IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 20 February 2023, accepted 5 April 2023, date of publication 15 May 2023, date of current version 23 May 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3276297

RESEARCH ARTICLE

CLUE-AI: A Convolutional Three-Stream Anomaly Identification Framework for Robot Manipulation

DOGAN ALTAN[1,2], (Member, IEEE), AND SANEM SARIEL[1], (Member, IEEE)
[1]Artificial Intelligence and Robotics Laboratory, Faculty of Computer and Informatics Engineering, Istanbul Technical University, 34469 Istanbul, Turkey
[2]Department of Validation Intelligence for Autonomous Software Systems, Simula Research Laboratory, 0164 Oslo, Norway

Corresponding author: Dogan Altan (daltan@itu.edu.tr)

This research was supported by Istanbul Technical University Scientific Research Projects (ITU-BAP) under Grant 40240 and the Scientific and Technological Research Council of Turkey (TUBITAK) under Grant 119E-436.

ABSTRACT Despite the great promise of service robots in everyday tasks, many roboethics issues remain to be addressed before these robots can physically work in human environments. Robot safety is one of the essential concerns for roboethics which is not just a design-time issue. It is also crucial to devise the required onboard monitoring and control strategies to enable robots to be aware of and react to anomalies (i.e., unexpected deviations from intended outcomes) that arise during their operations in the real world. The detection and identification of these anomalies is an essential first step toward fulfilling these requirements. Although several architectures have been proposed for anomaly detection; identification has not yet been thoroughly investigated. This task is challenging since indicators may appear long before anomalies are detected. In this paper, we propose a ConvoLUtional threE-stream Anomaly Identification (CLUE-AI) framework to address this problem. The framework fuses visual, auditory and proprioceptive data streams to identify everyday object manipulation anomalies. A stream of 2D images gathered through an RGB-D camera placed on the head of the robot is processed within a self-attention-enabled visual stage to capture visual anomaly indicators. The auditory modality provided by the microphone placed on the robot's lower torso is processed within a designed convolutional neural network (CNN) in the auditory stage. Last, the force applied by the gripper and the gripper state is processed within a CNN to obtain proprioceptive features. These outputs are then combined with a late fusion scheme. Our novel three-stream framework design is analyzed on everyday object manipulation tasks with a Baxter humanoid robot in a semi-structured setting. The results indicate that CLUE-AI achieves an f-score of 94%, outperforming the other baselines in classifying anomalies.

INDEX TERMS Cognitive robots, robot manipulation, robot safety, anomaly identification, robot learning.

I. INTRODUCTION

Safety is of great importance when robots work in human environments [1], [2]. They must operate without any potential damage to humans and their surroundings in such settings [3]. Even though robots are designed by applying safety engineering methods [4], various sources of uncertainty may lead to unexpected outcomes, i.e., anomalies, in the real world, which must be detected and identified for robust execution. This work addresses the anomaly identification problem, which asks for classifying a detected anomaly to enhance the situation awareness capabilities of robots.

Detecting and identifying anomalies is essential, and diagnostic procedures are needed to recognize anomalies and recover from them [5], [6], [7]. The first phase of a diagnostic procedure is to indicate an anomaly, called *anomaly detection*. The classification of the anomaly type follows anomaly detection. This procedure is called *anomaly identification*. After identifying the anomaly, the robot should apply recovery actions to return to the nominal state and achieve the task. This is called *recovery*. Our main motivation in this research is the anomaly identification task, where the robot classifies

The associate editor coordinating the review of this manuscript and approving it for publication was Lorenzo Ciani.

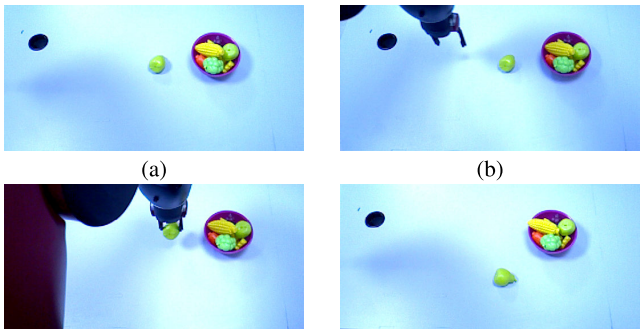


FIGURE 1. An anomalous execution from the viewpoint of a robot with a parallel gripper hand. The task is to place the plastic pear into the container. The action fails since the container is full of other objects.

anomaly cases after they are detected. Such a task is vital, as being aware of the occurred anomaly type enables the robots to come up with effective recovery plans by associating anomaly contexts with upcoming plans [8].

In this study, we consider robot manipulation anomalies as deviations from rules, specifications, or expectations. Based on this definition, unexpected situations caused by nature, human interventions, or the robot during manipulation are considered anomalies. Anomaly definition also includes violations of expected operational outcomes, particularly failures encountered due to improper computation of grasp, push, place placements, or unsafe executions. Moreover, some anomaly cases may appear, although the robot operates safely in a normal situation. A sample case is illustrated in Figure 1 from the view of the robot. In the illustrated case, the robot stands behind a table on which some objects to be manipulated are located. The robot is tasked with placing an object (a plastic pear) into a container full of other plastic objects (Figure 1a). First, the robot reaches the pear by its arm (Figure 1b), picks it up, and tries to place it into the container (Figure 1c). However, since the container is almost full of objects (which is not an anomalous situation at that particular time frame), the plastic pear bounces back and falls onto the table during a safe place-in-container operation (Figure 1d). Following this step, the robot should detect this anomaly for further inference about the case. The underlying reason for this anomaly is that the robot cannot assess the depth of the container and predict that the placement operation will fail. At this point, the anomaly identification procedure is expected to ensure that the robot is aware of the fact that a place action is failed due to the situation of the container. This is the key to taking the necessary precautions for handling future place-in-container tasks on the same container or in a similar case.

Incorporating sensory readings is the key to an effective anomaly identification procedure. Indeed, a single sensor modality may not be sufficient to accurately identify different cases. The use of multiple sensory sources may be complementary to an effective identification process [9]. Yet this information should be combined and fused effectively to identify the occurred anomaly type during task execution.

In this paper, we propose a convolutional three-stream anomaly identification (CLUE-AI) framework that processes multimodal data within distinct stages to classify anomaly cases. In our previous work [7], we presented a symbol-based anomaly identification framework that adopts an early fusion scheme of multimodal data to identify anomalies. Different from the earlier work [7], the CLUE-AI framework does not require any hand-crafted features, domain symbols, or task-related features, and it adopts a late fusion mechanism to fuse attention-enabled visual, auditory and proprioceptive streams that are processed within distinct stages. The CLUE-AI framework takes into account visual, auditory and proprioceptive sensor modalities to reveal anomaly types during execution. The visual data stream is processed, taking into account the contribution of each image, by enabling a self-attention mechanism. Auditory and proprioceptive data streams are also processed within distinct CNN designs to extract anomaly-related features. A late fusion scheme combines the outputs of these stages to capture anomaly indicators obtained in distinct sensor modality stages. The CLUE-AI framework is evaluated on real-world scenarios performed by a Baxter robot. A comparison of the framework with other baselines is presented, and the performance of the framework is analyzed for different feature extraction techniques. An ablation study is also presented to analyze the contribution of each sensory modality and the attention mechanism in classifying the anomaly type.

The main contributions of this study are threefold:

- We propose a novel convolutional three-stream anomaly identification framework, namely CLUE-AI, that incorporates visual modality together with auditory and proprioceptive modalities to capture anomaly indicators for object-related perceptions. These multimodal data are processed in different stages to identify everyday object manipulation task anomalies.
- We deal with cases where anomaly indicators appear long before [7] an anomaly is detected with a self-attention-enabled framework design.
- We address the identification of anomaly cases that arise during object manipulation episodes in semi-structured environments. In such environments, task specifications and/or object placements are not fixed or stable, unlike in the case of engineered settings.

This paper is organized as follows: First, the literature on anomaly identification is summarized. This section is followed by the proposed CLUE-AI framework. Later on, the evaluation of the framework is presented, which includes an ablation study and an experimental evaluation of the presented framework. Finally, the paper is concluded with potential future directions.

II. LITERATURE REVIEW

Anomaly detection and identification have been widely investigated in the literature [10], [11], [12], [13] and several taxonomies of failures that could occur in task environments are

presented [14], [15], [16]. This section presents a summary of the related work on the anomaly identification literature.

Anomaly identification can be achieved by using hypothesis-based methods by either maintaining cost-attached hypotheses in a hypothesis pool and analyzing inconsistencies among them [17] or comparing the differences between the theory and the model of the world [18]. A cooperative diagnostic method may also be presented for a multi-robot domain to diagnose failures where robots help each other to do so [19].

Clustering-based methods are also investigated to identify failure cases. In one study [20], a multi-level sensor fusion method is investigated to detect and identify abnormal cases by clustering the outputs of the sensors. In another clustering-based method, a global fuzzy c-means clustering algorithm is used to maintain clusters to identify failures [21]. Combining unsupervised learning with supervised learning is also studied in the literature. In a clustering-based method [22], outliers in the data are first detected with unsupervised learning techniques, which are followed by differentiating between special modes and anomalies with supervised techniques.

In the literature, particle filter-based (PF) [23], Kalman filter-based [24] and hidden Markov model-based (HMM) [9], [25], [26], [27] methods are widely used for handling anomalies. Various types of HMMs are used for detecting and identifying failures that occur during assistance tasks [28]. In one study [29], HMMs are studied with gradient analysis to identify anomaly cases. Bayesian filters are also studied to analyze failures [30]. A hierarchy for HMMs and PFs is proposed to isolate failure cases in a mobile robot setting [25], [26], [31]. In another work [32], a probabilistic method is proposed to predict failure cases for humanoid robots in hazardous environments by associating risks with related actions. Another study uses Hierarchical Dirichlet Process HMMs [33] to identify and classify anomalies that arise during collaborative kitting tasks. Yet another work presents a tensor voting-based method combined with support vector machines (SVMs) for classifying surface anomalies using 3D point cloud data [34].

Deep learning-based methods are also applied to handle anomaly cases. An autoencoder-based method that uses a stacked denoising autoencoder (SDA) is proposed [35] to identify the health state of rotary machines. Transfer learning (TL) is also studied to diagnose industrial failures. A deep transfer learning (DTL) method is proposed to handle motor bearing failures [36]. The model of task execution can be constructed with convolutional neural networks (CNNs) to identify anomalies [37]. In another work [38], recurrent neural networks (RNNs) are used to detect anomalies that arise in the Internet of Things (IoT) networks. In another work, anomalies are detected with a multimodal sensor fusion-based deep neural network design [6]. Another autoencoder-based method adopts a variational long short-term memory (LSTM) autoencoder to detect anomalies in robot-assisted feeding tasks [39]. In yet another study, multilayer perceptrons (MLPs) combine the temporal dependencies in multimodal data captured by HMMs, and the convolutional features are extracted from visual data by VGG16 to classify anomaly types in the domain of human-robot interaction in human feeding scenarios [40]. Different from this study, we address everyday object manipulation anomalies that arise due to uncertainties in perception and/or execution failures. Moreover, instead of extracting temporal features with HMMs and temporal pyramid pooling [41] to capture unexpected trends in hand-engineered features (i.e., sound energy, spoon-mouth distance, desired spoon displacement, force, etc.), we employ specific CNN-based designs for each distinct multimodal sensory data type to associate anomaly indicators with anomaly types without any preprocessing efforts. Furthermore, in our work, we temporally process a sequence of images collected during execution with an attention mechanism as we deal with anomaly cases whose indicators may appear long before their occurrences (i.e., unstable sub-tower structure causing a collapse at later time steps or pouring objects into an almost full container, etc.) rather than instant anomalies.

In our previous work [7], we present a symbolic-level anomaly identification method that processes the outputs of a visual scene modeling system [42], proprioceptive sensors and auditory data to identify anomalies with preprocessed hand-crafted features. In this study, we extend it by presenting a three-stream anomaly identification framework that extracts low-level features from 2D images directly without considering high-level symbolic domain symbols, which does not require any hand-crafted feature engineering effort. Furthermore, we deal with a more extensive set of anomaly cases for a service robot performing everyday tabletop scenarios.

III. CONVOLUTIONAL THREE-STREAM ANOMALY IDENTIFICATION (CLUE-AI) FRAMEWORK

The CLUE-AI framework consists of three steps of processing three different sensory streams. The visual data (2D images) collected from an RGB-D camera (ASUS Xtion RDB-D Camera) mounted on the robot's head is processed in the first stage. The second stage analyzes auditory data obtained by a microphone (PSEye microphone) mounted on the Baxter robot's lower torso. The final stage deals with gripper-related data (i.e., the position of the gripper (the distance between the gripper tips) and the force applied by the gripper to the object at hand). The following subsections elaborate on the procedures that take place in these aforementioned streams.

A. VISUAL STREAM

Sequential RGB frames are retrieved from videos obtained from the head camera in this research, and these frames are sampled at a fixed frequency of 0.125 Hz. Using the timestamps of the frames, this sampling method provides a frame sequence that is sorted in ascending order. To capture temporal dependencies among anomaly indicators, convolutional

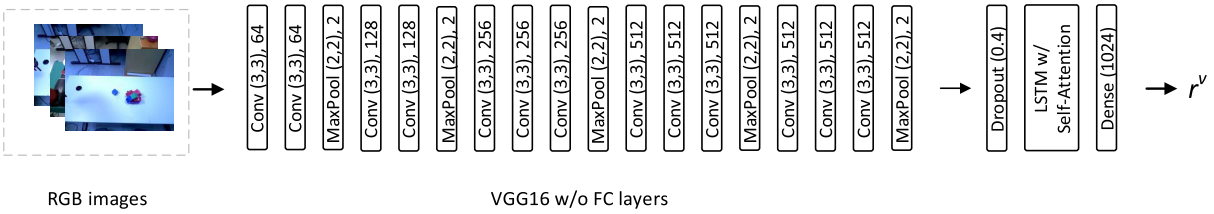


FIGURE 2. The architecture for processing visual stream.

visual features are extracted from each sampled image, and the sequences of these features are fed into the LSTM layers while retaining the order of the sampled frames. These attributes are retrieved from the consecutive 2D images collected by the RGB-D camera mounted on the robot's head using a pre-trained convolutional neural network (CNN) structure. Images are cropped and scaled ($\mathbb{R}^{224\times224}$) before the feature extraction procedure. The altered images are sent into a pre-trained CNN structure after this transformation phase to detect the crucial parts of them. The following CNN architectures are utilized to extract features from consecutive 2D images in order to achieve this goal:

- **Residual Net (ResNet)** [43]: ResNets are neural networks that make training deeper neural networks easier. For deeper networks, they are not only easier to optimize, but they also provide high accuracy. The basic principle underlying this architecture is that instead of learning functions without references, the layers are treated as residual functions that reference the layer inputs.
- **AlexNet** [44]: AlexNet is a CNN architecture with five convolutional layers and three fully-connected layers. A max pooling operation is performed after the first, second, and fifth convolution layers. On the ImageNet dataset [45], the design performs with high accuracy.
- **VGG** [46]: The principle behind this structure is to use architecture with small (3×3) convolution filters to evaluate networks with increasing depths. It typically has 16 to 19 layers. The VGG variation with 16 layers (VGG16) is used in this research. VGG16 has thirteen convolutional layers and three fully connected layers. Each convolution layer is followed by ReLU, and after the second, fourth, seventh, tenth, and thirteenth convolution layers, a max pooling operation is performed.

In this research, pre-trained implementations of these vision models on ImageNet [47] are used. These models are trained on a variety of images, including a variety of objects. The final layers of these pre-trained CNNs, which contain fully-connected layers, are not employed as we use these models for feature extraction. f_t^v is the symbol for each feature vector at time t. Following the extraction of relevant visual features (f^v) related to anomaly indicators, a dropout function with a probability of 0.4 is used. The generated features are then fed into LSTM cells, which are used to learn anomaly patterns. The LSTM result is then sent into an attention layer to assess the importance of the images in the sequence that contribute to the anomaly decision. The LSTM layers in this work adopt scaled dot-product self-attention [48] to come up with attention values associated with the images in the sequence. After calculating self-attention scores, they are fed into a dense layer. The output vector is then created, which includes the concatenation of the attention output with the LSTM outputs ($r^v \in \mathbb{R}^{1024}$) and is fused with the auditory and proprioceptive modality outputs.

The steps of the proposed CLUE-AI framework while processing visual modality are depicted in Figure 2. To handle sequential RGB images, the proposed visual processing technique involves two phases: extraction of features and learning of these extracted features. Inputs are accepted in the form of sequential 2D RGB images.

The feature extraction stage includes convolutional and pooling layers. The convolution layers are labeled with the kernel sizes and channels that correspond to them. Without its classification layers, which contain fully connected layers, these sequential layers form the VGG network. A dropout layer follows this structure, which is followed by LSTM layers that learn the visual anomaly indicators. The LSTM outputs are then passed into an attention layer, which generates attention values based on the image positions in the sequence. Note that after each convolution layer, a rectified linear unit (ReLU) is utilized as an activation function; these are omitted in the figure for brevity.

B. AUDITORY STREAM

The audio data collected by the microphone mounted on the robot's torso is processed in two steps. The initial phase involves extracting features from the collected audio data. Mel-frequency Cepstral Coefficients (MFCC) features are used to do so. For this purpose, the library Librosa [49] is utilized. f_t^a represents the extracted auditory features of an observation collected at time step t. In the second phase, these extracted MFCC features (f^a) are sent into a CNN block. The block is made up of four convolution layers that are stacked one on top of the other. As an activation function, each convolution layer is followed by a ReLU unit. Max-pooling is applied after these procedures, and the resulting vector is fed into a dense layer, which outputs the resulting auditory feature vector ($r^a \in \mathbb{R}^{64}$).

The contents of the proposed framework's auditory stage, which analyses audio data, are shown in Figure 3. The audio

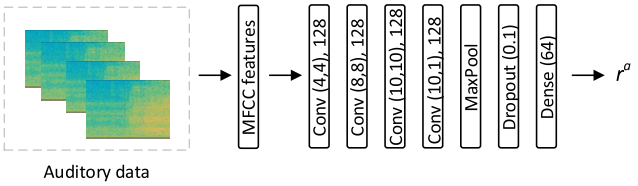


FIGURE 3. The architecture for processing auditory stream.

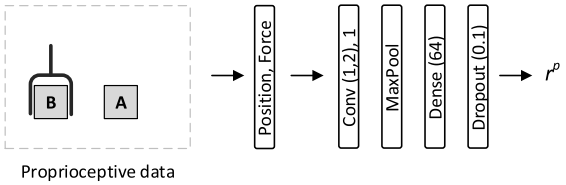


FIGURE 4. The architecture for processing proprioceptive stream.

data is processed in two steps, as shown in the figure. The audio data's MFCC features are extracted first, and then they're processed in convolution layers. The convolution layers' outputs are fed into a dense layer, which creates a vector that includes processed audio features.

C. PROPRIOCEPTIVE STREAM

The gripper on the robot's arm is used to manipulate objects in the environment. With the gripper, continuous proprioceptive observations are gathered. The position of the gripper and the force applied by the gripper to the object in hand are considered features (f^p) in this study. Figure 4 depicts how these features are processed within a tiny CNN. This CNN's output is fed into a max pooling layer, which is then followed by a dense and dropout layer. As a result, the final vector for the gripper data is generated ($r^p \in \mathbb{R}^{64}$).

D. LATE FUSION

The explained feature vectors make up the aforementioned visual, auditory, and proprioceptive streams. These collected visual, auditory, and proprioceptive features are concatenated to a single fused vector ($r^{fused} \in \mathbb{R}^{1152}$) using a late fusion technique. Following that, the fusion's resultant vector is fed into a dense layer, which is then followed by a dropout layer with a probability of 0.4. The resulting vector is then passed into another dense layer and a softmax function, where the result is mapped into seven classes (six anomaly classes and the success class). Based on these findings, the most likely anomaly type is picked for the anomaly's identification. Note that our representation handles the described problem as a sequence-to-one model. The late fusion method is depicted in Figure 5.

IV. EXPERIMENTAL ANALYSIS

The CLUE-AI framework is evaluated in real-world experiments using the Baxter robot, which performs everyday object manipulation scenarios. This section presents the experimental setup and the results.

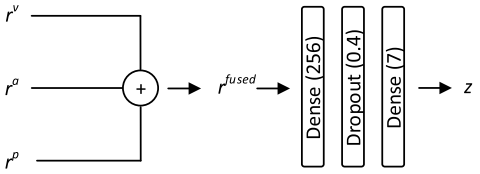


FIGURE 5. The late fusion scheme of the CLUE-AI framework.

FIGURE 6. Baxter robot and the objects that are used in the experiments.

A. EXPERIMENTAL SETUP

In this section, the CLUE-AI framework is evaluated in real-world experiments using the Baxter robot performing everyday object manipulation scenarios. Experiments based on anomaly situations are carried out with an RGB-D camera attached to Baxter's head and a microphone mounted on its body (Figure 6). The robot interacts with objects to perform tasks by executing its actions in everyday object manipulation tasks. We use a variety of objects in terms of color and shape that are placed in random poses in the environment. In this study, we investigate the following types of everyday object manipulation anomalies:

1) LOCATION CHANGE ANOMALY (LOC)

This anomaly type refers to situations in which object states (particularly their locations) are changed without the robot's knowledge, yet the objects remain in the scene. Another robot or a human may perform this change.

2) OBJECT DISAPPEARANCE ANOMALY (DIS)

This anomaly type refers to situations where one or more objects with which the robot interacts are taken out of the environment or from the robot's field of view. Another robot or human in the environment may be responsible for this disappearance.

3) EARLIER UNSTABLE ACTION ANOMALY (EUA)

This anomaly occurs when the robot is assigned the task of constructing a tower out of a collection of objects. The tower, however, collapses during construction as a result of an earlier

unstable action that results in a misplaced or inappropriately placed sub-tower.

4) OVERTURNING ANOMALY (OTA)

To avoid encountering anomaly cases, a robot should select the appropriate push point whenever it needs to push an object. Otherwise, the object may collapse, fall, or shift its orientation. This anomaly type corresponds to such cases.

5) SPILLED OBJECT ANOMALY (SPC)

When the robot is tasked to pour objects or liquids into a container, the destination container may be empty or partially full. However, the robot may not recognize the destination container's fullness, causing the destination container to overflow and spill the contents of the first container onto the table.

6) FULL CONTAINER ANOMALY (FCA)

Unlike the previously described anomaly type (SPC), this anomaly type refers to situations in which an object is placed or stacked into an already full container. As a result, the object falls as a result of the robot's placement of the object in the container.

The robot can perform five actions in this study. To move its arm to an object location, the robot executes *move-to-object*. To locate its arm to a destination point, it executes *move-to-location*. *pick* is used to grab a target object with the gripper, and *place* is used to place the object to its destination. It can also use the *push* action to move a target object along a specified dimension and distance with its gripper.

Experiments on seven distinct classes involving six anomaly cases and the safe cases are conducted to evaluate the presented framework. Safe scenarios (SAFE) are also included in the evaluation to identify and deal with any potential false alarms that may arise during the anomaly detection procedure. Experiments are carried out on a server with the following specs: Intel Core i7-7700K 4.20GHz CPU, 32 GB RAM, and an NVIDIA Quadro P6000 GPU with 24 GB memory. The proposed framework is evaluated using 249 real-world anomaly scenarios collected by the Baxter robot (68 SAFE, 22 LOC, 41 DIS, 33 EUA, 18 OTA, 43 SPC, 24 FCA), and the average duration of the conducted scenarios is 81.58 seconds. When updating the network, an adaptive weighting approach is used to deal with the data set's class imbalance problem, where anomaly classes with the lowest total number of instances are valued more. For each class, the data set is randomly partitioned into the train (80%) and test (20%) sets. The following parameters are set: For each training epoch, the Adam algorithm [50] is used as an optimizer, and cross-entropy loss to calculate the loss. The models are trained for 40 epochs, which are set empirically considering the average number of iterations that is sufficient with a learning rate of 10^{-4} (η). The hidden size is 512, and the LSTM structure is made up of one layer. The results are reported based on the average scores on the test set for ten random seeds.

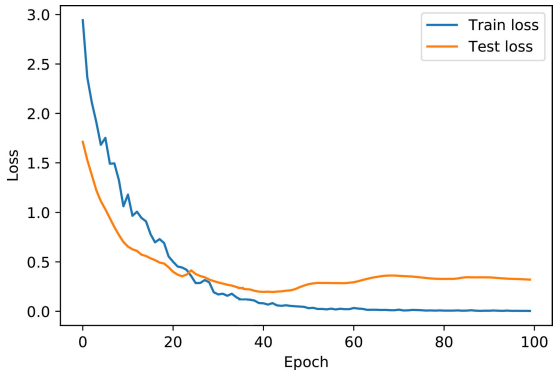


FIGURE 7. Change on the train and test loss for $\eta = 10^{-4}$ for the VGG16-based setting.

TABLE 1. Number of epochs that each model is trained and the number of parameters (in millions, w/o the dense layers) that each vision model includes.

	VGG16	AlexNet	ResNet18	ResNet101
# of epochs	40	55	55	60
# of parameters	14.7	2.4	11.1	42.5

We conduct an analysis to find the number of epochs needed to train CLUE-AI. An analysis of the loss values with regard to the number of epochs is shown in Figure 7. The epoch number is shown on the x-axis, while the loss is shown on the y-axis. As observed in the plot, the training process is better to be ended around the fortieth epoch.

In order to determine the number of epochs required for training for each used pre-trained visual model for feature extraction, an experiment is conducted where ten different random seeds are used. The average number of epochs for each pre-trained vision model in the visual stage of CLUE-AI are shown in Table 1 along with the number of parameters of the corresponding vision model excluding the fully connected layers.

B. EXPERIMENTAL EVALUATION

We present the evaluation of the presented CLUE-AI framework from various aspects. First, we perform a feature extraction analysis to show how different vision models perform in extracting visual features from images for the anomaly identification task. Second, we show the effectiveness of CLUE-AI in identifying object manipulation anomalies compared to the other methods in the literature. Third, we perform an ablation study on how different sensory modalities contribute to identifying anomalies. Fourth, we show the robustness of CLUE-AI to noisy data. Last, we perform an analysis to show how the kernel shape affects the anomaly identification performance when processing the auditory data stream.

1) FEATURE EXTRACTION ANALYSIS

The CLUE-AI framework is tested with four different feature extraction settings, employing the following CNNs to justify our design choices: ResNet18, ResNet101, AlexNet, and

VGG16. It's important to note that ResNet18 and ResNet101 are two ResNet variants with different numbers of layers in the corresponding CNN structure. As we formulate anomaly identification as a multi-class classification problem, we present confusion matrices to provide insights into the performance of CLUE-AI for each anomaly class. Moreover, we present precision, recall and f-score scores as overall performance metrics to provide further insights into the performance as we deal with an imbalanced dataset.

The normalized confusion matrices for different CNNs that extract visual features are shown in Figures 8a-8d. Given the results in Figure 8a, one might conclude that ResNet18 is unable to extract visual features for some classes. The average pooling (AP) layers of ResNets (with a square kernel size of seven) that are placed before the excluded dense layers are the main cause of this situation. As a result, the size of the extracted features shrinks. In this study, only the dense layers of these models are excluded. Since anomaly indicators cannot be distinguished in such a situation, ambiguity in anomaly classes arises. For example, in a ResNet18-based setting, SAFE and LOC are frequently confused classes. LOC (the anomaly cases where an object's location is changed by an external agent) is confused with DIS, as shown in Figure 8a (the anomaly cases where an object is taken out of the scene by a human). That is, the object's change of location is mistaken for its disappearance. The ResNet101-based setting and the ResNet18-based setting produce similar results. For the anomaly identification task, however, AlexNet and VGG16-based settings provide more accurate classification results (Figures 8c-8d).

The minimum classification score for the AlexNet structure as the feature extractor belongs to class SAFE among the classes with a value of 0.85. When VGG16 is used to extract relevant information about anomaly symptoms, the SAFE class has a minimum score of 0.85, indicating that no anomaly has occurred. The interpretation of this situation is that a small number of chickpeas may be spilled on the table, being occluded by the container on some occasions.

The performance analysis of the presented feature extraction methods in terms of precision, recall, and f-score metrics are summarized in Table 2. Both variations of the ResNet settings, with and without average pooling (AP) layers placed last, are taken into account in this analysis. In comparison to the setting with AP layers, the ResNet18 setting without AP layers produces better results. The AP layers (with a square kernel size of seven) of ResNets that are placed before the excluded dense layers are the main cause of this situation. As a result of the usage of these AP layers, the size of the extracted features reduces. Despite the fact that the ResNet-based feature extraction method provides better performances without AP layers, the AlexNet-based feature extraction technique has an f-score of 93.88%. As shown in the table, 5GG16 is capable of extracting relevant visual features, outperforming the other ResNet-based settings and slightly outperforming the AlexNet-based setting with an f-score of 94.34%. As can be seen from the presented results,

TABLE 2. Performance evaluation of different feature extraction methods.

	Precision ($\mu \pm \sigma$)	Recall ($\mu \pm \sigma$)	F-score ($\mu \pm \sigma$)
ResNet18 w/ AP	89.32 ± 2.17	87.60 ± 2.39	87.16 ± 2.75
ResNet18 w/o AP	92.65 ± 2.54	91.52 ± 2.82	91.43 ± 2.85
ResNet101 w/ AP	89.63 ± 4.50	88.04 ± 4.48	87.76 ± 4.49
ResNet101 w/o AP	91.86 ± 3.80	90.43 ± 4.03	90.33 ± 4.15
AlexNet	94.63 ± 2.57	93.91 ± 2.71	93.88 ± 2.76
VGG16	94.90 ± 2.23	94.34 ± 2.21	$\mathbf{94.34 \pm 2.26}$

TABLE 3. Time elapsed (sec) during training and testing with different feature extraction techniques.

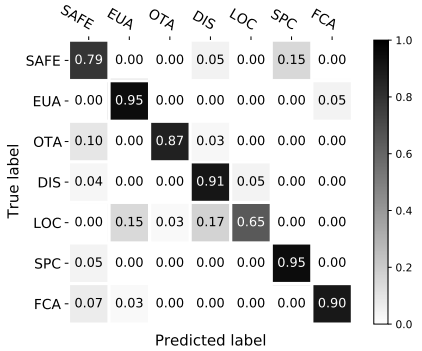
	Train (μ)	Test (μ)
CLUE-AI w/ ResNet18 w/ AP	0.037	0.007
CLUE-AI w/ ResNet18 w/o AP	0.052	0.022
CLUE-AI w/ ResNet101 w/ AP	0.057	0.024
CLUE-AI w/ ResNet101 w/o AP	0.095	0.048
CLUE-AI w/ AlexNet	**0.036**	**0.004**
CLUE-AI w/ VGG16	0.059	0.024

the performance of CLUE-AI also depends on the vision model used to extract visual features. The pre-trained models contribute to extracting relevant parts of images for the anomaly identification task, where VGGs provide the highest scores in our case.

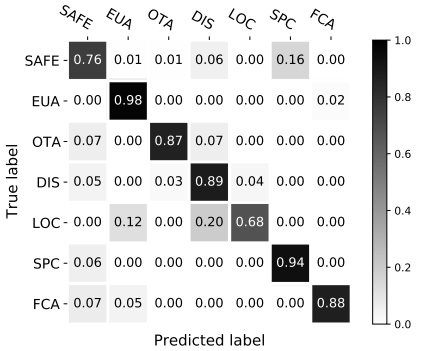
The elapsed time for training the anomaly models with features extracted by different CNNs is shown in Table 3. Each column corresponds to the elapsed times for training and testing in seconds (sec), and each row contains the results of a different feature extraction method. The average execution times are reported for ten executions. As can be seen from the results, AlexNet takes the least amount of time to train and test models. The most time (approximately 0.095 seconds) is required to train for a single instance using a ResNet101 without average pooling layers and feeding the extracted features into LSTMs.

2) OVERALL PERFORMANCE ANALYSIS

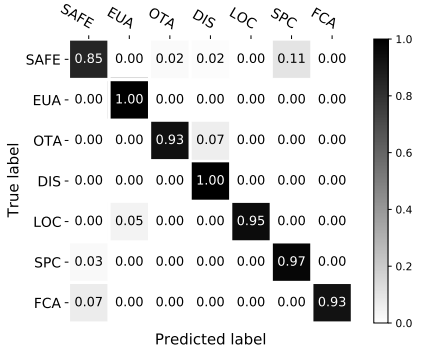
In this experimental analysis, the proposed CLUE-AI framework is compared to various methods for the anomaly identification task. These methods include hidden Markov models (HMMs) and recurrent neural networks (RNNs). We choose HMMs for comparison as they are commonly used in the literature to address the anomaly identification problem [9], [25], [27], [28], [40]. For the RNN-based method, LSTMs are replaced with the corresponding structures in the CLUE-AI framework. To extract relevant features from the input images for the HMM-based method, incremental principal component analysis (IPCA) is used. These features are then fed into HMMs to classify the anomaly type. For each anomaly type, an HMM model with binary latent states (safe and the corresponding anomaly type) is trained. The observation history, including images, is fed into each trained model in the event of an anomaly. As a result, the scores for each anomaly model that corresponds to the likelihood of that anomaly model are generated. The anomaly type is then determined by



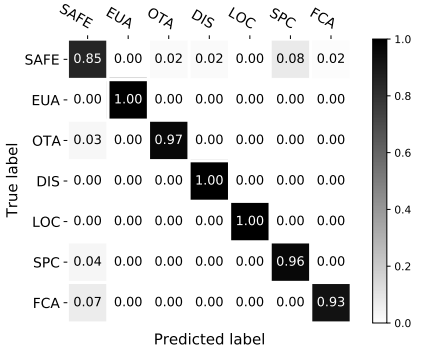
(a) ResNet18 as a feature extractor.



(b) ResNet101 as a feature extractor.



(c) AlexNet as a feature extractor.



(d) VGG16 as a feature extractor.

FIGURE 8. **Normalized confusion matrices of different feature extraction techniques for the visual stream.**

TABLE 4. **Comparative performance analysis of different methods.**

	Precision ($\mu \pm \sigma$)	Recall ($\mu \pm \sigma$)	F-score ($\mu \pm \sigma$)
HMM	74.05 ± 4.31	71.63 ± 5.73	70.75 ± 6.08
Violet-LSTM [7]	96.20 ± 3.77	89.32 ± 6.82	92.14 ±5.99
CLUE-AI w/ RNN	93.51 ± 3.83	92.39 ± 4.48	92.34 ± 4.51
CLUE-AI w/ LSTM	94.90 ± 2.23	94.34 ± 2.21	**94.34 ± 2.26**

combining these likelihoods with the results of the auditory stage. The weighted average scores are reported in Table 4 after each method is run ten times with different seeds.

In terms of all criteria, VGG16 as the feature extractor with LSTMs (CLUE-AI w/ VGG16-LSTM) outperforms the other deep learning-based methods and HMMs with IPCA as the feature extractor (IPCA-HMM). VGG16 performs at an f-score of 92.34% when the extracted features are processed with RNNs. When VGG16 and LSTMs are combined in the CLUE-AI framework, they outperform the other methods with an f-score of 94.34% in classifying anomaly types.

When compared to the results obtained with the anomaly identification algorithm in [7] that uses symbols as visual features, it can be concluded that the CLUE-AI framework with VGG16-LSTM outperforms the symbol-based anomaly identification algorithm with an f-score of 94%. In comparison to the symbol-based anomaly identification algorithm, the CLUE-AI framework is evaluated on a more extensive set of anomaly scenarios (i.e., a higher number of anomaly classes). Moreover, a challenging set of scenarios for a scene interpretation system is included in this extended set of anomaly scenarios (i.e., cluttered environments or containers full of objects).

3) ABLATION STUDY OF CLUE-AI

The CLUE-AI framework proposed in this paper processes data from various sensory modalities using LSTMs with a dot-product self-attention mechanism. To identify the anomaly types, a multimodality analysis is used to examine the contribution of each processed sensory modality and the dot-product self-attention mechanism.

The precision, recall, and f-score metrics for different sensory modality settings of the CLUE-AI framework using LSTMs with a dot-product self-attention mechanism are presented in Table 5. Furthermore, one column in the table is set aside to indicate whether the dot-product self-attention mechanism is used in the given situation. Each column in the table represents the scores for the corresponding metric, and each row corresponds to a different setting. When only the visual modality with dot-product self-attention is considered for classification, this setting yields an f-score of 88.70%. When the gripper-related data is combined with the visual modality, the f-score improves slightly (88.72%).

Combining visual and auditory modalities with dot-product self-attention leads to improved performance with an f-score of 93.51%. Incorporating the auditory modality

TABLE 5. Sensory modality and attention analysis.

Modality			Attention	Precision	Recall	F-Score
s_v	s_a	s_p				
		✓		52.40 ± 9.72	54.78 ± 8.68	50.52 ± 9.24
	✓			82.48 ± 5.22	81.73 ± 4.14	80.52 ± 4.95
✓			✓	89.83 ± 4.62	88.91 ± 4.50	88.70 ± 4.69
✓		✓	✓	90.24 ± 3.91	88.91 ± 3.82	88.72 ± 4.83
✓	✓		✓	94.23 ± 3.09	93.47 ± 3.36	93.51 ± 3.29
✓	✓	✓		93.55 ± 2.44	92.33 ± 2.88	92.33 ± 2.90
✓	✓	✓	✓	94.90 ± 2.23	94.34 ± 2.21	$\mathbf{94.34 \pm 2.26}$

allows the robot to more effectively distinguish anomaly cases in which sound is a discriminating indicator (for example, when an object falls down as it is pushed, the robot perceives the sound as an observation; on the other hand, it does not perceive sound when an external agent changes the object's location).

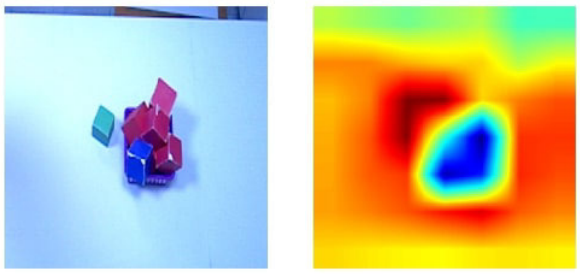
In our CLUE-AI design, we integrate an attention mechanism to maximize the contribution of relevant visual clues or indicators obtained from previous gatherings to the overall decision. As seen in the table's last two rows, while incorporating all sensory modalities provides better scores, the dot-product self-attention-enabled CLUE-AI design provides improved scores as the best results, as a 2% increase in the overall f-score.

The audio modality is used as complementary data in this study. Nonetheless, in the majority of disappearance cases, no auditory perception is gathered. Furthermore, the sounds that occur when an object hits the table, or the container intermingle in SPC (the anomaly cases where objects are spilled in a pouring task due to overflowing) and SAFE scenarios. As a result, it may not be an essential indicator for these anomaly types.

A class activation map representation of an FCA case is shown in Figure 9. Representations based on gradient-weighted class activation mapping (Grad-CAM) [51] are presented in this study. Class activation maps (CAMs) reveal which segments of the input impact the final decision. The contribution of red regions to the final decision is high, while the contribution of blue regions is low. The left image represents the images perceived through the robot's camera, while the right image represents the corresponding CAM. An object falls out of the container on the table in this anomaly type (FCA). The framework appears to be focused more on the region where the object that falls out of the container lands than on the container itself.

4) PERFORMANCE ON NOISY DATA

We have conducted an experiment to analyze the robustness of CLUE-AI to noise by blurring the perceived visual observations. Figure 10 shows the f-score performance of CLUE-AI with respect to varying kernel sizes with bar graphs. We use a Gaussian blur filter to blur the perceived visual observations with varying kernel sizes. We adaptively set the standard deviation (σ) of the Gaussian filter as follows, where



(a) (b)

FIGURE 9. (a) The scene from the RGB-D camera of the robot. (b) CAM of the corresponding FCA case.

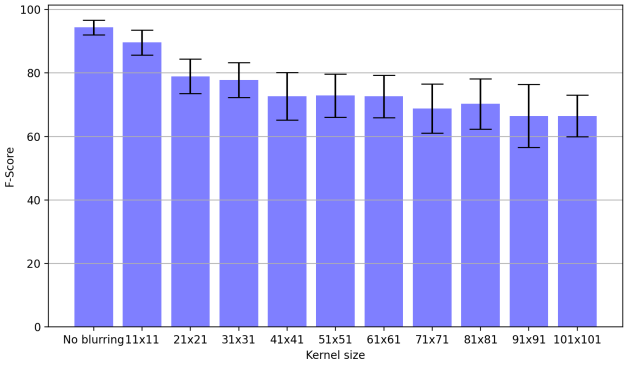


FIGURE 10. Performance analysis of the CLUE-AI framework on blurred observations.

k is the kernel size:

$$\sigma = (k - 1)/4 \tag{1}$$

In the figure, the y-axis corresponds to the f-score value achieved by that setting, and the x-axis corresponds to a distinct kernel size setting. The first bar represents the case where no blurring is applied to the input. As can be seen from the results, blurring the input images with a kernel size of 11 provides approximately an f-score of 90%, which is a slightly degraded performance compared to the setting where no blurring takes place. When the images are blurred more (kernels from 21 up to size 31), approximately an f-score of 78% is achieved. As the kernel size increases from 41 to 101, the lower and comparable performance (approximately an f-score of 66%) becomes, as expected, as the images are highly blurred in these settings.

5) ANALYSIS OF AUDIO PROCESSING

The CLUE-AI framework treats the auditory stream as 2D data ([time, features]) and convolution layers with 2D kernels are used in the framework. The processing of auditory data with various shaped kernels is evaluated in this study. Rectangular kernels (i.e., (16,4) and (16,5) with strides of the same size) are introduced to the CLUE-AI framework's auditory stage to accomplish this.

The scores achieved by different shaped kernels for auditory stream processing are shown in Figure 11. The x-axis

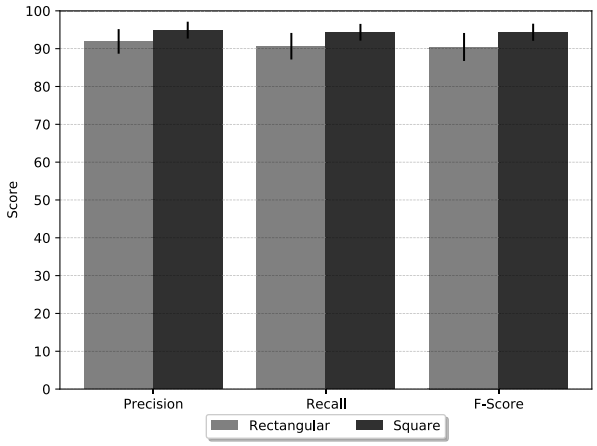


FIGURE 11. Performance analysis of the CLUE-AI framework with different kernels.

shows the metrics for each setting, while the y-axis shows the scores. For each metric, the first bars (light gray bars) represent the setting with rectangular kernels, while the second bars (dark gray bars) represent the setting with square kernels. As can be seen from the graph, the CNN structure with square kernels on the auditory stream improves each metric, particularly the f-score, which improves by 4%.

V. DISCUSSION AND CONCLUSION

In this paper, we present CLUE-AI for cognitive robots to identify anomaly cases that may be encountered during task execution. CLUE-AI takes into account three distinct sensory streams, visual, auditory and proprioceptive modalities, to handle anomaly cases that may arise during everyday object manipulation tasks. In the first stream, convolutional visual features are extracted by VGG16 and fed into self-attention-enabled LSTM cells to capture anomaly indicators. In the second one, MFCC features are extracted from the auditory modality, and a CNN block is employed. Last, gripper-related data is processed with a designated CNN layer. These sensory data are combined with a late fusion methodology. The framework is evaluated on real-world scenarios with a Baxter robot performing everyday object manipulation tasks. The feature extraction analysis on the visual stream shows that VGG16 provides better scores in identifying anomalies. Class activation maps are analyzed to investigate the contribution of input images' regions to the identification results. Different kernel shapes are analyzed for processing the auditory stream, and the performance on noisy data is presented. The comparative analysis indicates that the framework has the ability to identify anomaly cases, scoring an f-score of 94%, outperforming the other methods.

There are some potential future directions for this work. Given that the CLUE-AI framework proposed in this study can only identify the modeled anomaly cases, one would expect that it would classify an unknown anomaly type considering the indicators of the most similar anomaly type. One future direction is to add predictive uncertainty measures into the framework to provide confidence estimates on the resulting anomaly classes. Another potential research direction would be processing the anomaly identification results in a reinforcement learning-based approach to select corresponding recovery actions. In such a work, segmentation techniques such as peripheral vision could be integrated, where models inspired by human eyes' peripheral ability are constructed [52].

ACKNOWLEDGMENT

The authors thank Dr. Sinan Kalkan and Dr. Gokhan Ince for their invaluable comments on this research; and Arda Inceoglu and Cihan Ak for their great contributions to the robot experiments.

The author Dogan Altan was with Istanbul Technical University by the time the research was conducted.

REFERENCES

[1] A. Grinbaum, R. Chatila, L. Devillers, J.-G. Ganascia, C. Tessier, and M. Dauchet, "Ethics in robotics research: CERNA mission and context," *IEEE Robot. Autom. Mag.*, vol. 24, no. 3, pp. 139–145, Sep. 2017.

[2] P. Lin, K. Abney, and G. Bekey, "Robot ethics: Mapping the issues for a mechanized world," *Artif. Intell.*, vol. 175, nos. 5–6, pp. 942–949, Apr. 2011.

[3] R. Etemad-Sajadi, A. Soussan, and T. Schöpfer, "How ethical issues raised by human–robot interaction can impact the intention to use the robot?" *Int. J. Social Robot.*, vol. 14, no. 4, pp. 1103–1115, Jun. 2022.

[4] N. Wan, L. Li, C. Ye, and B. Wang, "Risk assessment in intelligent manufacturing process: A case study of an optical cable automatic arranging robot," *IEEE Access*, vol. 7, pp. 105892–105901, 2019.

[5] A. C. Ak, A. Inceoglu, and S. Sariel, "When to stop for safe manipulation in unstructured environments?" in *Proc. 18th Int. Conf. Auto. Agents MultiAgent Syst.*, 2019, pp. 1767–1769.

[6] A. Inceoglu, E. E. Aksoy, A. C. Ak, and S. Sariel, "FINO-Net: A deep multimodal sensor fusion framework for manipulation failure detection," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Sep. 2021, pp. 6841–6847.

[7] D. Altan and S. Sariel, "What went wrong? Identification of everyday object manipulation anomalies," *Intell. Service Robot.*, vol. 14, no. 2, pp. 215–234, Apr. 2021, doi: 10.1007/s11370-021-00355-w.

[8] S. Karapinar and S. Sariel, "Cognitive robots learning failure contexts through real-world experimentation," *Auto. Robots*, vol. 39, no. 4, pp. 469–485, Dec. 2015, doi: 10.1007/s10514-015-9471-y.

[9] A. Inceoglu, G. Ince, Y. Yaslan, and S. Sariel, "Comparative assessment of sensing modalities on manipulation failure detection," in *Proc. IEEE ICRA Workshop Perception, Inference Learning Joint Semantic, Geometric Phys. Understand.*, Jun. 2018, pp. 1–6.

[10] O. Pettersson, "Execution monitoring in robotics: A survey," *Robot. Auto. Syst.*, vol. 53, no. 2, pp. 73–88, Nov. 2005.

[11] C. Fritz, "Execution monitoring—A survey," Dept. Comput. Sci., Univ. Toronto, Toronto, ON, Canada, Tech. Rep., 2005.

[12] M. Fahim and A. Sillitti, "Anomaly detection, analysis and prediction techniques in IoT environment: A systematic literature review," *IEEE Access*, vol. 7, pp. 81664–81681, 2019.

[13] K. Choi, J. Yi, C. Park, and S. Yoon, "Deep learning for anomaly detection in time-series data: Review, analysis, and guidelines," *IEEE Access*, vol. 9, pp. 120043–120065, 2021.

[14] J. Carlson, R. R. Murphy, and A. Nelson, "Follow-up analysis of mobile robot failures," in *Proc. IEEE Int. Conf. Robot. Autom.*, Apr. 2004, pp. 4987–4994.

[15] S. Karapinar, D. Altan, and S. Sariel-Talay, "A robust planning framework for cognitive robots," in *Proc. 26th AAAI Conf. Artif. Intell.*, 2012, pp. 1–7.

[16] F. Lopez, M. Saez, Y. Shao, E. C. Balta, J. Moyne, Z. M. Mao, K. Barton, and D. Tilbury, "Categorization of anomalies in smart manufacturing systems to support the selection of detection mechanisms," *IEEE Robot. Autom. Lett.*, vol. 2, no. 4, pp. 1885–1892, Oct. 2017.

[17] S. Gspandl, S. Podesser, M. Reip, G. Steinbauer, and M. Wolfram, "A dependable perception-decision-execution cycle for autonomous robots," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2012, pp. 2992–2998.

[18] G. Steinbauer and F. Wotawa, "Robust plan execution using model-based reasoning," *Adv. Robot.*, vol. 23, no. 10, pp. 1315–1326, Jan. 2009.

[19] M. G. Morais, F. R. Meneguzzi, R. H. Bordini, and A. M. Amory, "Distributed fault diagnosis for multiple mobile robots using an agent programming language," in *Proc. Int. Conf. Adv. Robot. (ICAR)*, Jul. 2015, pp. 395–400.

[20] A. Abid, M. T. Khan, and C. W. de Silva, "Fault detection in mobile robots using sensor fusion," in *Proc. 10th Int. Conf. Comput. Sci. Educ. (ICCSE)*, Jul. 2015, pp. 8–13.

[21] G. Schleyer and R. A. Russell, "Disturbance and failure classification in walking robots," in *Proc. Australas. Conf. Robot. Automat.*, 2011, pp. 1–8.

[22] G. Biswas, H. Khorasgani, G. Stanje, A. Dubey, S. Deb, and S. Ghoshal, "An application of data driven anomaly identification to spacecraft telemetry data," in *Proc. Annu. Conf. PHM Soc.*, 2016, pp. 1–10.

[23] V. Verma, G. Gordon, R. Simmons, and S. Thrun, "Real-time fault diagnosis [robot fault diagnosis]," *IEEE Robot. Autom. Mag.*, vol. 11, no. 2, pp. 56–66, Jul. 2004.

[24] G. G. Rigatos, "Particle and Kalman filtering for fault diagnosis in DC motors," in *Proc. IEEE Vehicle Power Propuls. Conf.*, Sep. 2009, pp. 1228–1235.

[25] D. Altan and S. Sariel-Talay, "Probabilistic failure isolation for cognitive robots," in *Proc. 27th Int. Flairs Conf.*, 2014, pp. 1–6.

[26] D. Altan and S. Sariel, "Empirical analysis of probabilistic methods for failure isolation in robots," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS) Workshop Cognit. Robot.*, Feb. 2016, pp. 1–7.

[27] A. Inceoglu, G. Ince, Y. Yaslan, and S. Sariel, "Failure detection using proprioceptive, auditory and visual modalities," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2018, pp. 2491–2496.

[28] D. Park, H. Kim, and C. C. Kemp, "Multimodal anomaly detection for assistive robots," *Auto. Robots*, vol. 43, pp. 611–629, Mar. 2018.

[29] S. Luo, H. Wu, H. Lin, S. Duan, Y. Guan, and J. Rojas, "Fast, robust, and versatile event detection through HMM belief state gradient measures," in *Proc. 27th IEEE Int. Symp. Robot Human Interact. Commun. (RO-MAN)*, Aug. 2018, pp. 1–8.

[30] E. Di Lello, M. Klotzbucher, T. De Laet, and H. Bruyninckx, "Bayesian time-series models for continuous fault detection and recognition in industrial robotic tasks," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Nov. 2013, pp. 5827–5833.

[31] D. Altan and S. S. Talay, "Hierarchical HMM-based failure isolation for cognitive robots," in *Proc. ICAART*, 2014, pp. 299–304.

[32] P. Long, M. Wonsick, and T. Padir, "A risk informed task planning framework for humanoid robots in hazardous environments," in *Proc. IEEE-RAS 18th Int. Conf. Humanoid Robots*, Nov. 2018, pp. 139–144.

[33] S. Luo, H. Wu, S. Duan, Y. Lin, and J. Rojas, "Endowing robots with longer-term autonomy by recovering from external disturbances in manipulation through grounded anomaly classification and recovery policies," *J. Intell. Robotic Syst.*, vol. 101, no. 3, pp. 1–40, Mar. 2021.

[34] J. Du, H. Yan, T.-S. Chang, and J. Shi, "A tensor voting-based surface anomaly classification approach by using 3D point cloud data," *J. Manuf. Sci. Eng.*, vol. 144, no. 5, May 2022.

[35] C. Lu, Z.-Y. Wang, W.-L. Qin, and J. Ma, "Fault diagnosis of rotary machinery components using a stacked denoising autoencoder-based health state identification," *Signal Process.*, vol. 130, pp. 377–388, Jan. 2017.

[36] L. Wen, L. Gao, and X. Li, "A new deep transfer learning based on sparse auto-encoder for fault diagnosis," *IEEE Trans. Syst., Man, Cybern. Syst.*, vol. 49, no. 1, pp. 136–144, Jan. 2019.

[37] J. Bowkett, J. Burdick, L. Matthies, and R. Detry, "Semantic understanding of task outcomes: Visually identifying failure modes autonomously discovered in simulation," in *Proc. Representing Complex World, Perception, Inference, Learn. Joint Semantic, Geometric, Phys. Understand.*, 2018, pp. 1–2.

[38] I. Ullah and Q. H. Mahmoud, "Design and development of RNN anomaly detection model for IoT networks," *IEEE Access*, vol. 10, pp. 62722–62750, 2022.

[39] D. Park, Y. Hoshi, and C. C. Kemp, "A multimodal anomaly detector for robot-assisted feeding using an LSTM-based variational autoencoder," *IEEE Robot. Autom. Lett.*, vol. 3, no. 3, pp. 1544–1551, Jul. 2018.

[40] D. Park, H. Kim, Y. Hoshi, Z. Erickson, A. Kapusta, and C. C. Kemp, "A multimodal execution monitor with anomaly classification for robot-assisted feeding," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Sep. 2017, pp. 5406–5413.

[41] M. Madry, L. Bo, D. Kragic, and D. Fox, "ST-HMP: Unsupervised spatio-temporal feature learning for tactile data," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2014, pp. 2262–2269.

[42] A. Inceoglu, C. Koc, B. O. Kanat, M. Ersen, and S. Sariel, "Continuous visual world modeling for autonomous robot manipulation," *IEEE Trans. Syst., Man, Cybern. Syst.*, vol. 49, no. 1, pp. 192–205, Jan. 2019.

[43] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[44] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, vol. 25, Dec. 2012, pp. 1097–1105.

[45] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 248–255.

[46] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, *arXiv:1409.1556*.

[47] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet large scale visual recognition challenge," *Int. J. Comput. Vis.*, vol. 115, no. 3, pp. 211–252, Dec. 2015.

[48] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 5998–6008.

[49] B. McFee, C. Raffel, D. Liang, D. Ellis, M. McVicar, E. Battenberg, and O. Nieto, "Librosa: Audio and music signal analysis in Python," in *Proc. 14th Python Sci. Conf.*, 2015, pp. 18–25.

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

[51] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-CAM: Visual explanations from deep networks via gradient-based localization," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 618–626.

[52] M. H. Mozaffari and W.-S. Lee, "Semantic segmentation with peripheral vision," in *Proc. Int. Symp. Vis. Comput.* Cham, Switzerland: Springer, 2020, pp. 421–429.

DOGAN ALTAN (Member, IEEE) received the B.S., M.Sc., and Ph.D. degrees in computer engineering from Istanbul Technical University (ITU), Istanbul, Turkey. He was a Research Assistant with the Faculty of Computer and Informatics Engineering, ITU. He is currently a Postdoctoral Fellow with the Simula Research Laboratory, Oslo, Norway. His current research interests include AI safety, anomaly detection, and anomaly identification.

SANEM SARIEL (Member, IEEE) received the B.S., M.Sc., and Ph.D. degrees in computer engineering from Istanbul Technical University (ITU), Istanbul, Turkey, in 1999, 2002, and 2007, respectively. She was a Researcher with the Georgia Institute of Technology, Atlanta, GA, USA, from 2004 to 2006. She is currently an Associate Professor with the Artificial Intelligence and Data Engineering Department, ITU, directing the Artificial Intelligence and Robotics Laboratory (AIR Laboratory). Her current research interests include robot cognition, AI safety, AI in games, and multi-robot systems.