinson, and L.-P. Morency, "Openface: an open source facial behavior analysis toolkit," in *2016 IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2016, pp. 1–10.

[24] D. Hendrycks and K. Gimpel, "A baseline for detecting misclassified and out-of-distribution examples in neural networks," *arXiv preprint arXiv:1610.02136*, 2016.

[25] R. W. Sternglanz and B. M. DePaulo, "Reading nonverbal cues to emotions: The advantages and liabilities of relationship closeness," *Journal of Nonverbal Behavior*, vol. 28, no. 4, 2004.

[26] R. R. Hassin, H. Aviezer, and S. Bentin, "Inherently ambiguous: Facial expressions of emotions, in context," *Emotion Review*, vol. 5, no. 1, 2013.

[27] C. Crivelli, S. Jarillo, J. A. Russell, and J.-M. Fernández-Dols, "Reading emotions from faces in two indigenous societies." *Journal of Experimental Psychology: General*, vol. 145, no. 7, 2016.