2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 14-18, 2024. Abu Dhabi, UAE

Multimodal Failure Prediction for Vision-based Manipulation Tasks with Camera Faults

Yuliang Ma[1], Jingyi Liu[1], Ilshat Mamaev[2], and Andrey Morozov[1]

***Abstract*— Due to the increasing behavioral and structural complexity of robots, it is challenging to predict the execution outcome after error detection. Anomaly detection methods can help detect errors and prevent potential failures. However, not every fault leads to a failure due to the system's fault tolerance or unintended error masking. In practical applications, a robotic system should have a potential failure evaluation module to estimate the probability of failures when receiving an error alert. Subsequently, a decision-making mechanism should help to take the next action, e.g., terminate, degrade performance, or continue the execution of the task. This paper proposes a multimodal method for failure prediction for vision-based manipulation systems that suffer from potential camera faults. We inject faults into images (e.g., noise and blur) and observe manipulation failure scenarios (e.g., pick failure, place failure, and collision) that can occur during the task. Through extensive fault injection experiments, we created a FAULT-to-FAILURE dataset containing 4000 real-world manipulation samples. The dataset is subsequently used to train the failure predictor. Our approach processes the combination of RGB images, masked images, and planned paths to effectively evaluate whether a certain faulty image could potentially lead to a manipulation failure. Results demonstrate that the proposed method outperforms state-of-the-art models in terms of overall performance, requires fewer sensors, and achieves faster inference speeds. The analytical software prototype and dataset are available at Github: MultimodalFailurePrediction.**

I. Introduction

Recent breakthroughs in robotics and AI have made unstructured manipulation feasible in industrial environments (e.g., BMW factory and Figure 01 Robot). This development highlights the importance of vision-based manipulation[1, 2, 3] and its robustness. Dense hardware and software integration facilitates robots' perception of physical worlds. However, such extensive integration of these internal components also significantly increases the behavioral and structural complexity of robots. In addition, the complex and dynamically changing environment also increases the uncertainty in the physical world. Therefore, different types of faults and errors are likely to occur in the robotic system, and some of them could lead to an execution failure that may damage the system or human operators. One promising way to prevent dangerous events from occurring is Deep Learning-based Anomaly Detection [4, 5, 6] (DLAD). A typical example of DLAD for time-series data is prediction-based methods. This approach first uses currently observed sensor data (e.g., joint position, joint velocity, IMU data) to predict the next value. Then, an anomaly is detected if the residual between the predicted value and the measured real value is larger than a certain threshold. However, fault tolerance mechanisms allow robots to continue performing their tasks normally even when an error occurs. Consequently, the termination whenever an error is detected reduces efficiency and increases running costs. A failure predictor is helpful for the system to make the decision once an error is detected. In this paper, our focus is directed towards the following questions: What are the potential subsequent outcomes (success or failure) for the given task if a robotic system encounters an error?

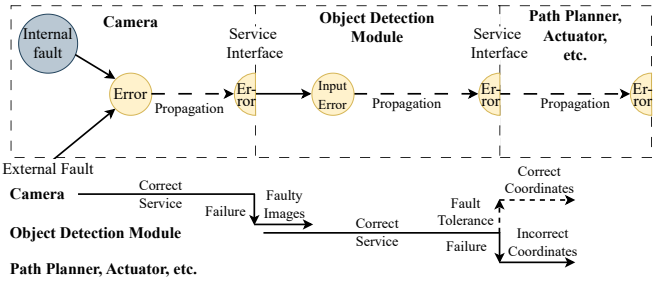

Fig. 1. The fault-error-failure chain adopted from [7]. In a typical vision-based manipulation task, various components generate and deliver heterogeneous services (e.g., images, coordinates, and trajectories) to accomplish the task. However, once an error occurs in one component, the manipulation outcome is uncertain.

Fig. 1 illustrates the error propagation process in a system. Following the definitions of faults, errors, and failures in [7], we conclude the basic features of these three concepts and illustrate some examples as follows:

- **Fault**: An active fault is the origin of an error, e.g., electrical interference, acting as an external fault, could potentially damage the camera sensor.
- **Error**: The deviation between correct service and incorrect service is called an error, e.g., a damaged camera sends a noisy image to the subsequent object detection module.
- **Failure**: A failure happens when the delivered service deviates from the correct service, e.g., the object detection module receives noisy images and gives incorrect location information of objects. Then, a grasp failure is likely to happen.

A fault could occur within different sensors (location), mani-

This work is funded by the Ministry of Science, Research and Arts of the Federal State of Baden-Württemberg for the financial support of the projects within the InnovationCampus Future Mobility (ICM).

[1]The authors are with the Institute of Industrial Automation and Software Engineering, University of Stuttgart, 70550, Stuttgart, Germany. {yuliang.ma, jingyi.liu, andrey.morozov}@ias.uni-stuttgart.de

[2]The author is with Karlsruhe University of Applied Sciences and with Proximity Robotics & Automation GmbH, Germany. mamaev@proximityrobotics.com

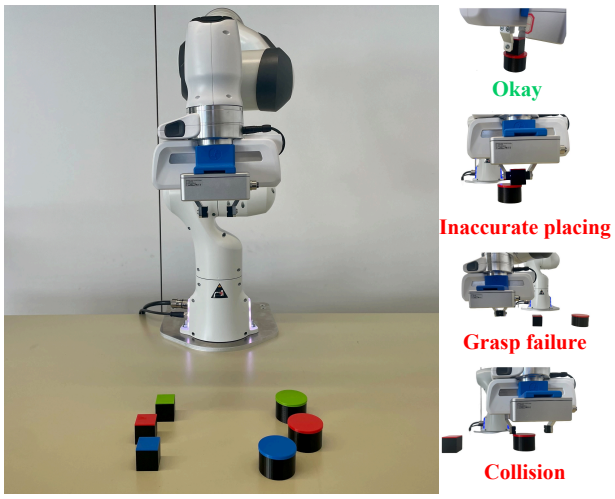

Fig. 2. Vision-based manipulation case study: The Panda manipulator engages in visual manipulation tasks involving objects, as depicted on the left. The objective is for the manipulator to stack small cubes entirely onto cylinders of matching colors. Four potential outcomes are illustrated on the right. "Okay" signifies successful task execution, while the remaining three depict potential failure scenarios.

festing as different patterns (type) and severities (magnitude). For a system, fault attributes may significantly impact the likelihood of failure occurrence. Prediction-based anomaly detection methods can identify errors by assessing deviations from sensor data. However, they often lack the capability to provide an explanation regarding whether the detected error could potentially impact the task or system negatively. In a large-scale factory, the anomaly detector may generate numerous alerts without clear insight into the actual impact of these errors. Inspired by this, our focus in this paper is to predict future manipulation failures when the robotic system has an erroneous image. We propose a multimodal method to detect 1) whether there is an image fault, and 2) to predict whether the detected erroneous image will lead to an execution failure. As such, the failure prediction task is naturally considered as a classification problem.

In this work, we choose the Franka Emika Panda manipulator as our robotic platform for conducting fault injection experiments and data collection. The manipulator is equipped with a wrist-mounted FRAMOS D435e depth camera and a MoveIt path planner [8, 9]. We employ the ROS as the middleware. The manipulation task is defined as defined as a 'Color Match Game' scenario in which the manipulator grasps the cube and places it on the cylinder with the corresponding color. Due to the fact that images are not guaranteed to be of good quality in many applications [10, 11], we inject faults into images and observe potential manipulation failures. Fig. 2 shows the real-world setup and actual outcomes that we observed: 1) *Successful*: the robot successfully positions the cube on the cylinder, with the cube lying entirely within the circular area of the cylinder; 2) *Inaccurate placing*: the cube is not completely positioned within the circular area; 3) *Grasp failure*: the robot completely misses the cube; 4) *Collision*: any unexpected interaction between the robot and objects. The proposed method predicts the probability of the execution failure. Our main contributions can be summarized as follows:

Primary contribution: A novel multimodal method that effectively fuses potentially faulty images with other information (e.g., object detection results, planned trajectory) and predict future execution failure. Compared to the state-of-the-art failure detection and prediction methods [12, 13, 14], our method requires fewer sensors and achieves better overall performance (F1-score). Additionally, the inference speed of our method is approximately 10 times faster than that of other state-of-the-art methods.

Secondary contribution 1: An ROS-based image fault injection method which could inject two types of faults, noise and blur, with parameterizable magnitudes. This method could be easily deployed in both simulation (Gazebo) and real ROS applications.

Secondary contribution 2: A FAULT-to-FAILURE dataset with multimodal information from real-world scenarios, including faulty images, normal images, planned paths, and object detection results (masked images). To the best of our knowledge, there are no publicly available datasets for the error propagation in robotic manipulation scenarios. The dataset and software prototype is available in our git-hub: MultimodalFailurePrediction.

II. Related Work

Inceoglu *et al* introduces a series of works for manipulator failure detection using multimodal sensor fusion in [12]. This work presents Failure Is Not an Option (FINO)-Net which could fuse RGB images, depth images, and audio readings. This end-to-end framework gives a binary failure detection result (i.e., success or failure) during different manipulation tasks. As an extension, they continue to investigate the performance of FINO-Net which classifies different failure modes (e.g., collision, miss the target, overturn) during execution in [15]. Additionally, for robot assistive feeding tasks, an LSTM-based variational autoencoder is adopted to fuse multimodal input including images, microphone readings, and joint states [16, 17]. Anomalous behaviors of the manipulator could be detected via the fusion of high-dimensional data. In other work in [18] and [19], multimodal inputs are used to detect manipulation and grasp failures. However, the aforementioned methods perform failure detection in a reactive manner, when the failure has already happen or is happening. Reactive failure detectors are not effective in preventing failures and protecting the system.

On the other hand, proactive failure detection could predict future failures by leveraging current sensory observation and planned actions. In the domains of robot navigation tasks and autonomous driving, researchers have proposed many interesting works. Ji *et al*. proposes a ProActive Anomaly Detection (PAAD) network to proactively detect anomalous behaviors for a field robot platform in [13]. The multimodal network fuses information of planned path, RGB image, and 2D point cloud to predict probabilities of failure (e.g., collision or no collision) in the next several time steps. This

predictive model has shown dependable failure detection performance of a real robot navigation task in a complex and cluttered field environment. Similar work to PAAD is proposed in LaND [20] and BADGR [21]. Conditioned by a sequence of future control actions, the neural network takes an RGB image as the input and predicts the probabilities of collision within a prediction horizon. While proactive failure detection methods for mobile robots has been extensively studied, it typically assumes that images are of good quality. However, this assumption is not always true in many real-world applications [10, 11]. As such, we consider a similar research question for vision-based manipulation tasks, but from the error propagation perspective. Our starting point differs from existing literature. We focus on proactively predicting the consequences of execution when the manipulator encounters a degraded image. This could help to ensure safety by identifying errors that may fail the task. Otherwise, detected errors are ignored to guarantee working efficiency.

III. Method

Our method helps to resolve the conflict between safety and efficiency for visual-based manipulation tasks. In many cases, the anomaly detector could ensure safety but it lacks the ability to assess whether the detected error will lead to an execution failure. In typical scenarios, the workflow for vision-based manipulation tasks comprises the following steps: 1) `Image Acquisition`; 2) `Feature Extraction`; 3) `Object Detection`; 4) `Path Planning`; and 5) `Task Execution`. The proposed method fuses the faulty image, the object detection result, and the planned path information to predict the probability of manipulation failure. The collected multimodal data is used for training the failure predictor. We will describe the data collection and the multimodal method in the following sections.

A. Data Collection

We inject faults into a robotic system by manipulating the original normal image. Two typical image faults are considered: *Blur* and *Noise*. Camera faults resulting from human factors, such as incorrect camera calibration, are not considered in this case study. These external faults are typically easy to detect and correct. Instead, our focus is on internal signal faults, particularly those that are commonly encountered in practical applications [10]. Representative faulty samples are shown in Fig. 3. The description of fault parameters is as follows:

- *Blur* could occur due to vibration in the surrounding environment of the manipulator. Here we inject blurred images by manipulating a configurable parameter *Kernel size*.
- *Noise* could result from electronic interference or environmental factors. Here we inject Gaussian Noise by randomly changing the *Variance*.

Fig. 4 illustrates the data collection process. During this process, faulty image x_i and masked image x_m are acquired as 480 × 640 RGB images. The masked image is generated

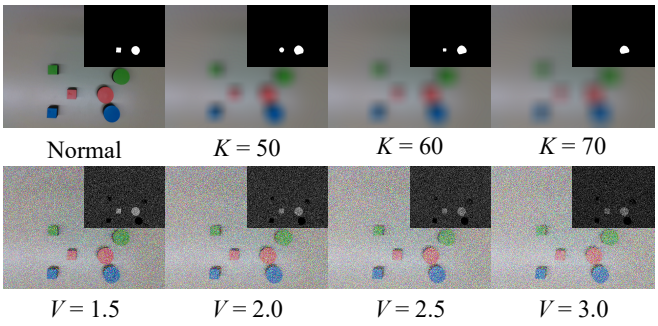

Fig. 3. Faulty images and object detection results. Blurred images with varying kernel sizes are displayed at the top, while noisy images with varying variance are shown at the bottom. The corresponding masked image of each sample is presented in the top-right corner.

using HSV boundaries and the OpenCV library. The planned trajectory x_p is a series of waypoints consisting of expected rotation angles of each joint. The trajectory is generated using the Rapidly-exploring Random Trees (RRT) algorithm. All these data can be obtained before the manipulator starts moving. The ground truth y is a binary output based on the manual observation. In this work, the failure prediction task is considered as a binary classification problem using multiple-modalities input.

B. Multimodal Method

Fig. 5 presents the overview of the multimodal method. This method consists of two modules: an Anomaly Detector (AD) and a Failure Predictor (FP). The AD module takes as input the RGB image, which may be normal or abnormal. Utilizing the AD output, the FP module assesses the probability of failure and determines whether to proceed or

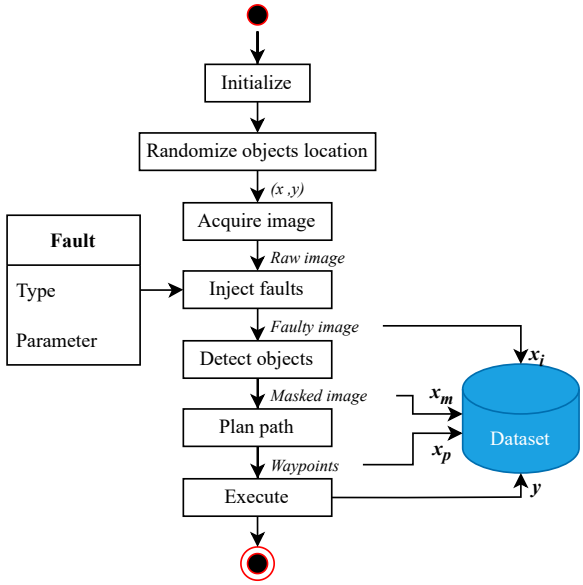

Fig. 4. Data collection process. For each round of pick and place tasks, only one type of image fault is injected. During this process, faulty images, masked images, and waypoints are recorded into Dataset. The dataset is later used for training the anomaly detector and the failure detector.

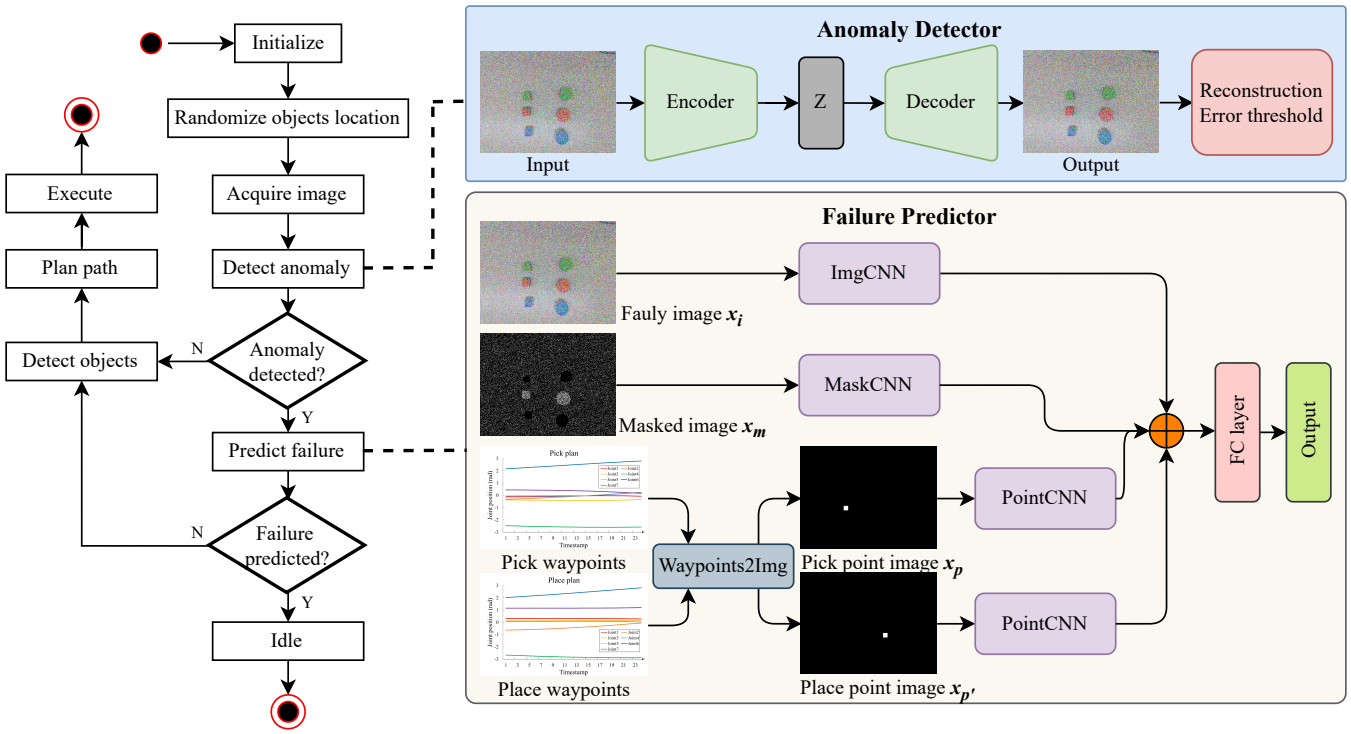

Fig. 5. The architecture of our case study system extended with the (i) anomaly detection and (ii) failure prediction modules introduced in this paper. An anomaly detector checks every acquired image and classifies it as normal or abnormal. A failure detector works once there is a detected faulty image. Faulty images, masked images, and planned waypoints are fed to the detector and it predicts success or failure.

terminate the task. Specifically, the FP module fuses faulty images (from the RGB camera sensor), masked images (from the object detection module), and planned waypoints (from the path planner) in parallel at the feature level. Leveraging these observations, FP evaluates whether a detected faulty image could result in a manipulation failure.

1) Autoencoder-based Anomaly Detector (AEAD): We employ an autoencoder network for unsupervised image anomaly detection. First, 4000 real-world normal images are sampled (which are available in provided dataset[1]), with randomized cubes and cylinders locations within a predefined area. Then, all normal images are utilized for the training process. During training, the Mean Squared Error (MSE) loss of pixel-wise values is employed to optimize the reconstruction performance. Finally, the well-trained AEAD is deployed and it reports an anomaly if the reconstruction error is higher than a certain threshold θ. The encoder and decoder share a symmetrical structure, each consisting of three convolutional layers with the filter size of 3×3. The dimension of the latent space, denoted as z, is set to 32.

2) Multimodal Failure Predictor (MFP): MFP comprises two branches: the image branch and the waypoints branch. In the image branch, faulty images and masked images are processed by ImgCNN and MaskCNN, respectively. Following convolution operations, both sets of generated features are flattened. Then, a concatenation operation is used to achieve feature-level fusion.

Another branch is for handling the planned trajectory derived from the onboard path planner. The fusion of planned path is helpful for improving the detection performance [13]. The Inverse Kinematics (IK) solver is often utilized in many vision-based manipulation tasks. However, IK solver may generate multiple solutions for the expected end-effector pose. In other words, although the target object coordinates remain constant, the planned rotation angle of each joint can vary significantly. Therefore, using 'noisy' action data directly as input for training is inefficient. Inspired by the idea of bird's eye view (BEV), we introduce a *Waypoints2Img* module to transform action data into a projected action point image from the camera's perspective. Specifically, *Waypoints2Img* module first transforms planned action data as end-effector's coordinates (x, y, z) in the world frame[2]. Then, BEV action point locations from the camera's perspective are obtained via coordinates transformation between camera frame and end effector's frame. The relative position between camera and end effector's is previously calibrated by hand-eye calibration operation. Finally, both pick and place action points are represented as a 10×10 pixels square area in a 640×480 (raw image size) pixels images with a black background. Since the performance of *Waypoints2Img* module is entirely dependent on the accuracy of calibration results, we verified it on pick and place tasks without camera faults. The result shows the manipulation precision is reliable.

After *Waypoints2Img* operation, BEV action point images

[1]https://www.kaggle.com/datasets/yuliangma/proactive-failure-prediction

[2]https://frankaemika.github.io/docs/control_parameters.html

are processed by PointCNN. All CNNs shown in Fig. 5 have three convolutional blocks. Each block has one convolutional layer, one batch normalization layer, and one maxpool layer. The filter number is $\{8, 16, 32\}$ and the filter size is 3×3. Extracted features are later flattened, concatenated, and processed by three fully connected (FC) layers. During training, an Adam optimizer is utilized over 50 epochs.

IV. Experiments

We collected 4000 real-world samples, comprising 2000 blurred images with varying kernel sizes (K from 50 to 70) and 2000 noisy images with varying variance (V from 1.5 to 3.0). Table 1 illustrates fault parameters and failure distributions within these samples. The results of two types of faults indicate that the failure rate increases in correlation with the severity of image degradation. Note that no failure occurs in the [50, 55) group of *Blur* fault. This indicates that 1) minor image degradation may not lead to an execution failure, and 2) the calibration result and the manipulation performance are reliable even the camera has faults. In order to evaluate the performance of our method across different fault types, we created three datasets: *Blur*, *Noise*, and *Combination*. Each dataset is split into training (80%) and test (20%) sets.

TABLE I
Fault Parameters and Failure Distributions

Fault types	Parameters	#Success / #Failure	Failure rate
Blur, $K\in$	[50, 55)	491 / 0	0.00%
	[55, 60)	469 / 3	0.64%
	[60, 65)	478 / 24	4.78%
	[65, 70]	286 / 249	46.54%
		Sum: 1724 / 276	13.80%
Noise, $V\in$	[1.5, 2.0)	564 / 13	2.25%
	[2.0, 2.5)	541 / 140	20.56%
	[2.5, 3.0]	214 / 528	71.16%
		Sum: 1319 / 681	34.05%

A. Baselines and Evaluation

To the best of our knowledge, while multimodal-based failure prediction methods are well applied in robot navigation tasks, the domain of robotic manipulation primarily relies on reactive failure detection approaches. In this case study, we benchmark the performance of our method against the following baseline methods:

- *FINO-Net* [12]: A reactive failure detection network for multiple manipulations tasks. FINO-Net takes RGB images, depth images, and audio readings as inputs and detects failures during the execution. To enable proactive failure detection, we replace depth images with masked images. In addition, the audio branch processes the planned waypoints data instead. Note that FINO-Net utilizes ConvLSTM module to capture temporal correlations among input frames/audios for consecutive failure detection. However, we map the discrete input to discrete execution results in our scenario, assuming faulty images and robot's execution outcomes as independent events within one execution round. As such, we omit the recurrent module of FINO-Net in experiments.
- *PAAD* [13]: A multimodal method that combines trajectory images, RGB images, and LiDAR observations. Through feature-level fusion, the PAAD network predicts future failure probabilities for mobile field robots. In our case, we replace the trajectory image with the pick and place point images. Additionally, LiDAR observations are replaced with masked images and a CNN-based supervised variational autoencoder model (SVAE) is employed instead. The prediction horizon is changed to one, focusing on either success or failure in a single execution round.
- *Cui et al.* [14]: A multimodal convolutional neural network predicts autonomous vehicle behavior using RGB images and robot actions (velocities, accelerations, etc.) as input. For a fair comparison, we replace the original vehicle actions data with the manipulator's waypoints data as the input.

We use the evaluation metrics from the aforementioned methods for a fair comparison, including Precision, Recall, and F1-score.

B. Experimental Results

Fig. 6 and Table 2 illustrate the failure prediction performances of different models. In a comparative analysis across different datasets, all models demonstrate significantly higher performance on the *Blur* dataset compared with the *Noise* dataset. This suggests that with the given fault configuration in Table 1, manipulation failures caused by blurred images are more likely to be predicted. In addition, on the *Combination* dataset, the other three baselines experience a decrease in F1-score compared to the *Noise* dataset, while our method exhibits a slight increase. This suggests that our method has a superior adaptability to more complex datasets.

When it comes to the performance comparison among all models, our method outperforms other state-of-the-art methods with a superior overall performance (F1-score) on all datasets. In particular, PAAD and our method demonstrate the highest Precision on the *Blur* dataset. This suggests that these two models excel at identifying faults that may not lead to manipulation failures, avoiding unnecessary fault alarms. However, PAAD shows a poor performance on detecting failures. Conversely, FINO-Net achieves the highest Recall, indicating its better capacity in detecting failures, while it struggles identifying unnecessary fault alarms. We argue that the aforementioned two models struggle to balance false-positive and false-negative rates. This may compromise either working efficiency or safety. In general, our method leads on the F1-score, which means that our approach has a better overall performance on the *Blur* dataset. In the *Noise* dataset, Cui *et al.* outperforms other models with the highest Recall, but it fails to recognize more true positive scenarios. PAAD achieves a comparable F1-score with our model, but it has a relatively higher false-positive rate. In the follow-up experiments conducted on the *Combination*

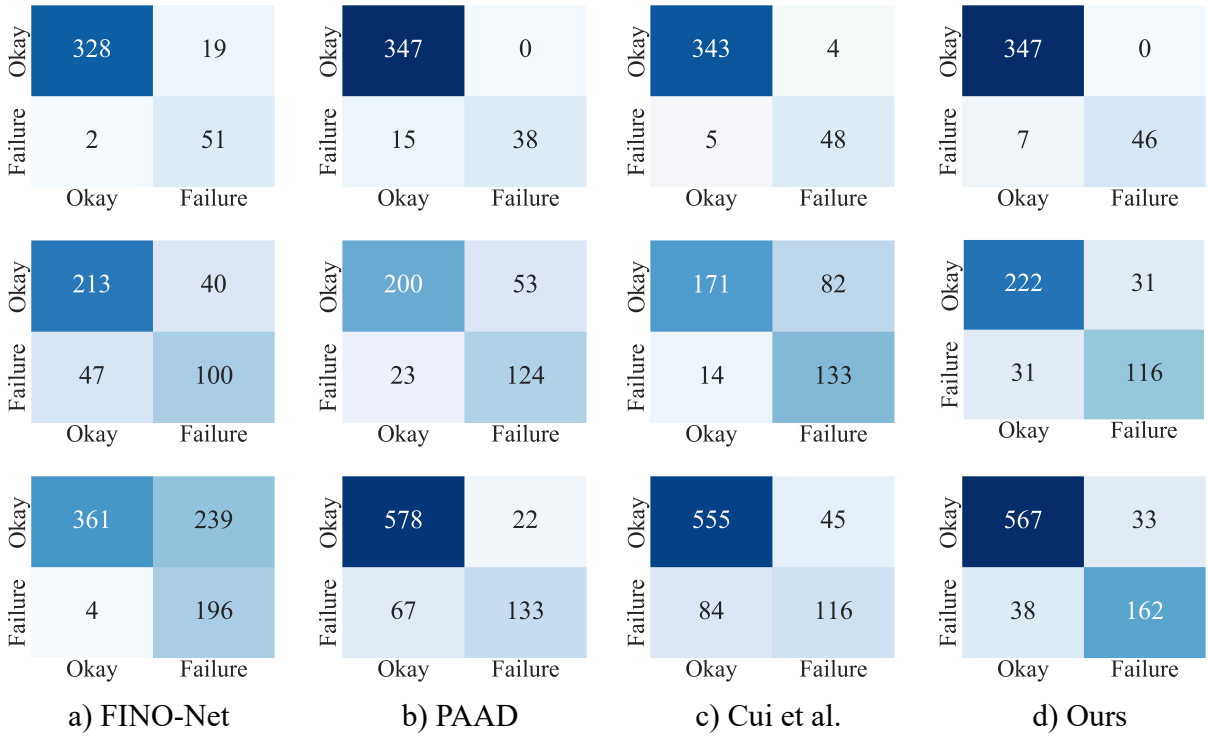

Fig. 6. Confusion matrix for Blur, Noise, and Combination dataset.

TABLE II

FAILURE PREDICTION PERFORMANCE WITH DIFFERENT METHODS

Models	Blur			Noise			Combination (Blur + Noise)			Inf. Time (msec)
	Precision	Recall	F1-score	Precision	Recall	F1-score	Precision	Recall	F1-score	
FINO-Net [12]	94.52	**99.40**	96.90	84.19	81.92	83.04	60.17	**98.90**	74.82	334.37
PAAD [13]	**100.00**	95.86	97.88	79.05	89.69	84.03	**96.33**	89.61	92.85	175.60
Cui *et al.* [14]	98.85	98.56	98.70	67.59	**92.43**	78.08	92.50	86.85	89.59	105.54
Our method	**100.00**	98.02	**99.00**	**87.75**	87.75	**87.75**	94.50	93.72	**94.11**	**13.38**

TABLE III

ABLATION STUDY RESULTS

Models	Precision	Recall	F1-score
RGB image only	93.17	90.02	91.56
RGB + Masked image	91.67	93.86	92.75
RGB + Point image	93.17	**94.27**	93.71
w/o Waypoints2Img	93.17	92.24	92.70
Selected combination	**94.50**	93.72	**94.11**

dataset, our model significantly outperforms the others in terms of F1-score, indicating its superior ability to adapt to more complex datasets. Notably, our method achieves superior results without requiring additional sensors (audio or LiDAR), simplifying the setup while delivering good performance. Additionally, our work considers the error propagation problem from component errors to execution failures, which is not addressed in aforementioned methods.

Additionally, we re-implemented the baseline models and run them on an Nvidia GeForce RTX 3050 Laptop GPU to investigate the offline inference speed. A shorter inference time is beneficial for online failure prediction. In our experiment, the inference time is measured for each test batch (batch size: 64), and this process is repeated 100 times to calculate the average inference time. The results demonstrate that our method is about 10 times faster than other three baseline models. We attribute this improvement to the use of a lighter neural network architecture with fewer hidden units in the fully connected layer.

C. Ablation Study

An ablation study is conducted to assess the contribution of different components in our model. We choose the *Combination* dataset as the reference and the ablated versions of the proposed model are 1) *RGB image only*: only the faulty image branch is used to extract the observation features; 2) *RGB + Masked image*: only ImgCNN and MaskCNN pipeline are used to extract the observation features; 3) *RGB + Point image*: only ImgCNN and PointCNN pipeline are used to extract the observation features; 4) *w/o Waypoints2Img*: the PointCNN module is replaced with a FC layer to process the planned waypoints data directly. The results summarized

in Table 3 indicate that each model design choice positively influences achieving a higher F1-score. Note that our model outperforms the model without the *Waypoints2Img* module in all metrics. This indicates that using action point images as the input instead of 'noisy' waypoints data contributes to better overall performance.

V. Conclusion

In this study, we introduced a multimodal method utilizing deep neural networks for failure prediction for vision-based manipulation tasks considering camera faults. By fusing RGB images, object detection results, and planned paths, our approach effectively predicts the future manipulation failure when an erroneous image is received. Compared to other state-of-the-art approaches, our method demonstrates better overall performance while requiring fewer sensors and achieving a faster inference speed. However, we acknowledge the limitation in this case study. As the next step, we are interested in testing our method using practical objects instead of cubes and cylinders. Simultaneously, we aim to explore the generalization performance of the method in other more complex tasks, e.g, pushing, drilling, and welding.

References

[1] Zeng, Andy, et al. "Tossingbot: Learning to throw arbitrary objects with residual physics." IEEE Transactions on Robotics 36.4 (2020): 1307-1319.

[2] Song, Shuran, et al. "Grasping in the wild: Learning 6dof closed-loop grasping from low-cost demonstrations." IEEE Robotics and Automation Letters 5.3 (2020): 4978-4985.

[3] Yen-Chen, Lin, et al. "Learning to see before learning to act: Visual pre-training for manipulation." 2020 IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2020.

[4] Luo, Yuan, et al. "Deep learning-based anomaly detection in cyber-physical systems: Progress and opportunities." ACM Computing Surveys (CSUR) 54.5 (2021): 1-36.

[5] Ding, Kai, et al. "On-line error detection and mitigation for time-series data of cyber-physical systems using deep learning based methods." 2019 15th European Dependable Computing Conference (EDCC). IEEE, 2019.

[6] Ding, Sheng, et al. "KrakenBox: Deep Learning-Based Error Detector for Industrial Cyber-Physical Systems." ASME International Mechanical Engineering Congress and Exposition. Vol. 85697. American Society of Mechanical Engineers, 2021.

[7] Avizienis, Algirdas, et al. "Basic concepts and taxonomy of dependable and secure computing." IEEE transactions on dependable and secure computing 1.1 (2004): 11-33.

[8] Mamaev, Ilshat, et al. "Grasp detection for robot to human handovers using capacitive sensors." 2021 IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2021.

[9] Tang, Yucheng, et al. "Towards Flexible Manufacturing: Motion Generation Concept for Coupled Multi-Robot Systems." 2023 IEEE 19th International Conference on Automation Science and Engineering (CASE). IEEE, 2023.

[10] Pei, Yanting, et al. "Effects of image degradation and degradation removal to CNN-based image classification." IEEE transactions on pattern analysis and machine intelligence 43.4 (2019): 1239-1253.

[11] Schlosser, Patrick, and Christoph Ledermann. "Robust Human Pose Estimation under Gaussian Noise." 2023 IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2023.

[12] Inceoglu, Arda, et al. "Fino-net: A deep multimodal sensor fusion method for manipulation failure detection." 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, 2021.

[13] Ji, Tianchen, et al. "Proactive anomaly detection for robot navigation with multi-sensor fusion." IEEE Robotics and Automation Letters 7.2 (2022): 4975-4982.

[14] Cui, Henggang, et al. "Multimodal trajectory predictions for autonomous driving using deep convolutional networks." 2019 International Conference on Robotics and Automation (ICRA). IEEE, 2019.

[15] Inceoglu, Arda, Eren Erdal Aksoy, and Sanem Sariel. "Multimodal Detection and Classification of Robot Manipulation Failures." IEEE Robotics and Automation Letters (2023).

[16] Park, Daehyung, Yuuna Hoshi, and Charles C. Kemp. "A multimodal anomaly detector for robot-assisted feeding using an lstm-based variational autoencoder." IEEE Robotics and Automation Letters 3.3 (2018): 1544-1551.

[17] Park, Daehyung, Hokeun Kim, and Charles C. Kemp. "Multimodal anomaly detection for assistive robots." Autonomous Robots 43 (2019): 611-629.

[18] Thoduka, Santosh, Juergen Gall, and Paul G. Plöger. "Using visual anomaly detection for task execution monitoring." 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, 2021.

[19] Gohil, Priteshkumar, Santosh Thoduka, and Paul G. Plöger. "Sensor Fusion and Multimodal Learning for Robotic Grasp Verification Using Neural Networks." 2022 26th International Conference on Pattern Recognition (ICPR). IEEE, 2022.

[20] Kahn, Gregory, Pieter Abbeel, and Sergey Levine. "Land: Learning to navigate from disengagements." IEEE Robotics and Automation Letters 6.2 (2021): 1872-1879.

[21] Kahn, Gregory, Pieter Abbeel, and Sergey Levine. "Badgr: An autonomous self-supervised learning-based navigation system." IEEE Robotics and Automation Letters 6.2 (2021): 1312-1319.