Deep Learning and Computer Vision-Based Autonomous Robotics System for Industrial Automation and Inspection

Veeramani V[1], Dr. M. Antonyraj Martin[2]
[1]Assistant Professor, Sciences and Mathematics Unit, Department of Supportive Requirements, University of Technology and Applied Sciences, Salalah, Sultanate of Oman.
E-mail: veeramani@utas.edu.om
[2]Lecturer, Sciences and Mathematics Unit, Department of Supporting Requirements, University of Technology and Applied Sciences, Salalah, Sultanate of Oman.
E-mail: antonyraj@utas.edu.om.

Article Info

Article History:

Received Apr 11, 2026
Revised May 09, 2026
Accepted Jun 12, 2026

Keywords:

Deep Learning,
Computer Vision,
Autonomous Robotics,
Robot Navigation,
Convolutional Neural Networks,
Real-Time Monitoring

ABSTRACT

Constant monitoring is the essence of industrial work so that machines and infrastructures function efficiently and with absolute safety. Manual inspection is usually a painstaking, labor-intensive, and error-prone procedure, especially during inspection of hazardous environments and large enterprises. Identifying problems such as signs of structural failure, corrosion, and malfunctioning beforehand is important for effective operation and minimizing downtime. The systems are capable of processing visual data instantly and then determining the right course of action after considering patterns learned from experience. A system of autonomous robotics that employs deep learning along with machine vision has been presented for industrial inspection. The use of convolutional neural networks allows for the identification of various types of defects such as cracks, rust, etc. The robotic platform operates using the technology of SLAM (Simultaneous Localization and Mapping). The use of reinforcement learning in optimizing navigation paths ensures effective inspection region coverage. The utilization of edge computing makes real-time inference possible at the level of the robot, which reduces latency and provides better responsiveness. The experimental results show a great increase in the precision of defect detection, accompanied by faster cycles of inspection compared with traditional manual and semi-automated practices. The system improves safety in the workplace, cuts maintenance costs, and boosts operational efficiency, making it widely applied in manufacturing, energy, and aerospace industries.

Corresponding Author:

Veeramani V,
Assistant Professor, Sciences and Mathematics Unit, Department of Supportive Requirements
University of Technology and Applied Sciences, Salalah, Sultanate of Oman.
E-mail: veeramani@utas.edu.om

1. INTRODUCTION

The automated process can be made very effective with the help of robotic arms that have a high intelligence. The use of industrial robots [1], surgical robots [2] and domestic robots [3] is a good example of robotic arms being required. Most robotic arms work in a synchronized way and are programmed to achieve a certain task or are controlled by humans. Robotic arms with the presence of different sensors and cameras have raised many interests in the field of research [4]. These robots possess significant computing powers, have large storage capacities, and apply the corresponding techniques that are based on AI. Such features give such robots an ability to imitate the behavior of human beings.

The use of smart robotic arms is becoming widespread in various sectors. As an example, in smart cities [5], these types of robots can be employed to scan buildings and allow building to create automated three-dimensional (3D) reconstruction. The meaning of 3D reconstruction in computer vision and computer graphics is described as a process of describing physical bodies' shape and appearance, constructing three-dimensional profiles, and measuring the coordinates of any point of the profile. Moreover, it is widely used in many applications range from medicine to free viewpoint video reconstruction, robotic mapping, urban planning, and gaming and virtual environments, among others [6].

Beside reconstructing in 3D, the intelligent robotic arms can prove useful in other aspects. For instance, in smart factories, they can be applied for manufacturing and production, packing and assembling in many industries. Furthermore, in smart hospitals, robot-assisted surgeries enable doctors to perform a lot of complicated operations with a higher accuracy than traditional methods allow. In smart homes, robot arms can help people with disabilities, elderly individuals and parents with children to improve their lives significantly. The speed, accuracy, and efficiency of robot arms let them complete everyday tasks without stress.

Even though advanced robotic arms can be employed in several sectors, it is very complicated to develop applicable structures for them. One of the main problems is making sure path planning for this kind of robotic arm is accurate, safe and efficient [8]. Path planning concerns getting the end-effector to the target while avoiding collisions with obstacles. There are many algorithms involved in this process that define not only the movement of the robotic arm, but also how it should execute its function. A lot of path planning methods using computer vision were analyzed. Since end-effectors are important for robotic arms, the use of end-effectors is a complicated task for robots [9].

Due to digital image processing, computer vision enables machines to accurately interpret their environment, recognize barriers, and estimate their position. This interdisciplinary subject requires the use of computer science, artificial intelligence, and image analysis to obtain valuable data from surroundings and allows computers to take decisions accordingly. Vision algorithms applied in such fields as robotics and handheld devices have produced success in practice and made a significant contribution into science.

A system of autonomous robotics that employs deep learning along with machine vision has been presented for industrial inspection. The use of convolutional neural networks allows for the identification of various types of defects such as cracks, rust, etc. The robotic platform operates using the technology of SLAM (Simultaneous Localization and Mapping). The use of reinforcement learning in optimizing navigation paths ensures effective inspection region coverage. The utilization of edge computing makes real-time inference possible at the level of the robot, which reduces latency and provides better responsiveness. The system improves safety in the workplace, cuts maintenance costs, and boosts operational efficiency, making it widely applied in manufacturing, energy, and aerospace industries.

2. LITERATURE REVIEW

[11] Construct a pioneering framework with the help of computer vision technology and deep learning to eliminate obstacles in IoT-enabled robotic systems in real-time. Our focus is on the streams/images captured by sensors and geographic data as opposed to using the algorithms of robotic path planning based on geolocation. The software makes use of real-time data from moving robots throughout different scenarios regarding time and space for research purposes. In the case of video/image data, it is transmitted using sensor nodes in robotic applications. In this research work, the foremost thing we do is to identify the future movement of an autonomous moving robot. The framework of our approach relies on efficient computer vision techniques combined with a deep learning classifier. The computer vision techniques are applied to enhance frame quality, segment objects and extract features from using the robot's video. The Long-Term Short Memory (LSTM) classifier will automatically identify the movements of robots based on the first features provided. The suggested model is chiefly created by an LSTM classifier, which is suitable for making prior marketing decisions based on the first sequence features of the video frames and for solving the issues of exploding and vanishing gradients. Thus, LSTM enhances the forecasting time while ensuring excellent success rates, which let the system in robotic applications intervene in situations where collision occurs due to obstacles during the inside and outside of the operations. Simulation results with the support of widely known research datasets prove the model efficiency compared to the existing ones. The overall success rate of the suggested model has been raised by about 5% and the computations complexity has been lowered by the approximate duration of 84%.

[12] Proposed a model mainly comprises an LSTM classifier, which is capable of predicting marketing decisions based on the first sequence details of the video frames and eliminating problems of exploding and vanishing gradients. Hence, the LSTM increases the forecasting time and provides a good success rate. It allows the systems to operate in robot applications under circumstance when collisions may take place due to the presence of any obstacle in the course of the robot inside or outside the operations. The simulation results, which are based on well-known research dataset show that the model has provided very good results compared to the currently used model. The success ratio of the suggested model has been increased by around 5% and the complexities of computations have been decreased by around 84%.

[13] Makes use of an imitation learning method to prepare a supervised set of data for robot control employing expert agent demonstrations obtained using Carla simulator under various traffic conditions. This supervised set of data includes three forms that complement each other, and we made it publicly available with our other materials. PilotNet neural model is applied in two modifications: 1st with the output of complete individual commands for braking and accelerating by adding the dropout; 2nd which improves the 1st modification and adds vehicle speed. Both models use the above mentioned set of data for training. The results of the experiments prove that models are able to work successfully in traffic despite their simple and shallow design, knowing no difficulty in driving in clear road conditions.

[14] Utilizes imitation learning methodology to create a supervised data set for driving the robot by leveraging demonstrations of the expert agent collected by Carla simulator in diverse traffic conditions. This supervised data set consists of three attractive and complementary varieties of data, which we share publicly in addition to our other materials. PilotNet model is utilized in two main versions and stages, namely, the 1st version to output pure commands of braking and acceleration combined with dropping out some of the data; the 2nd version improves the 1st version and adds some additional parameters such as the speed of the vehicle. The two variants of PilotNet models utilize the above-mentioned data set for training. Experiment results show that the

models performed quite well in traffic with no problems, despite being simple and shallow and not being able to drive in difficult conditions.

3. METHODOLOGY

This methodology puts together computer vision, deep learning, SLAM-based localization, reinforcement learning, and edge computing into one autonomous robot inspection framework. Each part is crafted to enable accurate defect localization, communication, and instant decision-making in an industrial milieu. This process leads to the formulation of a reliable, intelligent, and fast inspection system as shown in Figure 1.

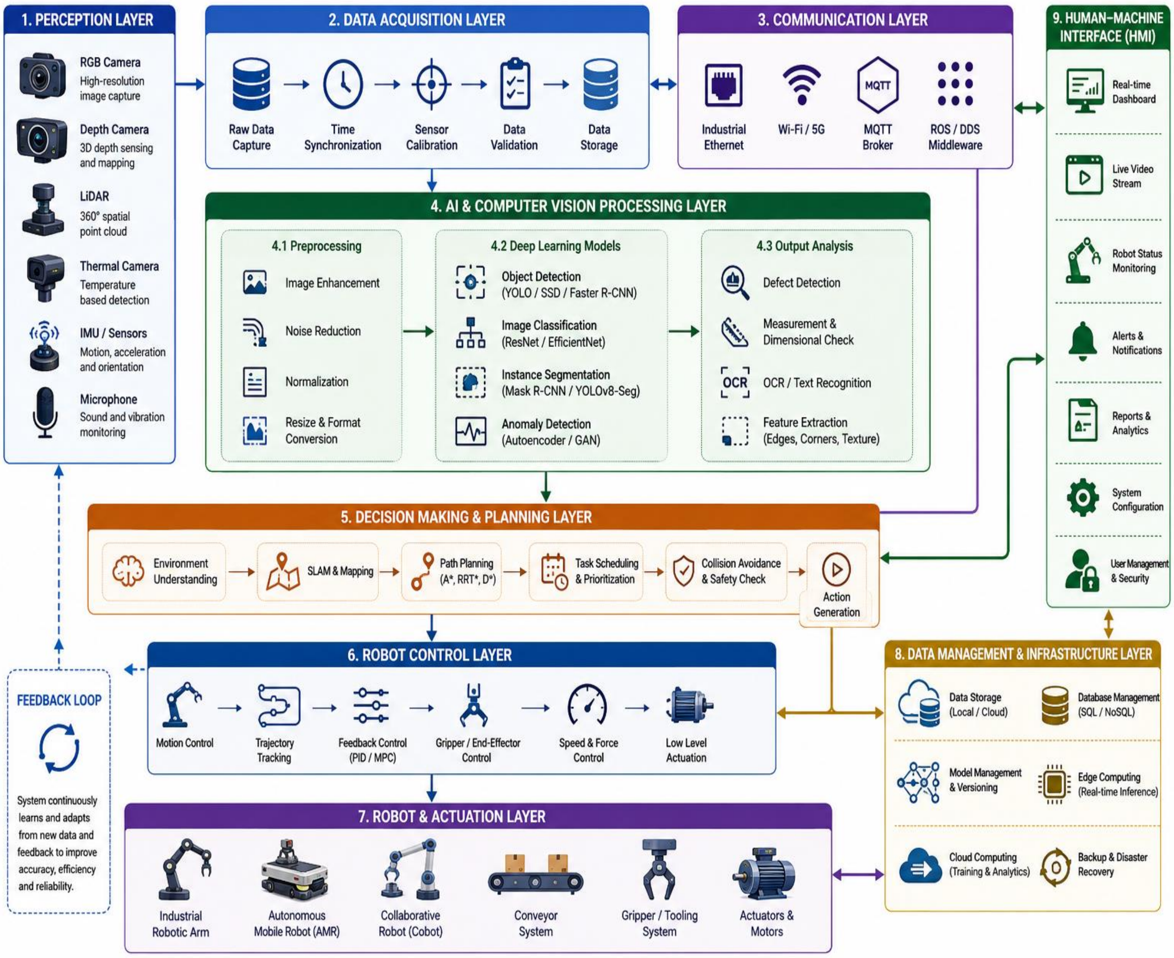

Figure 1. Working process of the proposed model

3.1. Data Collection and Preprocessing

During the data collection stage, images are obtained through the camera sensors installed on the robotic platform, while the platform goes through the inspection zones. The collected data set includes various surface imperfections like cracks, rust, and irregularities in the surface area. All images taken are in place and processed for the training and testing of the deep learning model. Normalization of images is performed as preprocessing in order to maintain uniformity in the values of the pixels of an image. This method enhances uniformity of images and minimizes the effect of different light conditions. It improves the efficiency and stability of CNN during the detection and identification procedure of defects.

$$I_{norm}(x,y) = \frac{I_{x,y}-I_{min}}{I_{max}-I_{min}} \tag{1}$$

Where $I_{norm}(x,y)$= the normalized pixel intensity value; $I_{x,y}$ = the original intensity of the pixel at the given location $(x,y)(x,y)(x,y)$. I_{min}= the minimum intensity of the pixel in the entire image; I_{max}= the maximum intensity of the pixel in the whole image. The normalization procedure applies scaling of the pixel values in the image by entering the pixel intensities into the range from [0,1], enabling improvement in the model performance for the defect identification based on the CNN technology [15].

3.2. CNN-Based Defect Detection and Classification

CNNs have been developed to analyze data in a hierarchical way. CNNs start by recognizing basic elements such as edges and curves, and move on to the identification of complex components such as human faces and object shapes. This hierarchical approach works similar to how the human visual system works and allows CNNs to process visual data effectively. B. CNN LAYERS A CNN typically consists of a number of layers with specific functions. Detailed information pertaining to individual layers is provided below:

The Convolutional layer is considered the basis of CNN architecture because it contains several convolutional kernels, or filters, that are crucial for the process of feature extraction. The convolutional kernel moves across the image, splitting it into smaller sections of overlapping matrices referred to as receptive fields. This segmentation of images makes it possible to extract various features from images. The operation of convolution takes place by moving the kernel over the image and carrying out element-wise multiplication between the kernel's weights and the outputs of the receptive field, thus creating a feature map, which represents the activation of a filter at a particular address in the image. The mathematical representation of the convolution operation can be provided as follows in Equation

$$f_k^l(p,q) = \sum_c \sum_{x,y} i_c\,(x,y) \cdot e_k^l(u,v) \tag{2}$$

Where $f_k^l(p,q)$ shows the activation of neuron (p,q) at the l^{th}feature map generated by the k^{th}filter; $i_c(x,y)$ indicates the pixel value at position (x,y) in the c^{th} channel of the input image, and $e_k^l(u,v)$ represents the value at the position (u,v) in the k^{th}filter for the l^{th}feature map.

The pooling layer has an important role in reducing the dimensions of feature maps, thereby making it easier to carry out convolution in subsequent layers. The action carried out by this layer, also called downsampling, means that information is lost simultaneously as a result of size reduction. This layer acts separately on each feature map and achieves downsampling through such well-known methods as average pooling and max pooling. Nonetheless, this loss proves beneficial because it reduces the computational burden and improves robustness regarding minor shifts to the input image. The pooling step is mathematically defined in Equation(3), where Z_k^l is the pooled feature map generated at the lth layer from the kth input feature map. The precise form of pooling used is defined through the function g_p.

$$Z_k^l = g_p(F_k^l) \tag{3}$$

The role of the activation layer is very important in CNNs as it is responsible for the incorporation of non-linearity in the network design. The introduction of non-linearity ensures that the output of the activation layer is not merely a linear transformation of the net inputs, but an actual non-linear transformation that helps the network better recognize complex and sophisticated patterns in input data and the output being produced.

The Fully Connected (FC) layer being referred to is a traditional neural network layer that connects the previous layer's neurons with the current layer. The various connections create a net of information exchange that allows the FC layer to understand the complicated relationships between features and classes. This layer is positioned at the end of the neural network and is responsible for producing the final output. Unlike pooling and convolution layers, the FC layer works globally, allowing it to receive inputs from the feature extraction stages and analyze the outputs of all previous layers globally. Because of its global connectivity, the FC layer is able to perform non-linear combination of selected features and obtain significant high-level abstractions and features necessary for performing accurate classification tasks.

The function of batch normalization is aimed at normalizing the output from the previous layer by means of taking the mean and dividing by the batch standard deviation. The process of normalization is completed by computing the mean μ_B and variance σ_B^2 of the feature maps in the mini-batch and using them for transforming the feature map, as illustrated in Equation 4.

$$N_l^k = \frac{F_l^k - \mu_B}{\sqrt{\sigma_B^2 + \varepsilon}} \quad (4)$$

Here, F_l^k is the input feature-map, N_l^k is the normalized feature-map, μ_B and σ_B^2 show the mean and variance of a feature-map for a mini batch, respectively, and ε is a tiny constant added to the variance to avoid division by zero. This approach addresses the problems caused by internal covariate shifts in the feature maps. Internal covariates shift refers to changes in the distribution of the activations of layers while training, which may contribute to the slow convergence of the network and complicate the parameters setting. The BN layer resolves this problem by normalizing the previous layers output and allowing the activations to obtain the zero mean and unit variance values. In addition, it results in smoothening the gradient fow and serves as a regularizer.

Dropout, which is a regularization technique used in neural networks, resolves the problem of overfitting. Overfitting is an occurrence where a neural network learns highly intricate, non-linear relationships between the inputs and outputs that it is trained upon. These relationships cause the neurons to co-adapt, so the model memorizes particular patterns from the training data, rather than acquiring information about attributes that could be generalized. Dropout solves this problem by introducing stochasticity into the training process and enhancing the model's ability to learn more robust features. Dropout is based on the principle of randomly removing a group of neurons at every training step. The random dropout yields an ensemble of different thin architectures with lesser number of connections. Upon completion of training, the weights of the remaining neurons are taken as the average between the weights of all thin architectures.

3.3. SLAM-Based Localization and Mapping

SLAM can be helpful for local areas as well. For instance, parking areas are one such area. The driving speeds are low in a parking area, making vision a better solution than in high-velocity cases. The parking area might be known (like one's own garage) or unknown (like a public parking garage) but can still benefit from using SLAM. In addition, since SLAM doesn't rely on GNSS, it is especially useful in indoor or underground parking areas, which use only visual sensors with odometry (speed and angle of turning), or IMU measurements. In the case of unknown public parking zones, the position of both the vehicle and any obstacles (including pillars, sidewalls, etc.) can simultaneously be recognized and then used by the parking system. For the home zone parking process, a pre-made map and a frequently repeated trajectory of the parking maneuvers can be stored in the software of an automated car. Each time the car comes back to the car home zone, the re-localization will be done by mapping the found features with the stored map.

The technique that employs multi-level surface (MLS) maps to locate the vehicle and plan out the routes in parking lots has been introduced. The method makes use of graph-based SLAM technology for mapping and subsequently the MLS map determines the global route from the starting point to the end of the route and provides accurate vehicle location. The work done in this project states that grid map technique along with EKF SLAM was used with W-band radar-based autonomous back-in parking system. This paper presents an effective EKF SLAM algorithm designed to achieve real-time processing. Authors present a system that makes use of AVM/Lidar sensor fusion for identification of parking lane and real-time loop closing. They proved that both filtering and optimization SLAM systems can provide an accurate vehicle parking assistance system without using GNSS. The task involves the application of pose graph optimization for the purpose of generating an optimally planned trajectory as well as a world map of the parking lot. The techniques used for the extraction of semantic features include the guide signs, parking lines, and speed bumps, which are more stable and reliable in comparison with traditional geometrical features, especially underground. In addition to that, an EKF was used for the completion of the localization system for the purpose of autonomous driving.

3.4. Reinforcement Learning-Based Path Planning

In this section, we illustrate the framework setup for DRL, including S and A, as well as the reward function. In our study, we are using 7-DOF robot arm manipulator, so if we take the joint configuration and speed as the input, the learning process is quite slow and might not give us the required results. To solve this issue, we will calculate the end-effector position and velocity of the robot calculate IK afterwards for controlling the actuator's position.

3.5. Edge Computing for Real-Time Inference

Edge computing is significant in allowing the rapid processing necessary for functioning autonomous vehicles and connected transport networks, where delays in making decisions affect both safety and productivity. In the robotic inspection system, edge computing is employed for making decisions on the robot itself in order for it to act immediately. Instead of sending the many inspection files to a cloud service, the data gathered is processed by built-in computing systems.

The model of a convolutional neural network is utilized in the edge device to detect and classify defects in real time. The visual images received from the robot cam are processed locally to detect surface defects including cracks and corrosion. This type of processing provides rapid feedback for navigation and inspection purposes. Inference at the edge improves the operation of the system due to faster response times and the ability to function without a network connection.

4. EXPERIMENTAL OUTCOMES

The efficiency of the autonomous robotic inspection system was examined in terms of fault detection, efficiency of navigation, and real-time inference. Various experiments were carried out in which images with flaws like cracks, corrosion, and other surface defects were presented. During this process the work of the convolutional neural network was analyzed through accuracy and loss metrics. The efficiency of the robotic navigation was evaluated through autonomous movement research, while the work of edge computing was evaluated according to real-time inference time.

The assessment involved four important metrics: classification accuracy, training convergence, defect detection ability, and inference time. These metrics indicate the effectiveness of the proposed method in performing robust and efficient industrial inspections outside the lab environment.

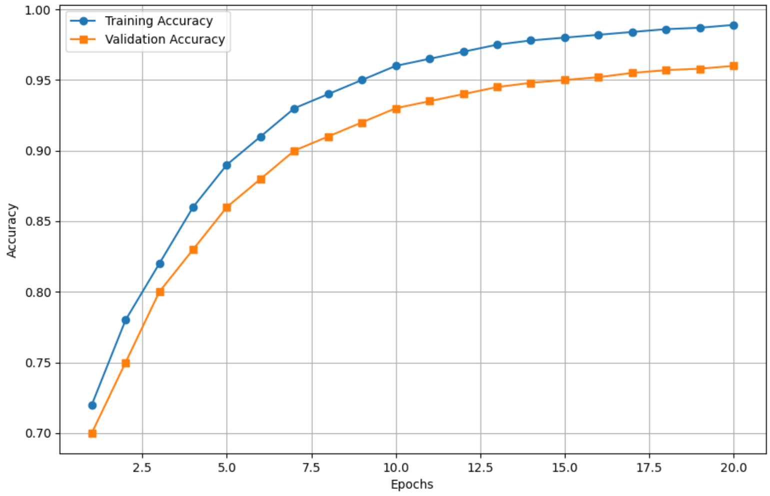

Figure 2. Training and Validation Accuracy Curve of the Proposed Model

In Figure 2, the accuracy curve shows the learning capability of the proposed method throughout the training process. The accuracy value improves as the number of epochs grows, which implies that the CNN learns useful features from the industrial defects images. The trend of the validation accuracy is similar to that of the training accuracy, confirming that the CNN can generalize the information it learned from the training, to the data it has never seen before. The small gap between the values of the two accuracies shows that there is little overfitting and thus provides evidence of the reliability of the proposed defect classification model.

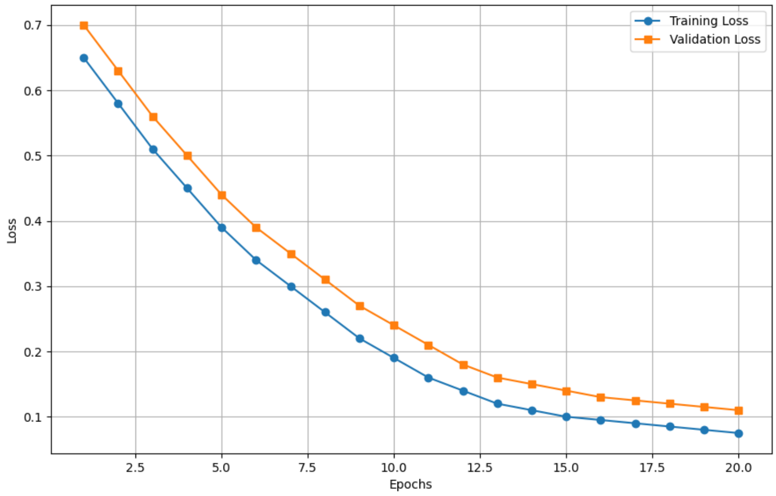

Figure 3. Training and Validation Loss Curve of the Proposed model

The loss curve shown in figure 3 illustrates the decline in prediction error in CNN training. One can see that training loss is decreasing as the model sets parameters and extracts defect features. Additionally, the decrease in validation loss confirms that the model shows stable performance on untested data. The convergence of both curves shows successful training of the model and optimization of CNN in industrial defect detection.

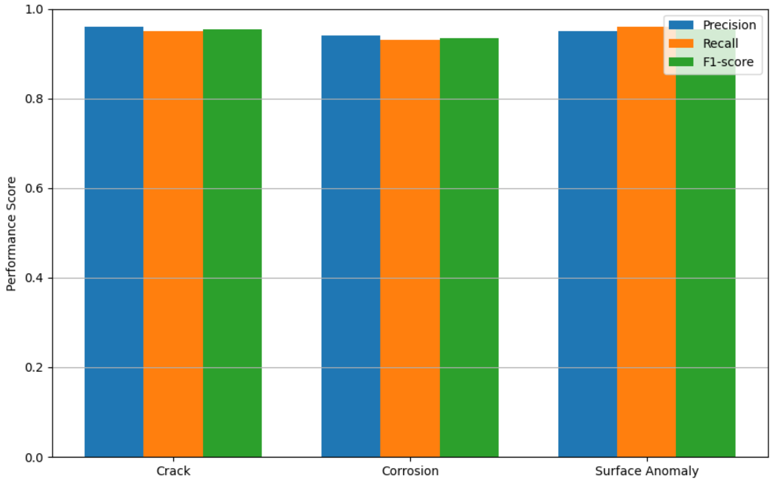

Figure 4. Defect Detection Performance

Figure 4 presents a performance comparison chart that assesses the classification capabilities of a CNN model with respect to various types of defects, such as cracks, corrosion, and surface defects. In this context, precision is representative of the accuracy of classes of defects that have already been detected, recall denotes the identification of all defects in the dataset, while the F1-score signifies a balanced assessment of precision and recall. The results obtained for different types of defects show that the proposed vision-based assessment system is able to detect various surface defects.

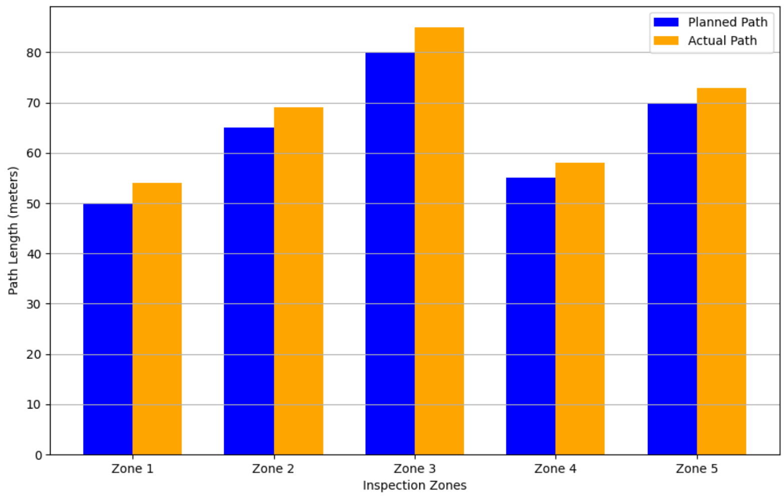

Figure 5. SLAM Navigation Path Efficiency Analysis

The navigation efficiency chart presented in Figure 5 displays a comparison between the optimized route generated by the autonomous navigation algorithm and the actual route traveled by the robot during industrial inspection. The planned route is the shortest possible course based on the application of SLAM mapping as well as the use of reinforcement learning methods for route optimization. The actual path shows how far a robot has actually moved in the real world with the influence of many environmental constraints. The fact that there is not much difference between the planned and the actual routes shows that the robot was localized well, successfully avoided obstacles, and did a great job in covering the inspection area. This is strong evidence that the proposed robotic system is capable of autonomous nature navigation in complex industrial environments.

5. CONCLUSION

In this study, we introduce a system of autonomous robotics that employs deep learning along with machine vision has been presented for industrial inspection. The use of convolutional neural networks allows for the identification of various types of defects such as cracks, rust, etc. The robotic platform operates using the technology of SLAM (Simultaneous Localization and Mapping). The use of reinforcement learning in optimizing navigation paths ensures effective inspection region coverage. The utilization of edge computing makes real-time inference possible at the level of the robot, which reduces latency and provides better responsiveness. The experimental results show a great increase in the precision of defect detection, accompanied by faster cycles of inspection compared with traditional manual and semi-automated practices. The system improves safety in the workplace, cuts maintenance costs, and boosts operational efficiency, making it widely applied in manufacturing, energy, and aerospace industries.

REFERENCES

[1] Kumbhar, M.F., Ghosh, S. and Attaullah, H.M., 2025. Machine Learning Algorithms for Computer Vision in Robotics. *Artificial General-Internet of Things (AG-IoT) for Robotics: Advanced Computer Vision Applications and Future Trends*, pp.165-192.

[2] Alatabani, L.E., Ali, E.S., Mokhtar, R.A., Khalifa, O.O. and Saeed, R.A., 2022, August. Robotics architectures based machine learning and deep learning approaches. In *8th International Conference on Mechatronics Engineering (ICOM 2022)* (Vol. 2022, pp. 107-113). IET.

[3] Raj, R. and Kos, A., 2025. An extensive study of convolutional neural networks: Applications in computer vision for improved robotics perceptions. *Sensors*, *25*(4), p.1033.

[4] Shahria, M.T., Sunny, M.S.H., Zarif, M.I.I., Ghommam, J., Ahamed, S.I. and Rahman, M.H., 2022. A comprehensive review of vision-based robotic applications: Current state, components, approaches, barriers, and potential solutions. *Robotics*, *11*(6), p.139.

[5] Bethini Kavya and Dr. Kakarla Hari Kishore (2025) "Internet of Robotic Things (IoRT): An Intelligent Framework for Automated Manufacturing", *IIRJET*, 10(3). doi: 10.32595/iirjet.org/v10i3.2025.220.

[6] Khan, A., Li, J.P., Hasan, M.K., Varish, N., Mansor, Z., Islam, S., Saeed, R.A., Alshammari, M. and Alhumyani, H., 2022. PackerRobo: Model-based robot vision self supervised learning in CART. *Alexandria Engineering Journal*, *61*(12), pp.12549-12566.

[7] Singh, A., Kalaichelvi, V. and Karthikeyan, R., 2022. A survey on vision guided robotic systems with intelligent control strategies for autonomous tasks. *Cogent Engineering*, *9*(1), p.2050020.

[8] Venkataraman, H. and Ramachandraprabu, S.P., 2025. A Hybrid Deep Learning and Edge Computing Framework for Real-Time IoT Data Processing. *Journal of Computer Applications and Information Technology*.

[9] Yan, W.Q., 2026. *Robotic Vision: From Deep Learning to Autonomous Systems*. Springer Nature.

[10] Raj, R. and Kos, A., 2025. An extensive study of convolutional neural networks: Applications in computer vision for improved robotics perceptions. *Sensors*, *25*(4), p.1033.

[11] Mahajan, H.B., Uke, N., Pise, P., Shahade, M., Dixit, V.G., Bhavsar, S. and Deshpande, S.D., 2023. Automatic robot Manoeuvres detection using computer vision and deep learning techniques: a perspective of internet of robotics things (IoRT). *Multimedia Tools and Applications*, *82*(15), pp.23251-23276.

[12] Upadhyay, R., Asi, A., Nayak, P., Prasad, N., Mishra, D. and Pal, S.K., 2023. Real-time deep learning–based image processing for pose estimation and object localization in autonomous robot applications. *The International Journal of Advanced Manufacturing Technology*, *127*(3), pp.1905-1919.

[13] Paniego, S., Shinohara, E. and Cañas, J., 2024. Autonomous driving in traffic with end-to-end vision-based deep learning. *Neurocomputing*, *594*, p.127874.

[14] Zalewski, J. and Hożyń, S., 2024. Computer vision-based position estimation for an autonomous underwater vehicle. *Remote Sensing*, *16*(5), p.741.

[15] Zheng, S., Wang, J., Rizos, C., Ding, W. and El-Mowafy, A., 2023. Simultaneous localization and mapping (slam) for autonomous driving: Concept and analysis. *Remote Sensing*, *15*(4), p.1156.

[16] Bablu, T.A. and Rashid, M.T., 2025. Edge computing and its impact on real-time data processing for IoT-driven applications. *Journal of advanced computing systems*, *5*(1), pp.26-43.