

# Multi-directional Bicycle Robot for Bridge Inspection with Steel Defect Detection System

Habib Ahmed, Son Thanh Nguyen, Duc La, Chuong Phuoc Le and Hung Manh La, *IEEE Senior Member*

**Abstract**—This paper presents a novel design of a multi-directional bicycle robot, which is developed for the inspection of steel structures, in particular, steel-reinforced bridges. The locomotion concept is based on arranging two magnetic wheels in a bicycle-like configuration with two independent steering actuators. This configuration allows the robot to possess multi-directional mobility. An additional free joint helps the robot adapt naturally to non-flat and complex steel structures. The robot's design provides the advantage of being mechanically simple and providing high-level mobility across diverse steel structures. In addition, a visual sensor is equipped that allows the data collection for steel defect detection with offline training and validation. The paper also provides a novel pipeline for Steel Defect Detection, which utilizes multiple datasets (one for training and one for validation) from real bridges. The quantitative results have been reported for three Deep Encoder-Decoder Networks (i.e., LinkNet, UNet, DeepLab) with their corresponding Encoder modules (i.e., ResNet-18, ResNet-34, RegNet-X2, EfficientNet-B0, and EfficientNet-B2). Due to space concerns, the qualitative results have been outlined in Appendix, with a link in Fig. 11 caption to access the result provided.

## I. INTRODUCTION

Steel structures are indispensable parts of modern civilization. Typical civil infrastructures include bridges, wind turbines, electric towers, and oil rigs, to name a few. Those steel structures require regular maintenance to ensure the safety and protection of the human population using these structures. Professional human inspectors still conduct most structural inspection tasks with non-destructive evaluation (NDE) sensors. This procedure is usually time-consuming, costly, and unsafe for the inspectors, especially when inspecting inaccessible regions of steel structures. For this reason, this research focuses on developing a novel robotic solution for steel bridge inspection with a novel video processing pipeline for Steel Defect Detection. Autonomous steel bridge inspection with varying sensory modalities has been discussed in the recent past [1]–[15]. Several innovative designs have been discussed, ranging from conventional wheeled robots [16]–[20], to robots that leverage tank-like tracks to

This work is supported by the U.S. National Science Foundation (NSF) under grants NSF-CAREER: 1846513 and NSF-PFI-TT: 1919127, and the U.S. Department of Transportation, Office of the Assistant Secretary for Research and Technology (USDOT/OST-R) under Grant No. 69A3551747126 through INSPIRE University Transportation Center, and the Vingroup Joint Stock Company's Vingroup Innovation Foundation (VINIF) under project code VINIF.2020.NCUD.DA094. The views, opinions, findings, and conclusions reflected in this publication are solely those of the authors and do not represent the official policy or position of the NSF, the USDOT/OST-R, and the VINIF.

The first two authors have made equal contributions. The authors are with the Advanced Robotics and Automation (ARA) Lab, Department of Computer Science and Engineering, University of Nevada, Reno, NV 89557, USA. Corresponding author: Hung La, email: hla@unr.edu.

widen the contact area of the robot on steel surfaces [2], [21]–[23]. These approaches work well on structures with large, flat, predictable surfaces such as ships or water tanks. However, these approaches may not work as efficiently on complex steel structures, e.g., bridges and oil rigs. Some unique developments of climbing robots in this field have also been demonstrated by [1], [24], [25].

Many designs have attempted to imitate the mobility of climbing animals, e.g., arachnids, quadrupedal animals, and inchworms. A spider-like robot with electromagnets has been developed by [26]. Another research highlighted the use of legged robots for structural inspection [27]. Other bio-inspired robots include an inchworm-like robot [28] and a hybrid robot [3]. However, most of these platforms' mechanical complexity is challenging to control in real-world applications. There are two robotic platforms, which resemble the bicycle-like design [29], [30]. Unfortunately, these platforms cannot traverse through complex structures and limited spaces. Significant differences exist between the proposed multi-directional bicycle robot and the platforms discussed in [29], [30], which will be discussed at the end of this section.

While the literature on steel corrosion detection is limited, some relevant studies have presented their approach. A study by [31] uses roughness analysis and color comparison on image patches to separate corrosion patches for steel images. This study makes a trade-off between the recall and precision levels where one is high while the other is low. Another study [32] made use of texture analysis with variables such as contrast, correlation, and energy. The study by [33] is used for crack and corrosion detection, which made use of a supervised classification method with a code-word dictionary consisting of stacked RGB histograms for image patches symmetric gray-level co-occurrence matrix for each patch.

Our corrosion detection system takes a different approach using deep learning. Instead of creating automatic corrosion classifiers, we use Neural Network to self-extract features by combining many Deep Encoder-Decoder Networks with their corresponding Encoder modules to find the optimal steel defect detection system.

This research provides several novel contributions to state-of-the-art steel bridge inspection and defect detection. This will enable the proposed research to impact the relevant research community substantially. The novel elements of the proposed study are discussed as follows:

(a). **Robotic Platform:** The proposed Multi-Directional Bicycle Robot provides an improved climbing capability and multi-directional locomotion; these capabilities were

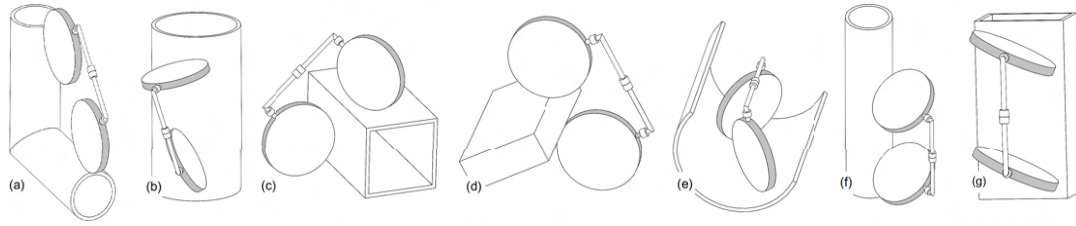


Fig. 1. (a) Mode 1 (bicycle-like): The robot can handle cylindrical structures with limited contacting areas. (b) Mode 1: The robot changes the direction by first stopping the back wheel. Then, the front steering servo turns 90 degrees, and finally, the front wheel moves to help the robot change direction. (c) Mode 1: With a free joint, the robot can travel on two intersecting surfaces. (d) Mode 1: The robot can traverse on edges that are thicker than the space between its two wheels (4cm). (e) Mode 1: The robot is flexible enough to travel on the internal surface of a tube. (f) Mode 2: Two steering servos turn the wheels at the same angle. The robot moves spirally around a circular tube. In this mode, the robot can also perform well on tube shapes such as rectangles or hexagons. (g) Mode 2: The robot can rotate around its body center or move sideways (left, right) at steering angles that are close to 90 degrees.

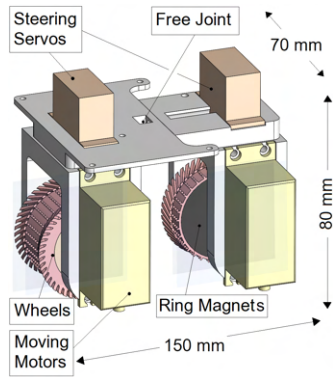


Fig. 2. The 3D mechanical design of our proposed robot.

lacking in [2]–[5], [29], [30]. These capabilities will allow the proposed robotic platform to traverse challenging steel structures. The use of permanent magnetic wheels and the adhesive forces generated has allowed the proposed robotic platform to be equipped with diverse sensory modalities. In conclusion, this robotic platform combines design simplicity with diversity and versatility in traversing complex, challenging steel structures.

(b). **Steel Defect Detection System:** In this research, a novel video processing pipeline for Steel Defect Detection System has been proposed. This video processing pipeline utilizes some of the state-of-the-art Deep Encoder-Decoder Networks with varying Encoder modules, which no other defect detection studies have used. A diverse range of performance metrics has been utilized to examine and compare the performance of the different Architecture-Encoder pairs. Offline training has been utilized, along with validation of video frame processing, which has considerable implications for the real-time processing of the Steel Defect Detection System. The datasets collected and utilized in this research came from many different sources. For Steel Defect Detection System, 4,500 high-resolution image data (dataset 1) was collected by research colleagues from multiple bridges located in Vietnam. The 1,500 image-based data (dataset 2) for validation were collected by authors using a Multi-Directional Bicycle Robot on a Highway-80 bridge in Lovelock, Nevada, USA.

## II. NOVEL MULTI-DIRECTIONAL BICYCLE ROBOT: MECHANICAL DESIGN AND ANALYSIS

Going into the brief details of the proposed mechanical design of the bicycle robot, the following requirements should be addressed and fulfilled by the proposed mechanical design of the bicycle robot:

- 1) The robot can climb surfaces with a wide range of outer diameters ( $\geq 150\text{mm}$ ), which are normally encountered on circular tubes or cylindrical surfaces;
- 2) The robot can pass convex or concave obstacles at structural transition joints on truss structures;
- 3) The robot can travel on steel structures with complex arrangements of obstacles such as bolts, nuts, and gaps;
- 4) The locomotion system can maneuver through narrow areas ( $\geq 100\text{mm}$  wide) and can move sideways with considerable flexibility.

The high mobility of the two steering actuators allows the robot to operate in two different modes. *Mode 1* shown in Fig. 1 (a-e) that supports only one steering unit and the robot works like a bicycle, which facilitates the robot to travel on structures with limited contacting areas Fig. 1 (a), and ability to change direction simultaneously Fig. 1 (b). The free joint in the middle allows the robot to traverse two intersecting surfaces Fig. 1 (c). The robot can also pass edges thicker than the space between its two wheels Fig. 1 (d), and traverse the internal surface of hollow cylindrical tube-like structures Fig. 1 (e). *Mode 2* as shown in Fig. 1 (f-g) allows both steering units to remain active with independent and parallel control, facilitating the robot to move spirally (around a cylinder), sideways (left or right) or rotate around its center. Fig. 2 shows the overall mechanical design of the bicycle robot. The robot's weight is 1 kg (without sensors), while it can carry 600g of load (sensors, onboard computer, etc.). Plastic is the primary material to ensure the frame is lightweight. The robot is powered by a 3000mAh LiPo battery that allows 1 hour of operation. The robot's physical dimensions are  $150\text{mm} \times 80\text{mm} \times 70\text{mm}$ . The ring magnets are placed at the cores of the wheels, which are covered by silicone tires. The wheels are driven by two high-torque gear DC motors (100kg-cm torque each), and the steering actuators are controlled by two servo motors (32kg-cm torque each). The front and back of the frame are linked by a bearing

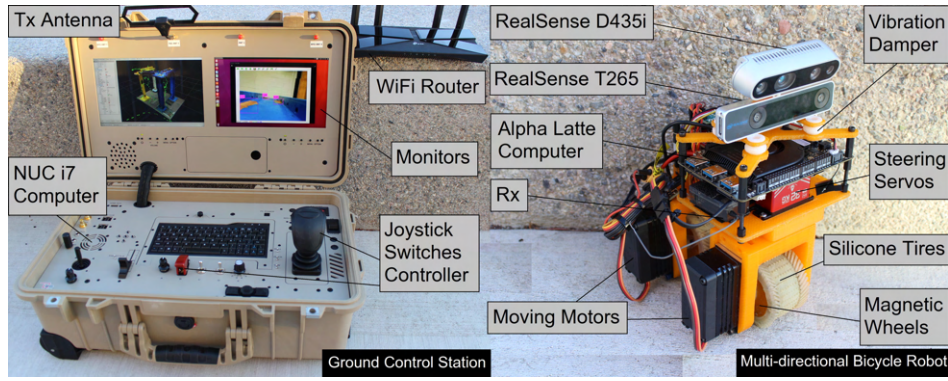


Fig. 3. Multi-directional Bicycle Robot is controlled remotely from a Ground Control Station (GCS) using a joystick via a radio channel. The robot carries a depth camera D435i and a pose tracking camera T265. An Alpha Latte computer onboard saves raw data from the sensors and continuously transfers the data back to GCS using Intel NUC i7 via a WiFi router. The GCS performs localization and object detection and visualizes the received data online on its two screens.

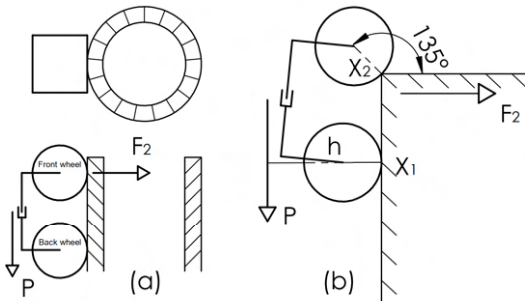


Fig. 4. The situations where the adhesive force is minimal, resulting in a high chance of falling: a) the robot traverses on a thin cylinder (15cm diameter). The contacting area reduces to only a single point for each wheel; b) The adhesive force of the front wheel is significantly reduced when the robot hits an edge.

acting as a free joint. The design concept of our robot is inspired by a bicycle, which involves using two revolute joints equipping the robot with two independent steering actuators, increasing its mobility considerably. An additional free joint in the middle of the robot's body allows its two wheels to make full contact with surfaces of different shapes and sizes. The moving wheels are designed with permanent ring-shaped magnets to generate large adhesive forces.

#### A. Magnetic Wheel Force Analysis

We analyze the robot's adhesive force to climb reliably in normal working conditions. We also perform the analysis in an extreme situation where the adhesive force between the magnetic wheels and the contacting structures is minimal: the robot climbs a cylindrical tube (Fig. 4a) and encounters a corner (Fig. 4b). Here,  $X_1$  and  $X_2$  are two contacting points of the back wheel and the front wheel, respectively. Given that  $P$  is the robot's weight,  $h$  is the distance from the robot's center of mass to  $X_1$ . If  $F_2$  is an adhesive force of the front wheel at  $X_2$ , then  $F_2$  is at its minimum when the front wheel hits the corner. To keep the robot from falling, the following condition needs to be satisfied:

$$F_2 \times X_1 X_2 > P \times h \Rightarrow F_2 > \frac{Ph}{X_1 X_2}. \quad (1)$$

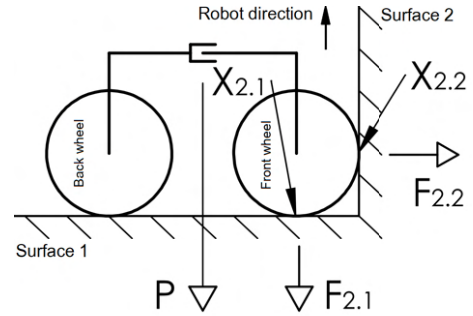


Fig. 5. When the robot passes an internal corner between two surfaces, the robot's load increases significantly.

According to ISO 3691 [34] for safe weight lifting, a safety factor of 5 was selected. Therefore, the real adhesive force  $F_2$  needs to be at least 5 times greater than the result from the above theoretical calculation (1).

#### B. Moving Motors Power Analysis

This analysis calculates the necessary motor torque when the robot stands the highest load. It is when the robot passes an internal corner between two surfaces (Fig. 5) that the front wheel bears an additional force  $F_{2,2}$ , which is the adhesive force of the front wheel at the surface 2. Similarly,  $F_{2,1}$  is the adhesive force of the front wheel at surface 1.  $F_{f2}$  is the friction of the front wheel on surface 2,  $r$  is the wheel's radius, and  $k$  is the static friction coefficient (between silicon and steel in our design). The minimum force of the front wheel that allows the robot to be able to pass the corner must satisfy:

$$\frac{M_{moving}}{r} > F_{2,1} + F_{f2} + \frac{P}{2}. \quad (2)$$

Therefore, the moving motor torque needs to satisfy:

$$M_{moving} > r \times (F_{2,1} + kF_{2,2} + \frac{P}{2}). \quad (3)$$

According to IEC 60034 [35], the actual torque selected to be at least double that of the theoretical calculation in (3).

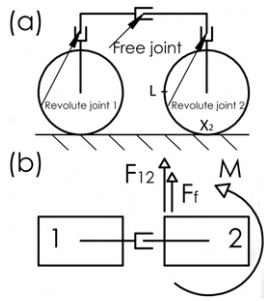


Fig. 6. An experiment is conducted to investigate the load on a steering servo motor. A dynamo-meter is mounted on one wheel's edge (point  $L$ ) to measure the load. The distance from  $L$  to the rotating point  $X_2$  is  $r$  (the wheel's radius). a) Side view. b) Top view.

### C. Steering Servos Power Analysis

An analysis is conducted to investigate the load torque on the revolute joints. There are two forces, as illustrated in Fig. 6: the static friction and the attractive force at the two magnetic wheels, with one affecting another. Let  $F_{12}$  be the adhesive force of wheel 1 affecting wheel 2,  $F_f$  be the friction at  $X_2$ . The measured load-force at point  $L$  (Fig. 6a) has to satisfy the following condition:

$$F_{12} + F_f < \frac{M_{steering}}{r}. \quad (4)$$

Thus, the steering servo torque needs to satisfy:

$$\Rightarrow M_{steering} > r \times (F_{12} + \frac{F_2 + P}{k}). \quad (5)$$

Based on IEC 60034 [35], the actual servo's torque is chosen to be at least two-fold compared to that of the theoretical calculation in (5). A complete analysis of other conditions can be seen in a similar robot design reported in [36].

## III. PROPOSED MODEL FOR STEEL BRIDGE DEFECT DETECTION SYSTEM

### A. Sensor Systems for Steel Bridge Defect Detection

The whole system is depicted in Fig. 3. On the robot side, we utilize an Intel RealSense D435i camera, which provides color and depth images. However, only the RGB color image data was used in this study to develop the Steel Defect Detection System. A LattePanda Alpha 864 is selected as an onboard computer, which connects the ground control station (GCS) computer (Intel NUC i7) via a WiFi router to the robot to enable manual steering operation of the robot using the GCS. The two computers form a Robot Operating System [37] network, in which the GCS acts as the master.

### B. Dataset for Steel Bridge Defect Detection

There are two datasets collected and used within this study to train and validate the performance of the Steel Bridge Defect Detection System. One of the datasets has been collected by the research colleagues of the authors from several different bridges located in Vietnam. The data was collected when the overall lighting conditions were optimal. Due to the high-quality and high-resolution of the original data, each large image was cropped and separately saved with image dimensions of  $512 \times 512 \times 3$ ; these dimensions

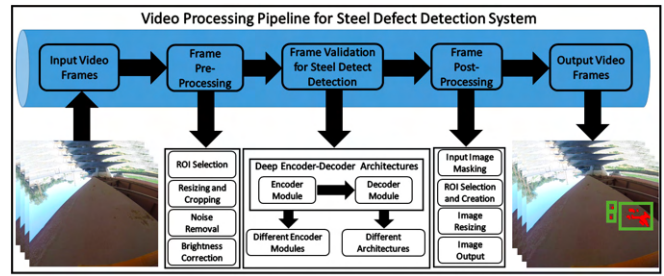


Fig. 7. Details regarding the Video Processing Pipeline for Steel Defect Detection System starting from Input Video Frames, which are put through several image pre-processing steps. The system is based on offline training and validation processes; some pre- and post-processing steps involve manual intervention from the human researcher. Only a portion of the high-resolution input frame is selected to be validated in later stages of the video processing pipeline. The different pre-processing steps ensure the quality of the input video frames is enhanced. The output image frame from the Deep Encoder-Decoder Network is put through different post-processing operations. Finally, the system's output is highlighted using red-colored pixel-level defects on the steel surface, along with green-colored bounding boxes.

are supported by the Deep Learning models that are being leveraged within the overall Video processing pipeline, which will be discussed in the next subsection. Consequently, 4,500 high-quality images were obtained using a high-definition hand-held digital camera with different steel bridge defects. The authors collected the second dataset during the bi-annual testing and evaluation of the bridges by the Nevada Department of Transportation (NDOT) at Highway-80, Lovelock, Nevada, USA. The images were obtained by operating the Bicycle Robot on the different steel parts of the bridge. The original images were cropped, resized, and saved with a resolution of  $512 \times 512 \times 3$ . This dataset, with a total of 1,500 images, has been used to validate the performance of the Steel Defect Detection System.

### C. Video Processing Pipeline for Steel Defect Detection System

In this section, a detailed evaluation of the video processing pipeline for the Steel Defect Detection System will be discussed. The complete block diagram of the video processing pipeline has been given in Fig. 7. As shown in Fig. 7, the video processing pipeline has five steps. Starting from the input video frames, which are individually pre-processed using many steps, e.g., the Region-of-Interest (ROI) selection. The original size of the high-resolution image frame is very large, due to which a selected region is separated. This ensures that the background regions are separated, and most steel regions close to the robot can be cropped, resized, and saved separately. The image ROI is resized to  $512 \times 512 \times 3$ ; the input size permitted for validating the input video frames using Deep Encoder-Decoder Networks.

These networks are pre-trained on the Vietnam bridge dataset. A variety of state-of-the-art Deep Encoder-Decoder Networks architectures have been used in this study, namely the U-Net [38], LinkNet [39] and DeepLabV3 [40]. Several different Encoder modules are leveraged for each of these architectures to examine and compare the performance of the

TABLE I

| System Specifications | System 1   | System 2   |
|-----------------------|--|--|
| Processor             | Intel® NUC10i7FNH1<br>Core i7 with 1.10 GHz<br>clock speed | Intel® Core i7-8700<br>CPU with 3.2 GHz<br>clock Speed |
| RAM                   | 16GB SDRAM   | 32GB SDRAM   |
| ROM                   | 256GB SSD  | N/A  |
| Hard Disk             | 1 TB HDD   | 350 GB HDD   |
| Operating System      | Ubuntu 20.04   | Ubuntu 18.04   |
| GPU                   | Intel® Integrated<br>UHD Graphics                          | NVIDIA® GeForce®<br>GTX 1080 TI GPU                    |

different Architecture-Encoder pairs. Some of the Encoders used in this study include the ResNet-18 [41], ResNet-34 [41], EfficientNet-b0 [42], EfficientNet-b2 [42], and RegNet-X2 [43].

One of the prime focuses was selecting Encoder modules that are not very large in terms of the number of layers and parameters. The output image from this stage in the video processing pipeline contains pixel-level masks highlighting steel defect locations. This output is modified to ensure that the predicted defect locations are highlighted using red pixels and a green color bounding box surrounding each pixel region.

#### D. Performance Evaluation for Video Processing Pipeline for Steel Defect Detection

For examining the performance of Deep Encoder-Decoder Networks, several different metrics have been used in the prior studies [38] [39] [40]. The data was annotated using pixel-level annotation, which is examined differently than bounding-box-based approaches for data annotation. In this research, some of the most important metrics will be used, which include F1-score, mean Intersection-over-Union (mIOU), Precision, and Recall. These metrics will enable the researchers to assess the accuracy of validation results compared to ground truth towards steel defect detection. At the same time, there is also a need to know the potential of the different Architecture-Encoder pairs towards Pipeline for Steel Defect Detection. The validation time will also be discussed in the proceeding section to examine this particular aspect. The performance evaluation of the different Architecture-Encoder pairs will also be examined, along with their real-time capabilities, to highlight the most suitable pair out of all the Encoder modules and Deep Encoder-Decoder architecture.

#### E. Systems Configuration

Two different types of systems were used to examine the performance of the Video Processing Pipeline for Steel Defect Detection. The training was conducted offline on System 2, equipped with an onboard GPU with details in Table I. The different Deep Learning models trained for varying Architecture-Encoder pairs were saved. The validation process was performed on two separate systems to examine whether the validation process could be performed in real-time for the two different PCs with varying system configurations. Table I highlights the different aspects of

TABLE II

| LinkNet [39] Architecture |      |             |              |              |              |
|---------------------------|------|-------------|--------------|--------------|--------------|
| Encoder                   |      | Dice Loss   | mIOU         | Precision    | Recall       |
| ResNet-18 [41]            | Max. | 16.80       | 97.33        | 99.74        | 97.68        |
|                           | Min. | 1.41        | 73.65        | 99.56        | 73.69        |
|                           | Avg. | 6.11        | 89.89        | 99.65        | 90.07        |
| ResNet-34 [41]            | Max. | 32.96       | 94.55        | 99.93        | 94.91        |
|                           | Min. | 4.23        | 53.44        | 99.56        | 53.14        |
|                           | Avg. | 12.20       | 82.25        | 99.75        | 82.31        |
| RegNet-X-2 [43]           | Max. | 19.91       | 98.52        | <b>99.77</b> | <b>99.80</b> |
|                           | Min. | <b>0.08</b> | 69.32        | 99.56        | 69.47        |
|                           | Avg. | 6.81        | 89.11        | 99.67        | 90.37        |
| Efficient-b0 [42]         | Max. | 31.20       | 94.17        | 99.78        | 94.45        |
|                           | Min. | 3.28        | 57.56        | 99.53        | 57.39        |
|                           | Avg. | 11.51       | 82.89        | 99.66        | 82.99        |
| Efficient-b2 [42]         | Max. | 16.83       | <b>99.36</b> | 99.73        | 99.78        |
|                           | Min. | 0.32        | 73.30        | 99.58        | 73.44        |
|                           | Avg. | 4.93        | 91.95        | 99.64        | 92.24        |

the two types of PCs used to examine the performance of pre-trained models in terms of providing real-time steel defect detection. It can be seen from Table I that system 1 has Intel® Integrated UHD Graphics Card, which is not supported by Nvidia® CUDA® libraries leading to slower validation time. In comparison, the onboard GPU within system 2 had full support from the Nvidia® CUDA® libraries, which allowed a faster training and validation processing time, which will be elaborated on in the next section.

## IV. RESULTS AND DISCUSSION

### A. Robot locomotion

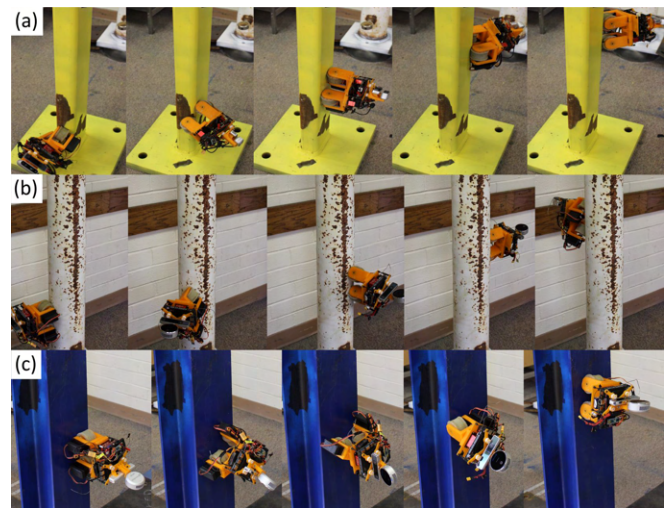


Fig. 8. a) A test of moving around thin rectangular tube: robot has to combine mode 1 and 2 for smooth travel. b) A test of moving sideways in mode 2: robot travels spirally outside a cylindrical structure. c) A turning test in mode 1: robot's head is turned 180 degrees on a narrow surface. The movement is depicted from left to right.

An indoor structure comprised of typical parts of general steel structures (cylinder, L, I, U-shaped beams) with structural transition joints is built to validate the robot's locomotion. Our robot can traverse smoothly to any location in the testing structure. In mode 1, the robot can handle well most testing situations. However, mode 2 is necessary when

operating in narrow spaces. For instance, as Fig. 8) shows, the robot must combine modes 1 and 2 to travel smoothly on a rectangular tube. Fig. 8b illustrates mode 2 while the robot is working on a cylindrical shape. Fig. 8c illustrates how the robot changes orientation in mode 1 on an I-shaped beam.

We deployed the robot in a field test during an annual Highway inspection in Nevada, USA. The bridge is located on Highway-80 in Lovelock city. Fig. 9 shows robot performance in an actual application, and the robot was able to traverse stably on the structure and collect visual data of this bridge.

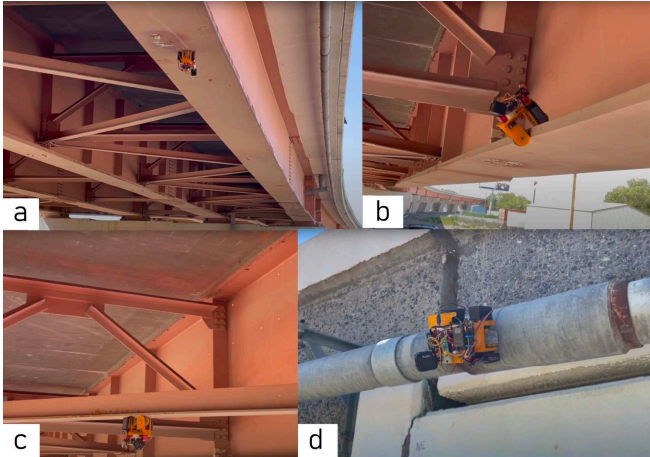


Fig. 9. Robot deployment on a bridge on Highway-80 located in Lovelock City, NV, USA. a) Robot traverses upside down on the main I-shape beam; b) Robot passes an edge of I-shape beam; c) Robot moves upside down on a T-shape member; d) Robot travels stably on bridge water draining tube.

### B. Corroded Detection

This section will outline the overall performance of the different Architecture-Encoder pairs. The different Architecture-Encoder pairs were trained on the Vietnam dataset. After offline training of the different Architecture-Encoder pairs, the validation was performed on six separate video frame data (1,500 images) obtained using Bicycle Robot taken from the actual bridge on Highway-80 located in Lovelock, NV, USA. There are variations in the level of defect and lightning conditions on the different parts of the bridge, reflecting the performance of the different Architecture-Encoder pairs. In order to capture this variation in performance, three separate rows for each Architecture-Encoder pair outlining the minimum, maximum and average values is given in Tables II, III and IV. Tables II, III and IV outline statistical evaluation for the different Architecture-Encoder pairs in terms of the different metrics, such as Dice Loss, mIoU, Precision, and Recall. Higher values reflect better performance for metrics such as mIoU, Precision, and Recall, while lower values reflect better performance for Dice Loss. The bold values in Tables II, III and IV specify the highest value, or lowest in case of Dice Loss, for a particular Architecture. The bold values with an underline specify the best performance value compared to all the different Architecture-Encoder pairs.

TABLE III

| UNet [38] Architecture |      |             |              |              |              |
|------------------------|------|-------------|--------------|--------------|--------------|
| Encoder                |      | Dice Loss   | mIOU         | Precision    | Recall       |
| ResNet-18 [41]         | Max. | 31.80       | 91.86        | <b>99.92</b> | 91.59        |
|                        | Min. | 4.37        | 54.87        | 99.54        | 54.86        |
|                        | Avg. | 12.59       | 80.88        | 99.73        | 81.02        |
| ResNet-34 [41]         | Max. | 28.11       | 96.40        | 99.83        | 96.57        |
|                        | Min. | 1.96        | 59.40        | 99.56        | 59.43        |
|                        | Avg. | 11.11       | 83.47        | 99.72        | 82.13        |
| RegNet-X-2 [43]        | Max. | 18.81       | 97.13        | 99.78        | <b>99.35</b> |
|                        | Min. | 1.59        | 71.56        | 99.55        | 71.71        |
|                        | Avg. | 7.26        | 88.01        | 99.65        | 87.06        |
| Efficient-b0 [42]      | Max. | 32.17       | <b>97.33</b> | 99.80        | 97.53        |
|                        | Min. | <b>1.41</b> | 55.85        | 99.53        | 55.92        |
|                        | Avg. | 11.44       | 83.26        | 99.61        | 83.46        |
| Efficient-b2 [42]      | Max. | 47.25       | 96.06        | 99.75        | 96.36        |
|                        | Min. | 2.18        | 43.56        | 99.56        | 43.60        |
|                        | Avg. | 14.39       | 69.84        | 99.65        | 81.87        |

TABLE IV

| DeepLab [40] Architecture |      |             |              |              |              |
|---------------------------|------|-------------|--------------|--------------|--------------|
| Encoder                   |      | Dice Loss   | mIOU         | Precision    | Recall       |
| ResNet-18 [41]            | Max. | 26.46       | 95.50        | <b>99.80</b> | 95.58        |
|                           | Min. | 2.52        | 62.24        | 99.55        | 62.02        |
|                           | Avg. | 9.05        | 86.26        | 99.68        | 86.26        |
| ResNet-34 [41]            | Max. | 26.46       | 93.76        | 99.82        | 93.82        |
|                           | Min. | 3.36        | 61.19        | 99.55        | 61.06        |
|                           | Avg. | 10.45       | 84.30        | 99.68        | 64.14        |
| RegNet-X-2 [43]           | Max. | 15.36       | <b>97.58</b> | 99.78        | <b>97.85</b> |
|                           | Min. | <b>1.30</b> | 75.79        | 99.56        | 75.94        |
|                           | Avg. | 6.71        | 89.41        | 99.69        | 90.02        |
| Efficient-b0 [42]         | Max. | 22.89       | 96.21        | 99.59        | 96.56        |
|                           | Min. | 1.99        | 65.52        | 99.55        | 65.57        |
|                           | Avg. | 9.40        | 85.13        | 99.55        | 85.39        |
| Efficient-b2 [42]         | Max. | 40.91       | 90.06        | 99.85        | 90.24        |
|                           | Min. | 5.76        | 48.12        | 99.46        | 48.16        |
|                           | Avg. | 17.12       | 75.38        | 77.17        | 55.55        |

It can be seen in Table II that for LinkNet [39] Deep Encoder-Decoder architecture, the most optimal performance has been shown by the RegNet-X2 [43] Encoder module for four out of five total performance metrics. For performance regarding UNet [38] Architecture, EfficientNet-B0 [42] outperforms other Encoder modules with the best performance for two out of four metrics.

For Table IV with details regarding DeepLab Architecture [40], RegNet-X2 [43] encoder performs the most optimal for three out of four performance metrics.

Fig. 10 highlights a side-by-side comparison between the validation time between System 1 and System 2. The values for each Architecture-Encoder Pair have been highlighted on top of each bar plot in Fig. 10. For system 1, the lowest values for validation are outlined by LinkNet architecture [39]. For system 2, the lowest values for validation time have been reported by UNet [38] architecture and associated Encoder modules. One of the architectures, namely LinkNet [39] has demonstrated the lowest validation time results for system 1. However, system 2 has the highest values out of all the other Architectures, which is an interesting contrast in the validation time results. Figure 11 contains image output from three Architecture-Encoder pairs that demonstrated the most optimal results in terms of qualitative aspects in terms of similarity with the ground truth and minimal presence of

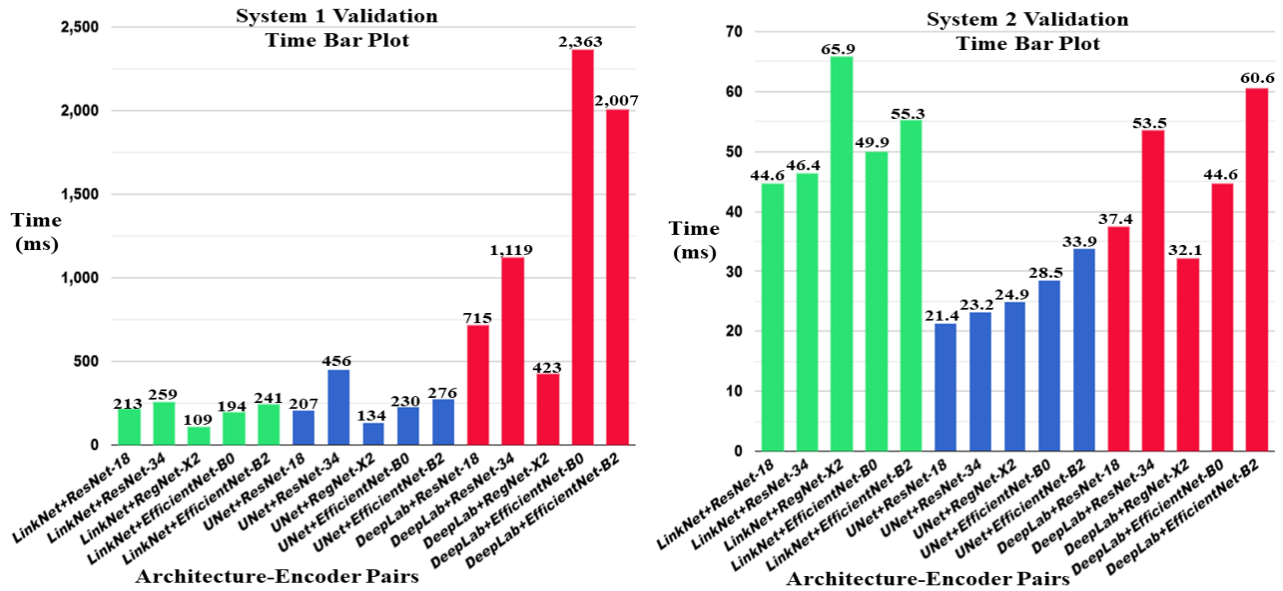


Fig. 10. A side-by-side comparison between the validation time for System 1 and System 2, which is mentioned in Table I. There are wide variations between the validation times for each image frame, as the maximum value for System 1 is 2,007 ms, and the maximum value for System 2 is 65.95 ms, which highlights the benefits of GPU for real-time steel defect detection. When comparing the different Architectures for System 1, the variations are more pronounced across architectures. For the case of System 2, the variations are much less pronounced, with the highest values ranging between 20 ms and 70 ms.

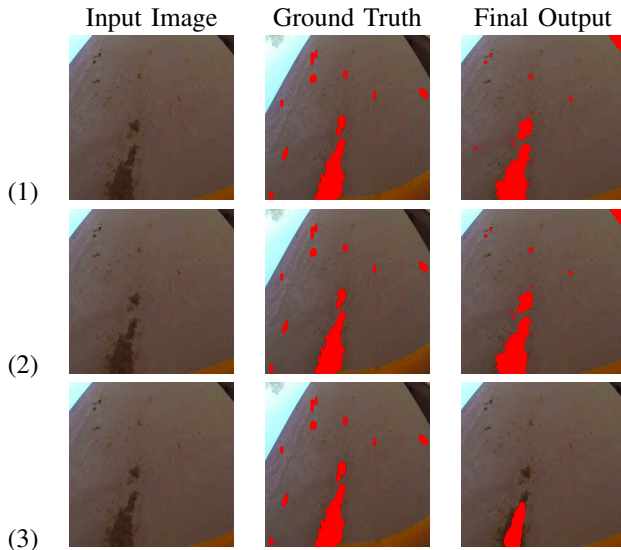


Fig. 11. The first column is the original images. The second column is the ground truths annotated with red color. The third column is the final outputs of the three Architecture-Encoder pairs that demonstrated the most optimal results. (1) is for LinkNet ResNet-18, (2) is for Unet ResNet-18, and (3) is for DeepLab ResNet-18. Link for more validation: <http://ara.cse.unr.edu/wp-content/uploads/2014/12/Appendix.pdf>

false-positive and true negatives. It also contains the link for more validations information.

When comparing the results in the other studies [31], [33] in terms of depth of evaluation and the metrics used within this study, the performance of the proposed system far surpasses other studies highlighted with demonstrable high-performance using quantitative and qualitative analysis. [31] has an average recall of 85% while [33] has an average precision and recall of 83% and 96%. Our system's precision

is well over 99% and recall of over 95%. In addition, other studies use a small set of uniformly illuminated and without misleading objects, whereas we use thousands of diverse images from 2 different bridges. Comparison to [32] is not applicable due to the difference in metric. Video demonstration of the robot test and deployment can be seen at: <https://youtu.be/kAQnpFI2huI>

## V. CONCLUSION AND FUTURE WORK

This paper presents the development of a novel multi-directional bicycle robot capable of climbing different steel structures to perform structural inspection and defect detection. The robot's novel configuration allows multi-directional mobility. The robot's design provides the advantage of being mechanically simple and provides a high level of mobility across diverse steel structures. The practical operation of the multi-directional robot has been validated on an actual bridge (Highway-80, Lovelock, NV, USA), which shows that the robot can firmly adhere to steel structures and operates well in tight spaces.

The onboard visual sensor is used to collect data from outdoor environments. This study used two novel datasets containing data from two separate sets of bridges. The quantitative results demonstrate a considerable promise of the proposed system for real-time processing with reliable performance for different Architecture-Encoder pairs. The link to Appendix also reveals compelling results regarding qualitative aspects of the proposed Steel Defect Detection System. In future works, an effort will be made to test and validate each Architecture-Encoder pair in real-time to assess the data processing issues and the effect of environmental conditions on the overall performance.

## REFERENCES

- [1] S. T. Nguyen and H. M. La, "Roller chain-like robot for steel bridge inspection," in *the 9th International Conference on Structural Health Monitoring of Intelligent Infrastructure*, 2019, pp. 1–6.
- [2] —, "Development of a steel bridge climbing robot," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2019, pp. 1912–1917.
- [3] S. T. Nguyen, A. Q. Pham, C. Motley, and H. M. La, "A practical climbing robot for steel bridge inspection," in *Proceedings of the 2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 9322–9328.
- [4] H.-D. Bui, S. Nguyen, U.-H. Billah, C. Le, A. Tavakkoli, and H. M. La, "Control framework for a hybrid-steel bridge inspection robot," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020, pp. 2585–2591.
- [5] S. T. Nguyen and H. M. La, "A climbing robot for steel bridge inspection robot," in *Journal of Intelligent and Robotic Systems*. Springer, 2021, pp. 1–21.
- [6] H. Ahmed, H. M. La, and N. Gucunski, "Review of non-destructive civil infrastructure evaluation for bridges: State-of-the-art robotic platforms, sensors and algorithms," in *Sensors*, 2020, pp. 1–35.
- [7] H. M. La, N. Gucunski, K. Dana, and S.-H. Kee, "Development of an autonomous bridge deck inspection robotic system," *Journal of Field Robotics*, vol. 34, no. 8, pp. 1489–1504, 2017. [Online]. Available: <https://onlinelibrary.wiley.com/doi/abs/10.1002/rob.21725>
- [8] H. M. La, N. Gucunski, Seong-Hoon Kee, J. Yi, T. Senlet, and Luan Nguyen, "Autonomous robotic system for bridge deck data collection and analysis," in *2014 IEEE/RSJ Intern. Conf. on Intelligent Robots and Systems*, Sep. 2014, pp. 1950–1955.
- [9] H. M. La, T. H. Dinh, N. H. Pham, Q. P. Ha, and A. Q. Pham, "Automated robotic monitoring and inspection of steel structures and bridges," *Robotica*, pp. 1 – 21, 2018.
- [10] H. M. La, R. S., B. B. Basily, N. Gucunski, J. Yi, A. Maher, F. A. Romero, and H. Parvardeh, "Mechatronic systems design for an autonomous robotic system for high-efficiency bridge deck inspection and evaluation," *Mechatronics, IEEE/ASME Transactions on*, vol. 18, no. 6, pp. 1655–1664, Dec 2013.
- [11] H. M. La, N. Gucunski, S.-H. Kee, and L. Nguyen, "Data analysis and visualization for the bridge deck inspection and evaluation robotic system," *Visualization in Engineering*, vol. 3, no. 1, pp. 1–16, 2015.
- [12] H. M. La, N. Gucunski, S. Kee, and L. Nguyen, "Visual and acoustic data analysis for the bridge deck inspection robotic system," in *The 31st International Symposium on Automation and Robotics in Construction and Mining (ISARC)*, July 2014, pp. 50–57.
- [13] H. M. La, N. Gucunski, S.-H. Kee, J. Yi, T. Senlet, and L. Nguyen, "Autonomous robotic system for bridge deck data collection and analysis," in *IEEE Intern. Conf. on Intelligent Robots and Systems (IROS)*, Sept 2014, pp. 1950–1955.
- [14] R. S. Lim, H. M. La, and W. Sheng, "A robotic crack inspection and mapping system for bridge deck maintenance," in *IEEE Transactions on Automation Science and Engineering*. IEEE, 2014, pp. 367–378.
- [15] N. Gucunski, A. Maher, B. Basily, H. M. La, R. S. Lim, H. Parvardeh, and S. H. Kee, "Robotic platform robot for condition assessment of concrete bridge decks using multiple nde technologies," in *Journal of Croatian Society for Non-Destructive Testing*. Croatian Society for Non-Destructive Testing, 2013, pp. 5–12.
- [16] A. Sirken, G. Knizhnik, J. McWilliams, and S. Bergbreiter, "Bridge risk investigation diagnostic grouped exploratory (bridge) bot," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017, pp. 6526–6532.
- [17] D. Zhu, J. Guo, C. Cho, Y. Wang, and K. Lee, "Wireless mobile sensor network for the system identification of a space frame bridge," *IEEE/ASME Trans. on Mechatronics*, vol. 17, no. 3, pp. 499–507, June 2012.
- [18] J. Guo, W. Liu, and K.-M. Lee, "Design of flexonic mobile node using 3d compliant beam for smooth manipulation and structural obstacle avoidance," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2014, pp. 5127–5132.
- [19] S. Kamdar, "Design and manufacturing of a meccanum wheel for the magnetic climbing robot," *Master Thesis, Embry-Riddle Aeronautical University*, May 2015.
- [20] N. H. Pham and H. M. La, "Design and implementation of an autonomous robot for steel bridge inspection," in *2016 54th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*. IEEE, 2016, pp. 556–562.
- [21] W. Shen, J. Gu, and Y. Shen, "Permanent magnetic system design for the wall-climbing robot," in *IEEE International Conference Mechatronics and Automation, 2005*, vol. 4. IEEE, 2005, pp. 2078–2083.
- [22] G. Lee, G. Wu, J. Kim, and T. Seo, "High-payload cbing and transitioning by compliant locomotion with magnetic adhesion," *Robotics and Autonomous Systems*, vol. 60, no. 10, pp. 1308 – 1316, 2012.
- [23] T. Seo and M. Sitti, "Tank-like module-based climbing robot using passive compliant joints," *IEEE/ASME Transactions on Mechatronics*, vol. 18, no. 1, pp. 397–408, Feb 2013.
- [24] H. Eto and H. H. Asada, "Development of a wheeled wall-climbing robot with a shape-adaptive magnetic adhesion mechanism," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 9329–9335.
- [25] Y. Takada, S. Ito, and N. Imajo, "Development of a bridge inspection robot capable of traveling on splicing parts," *Inventions*, vol. 2, 2017.
- [26] T. Bandyopadhyay, R. Steindl, F. Talbot, N. Kottege, R. Dungavell, B. Wood, J. Barker, K. Hoehn, and A. Elfes, "Magneto: A versatile multi-bed inspection robot," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018, pp. 2253–2260.
- [27] A. Mazumdar and H. H. Asada, "Mag-foot: A steel bridge inspection robot," in *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2009, pp. 1691–1696.
- [28] P. Ward, P. Manamperi, P. R. Brooks, P. Mann, W. Kaluarachchi, L. Matkovic, G. Paul, C. H. Yang, P. Quin, D. Pagano, D. Liu, K. Waldron, and G. Dissanayake, "Climbing robot for steel bridge inspection: Design challenges," in *Austrroads Publications Online, ARRB Group*, 2015.
- [29] F. Tâche, W. Fischer, G. Caprari, R. Siegwart, R. Moser, and F. Mondada, "Magnebike: A magnetic wheeled robot with high mobility for inspecting complex-shaped structures," *Journal of Field Robotics*, vol. 26, no. 5, pp. 453–476, 2009.
- [30] G. Caprari, A. Breitenmoser, W. Fischer, C. Hürzeler, F. Tâche, R. Siegwart, O. Nguyen, R. Moser, P. Schoeneich, and F. Mondada, "Highly compact robots for inspection of power plants," *Journal of Field Robotics*, vol. 29, no. 1, pp. 47–68, 2012.
- [31] M. Khatayazad, L. De Pue, and W. De Waele, "Detection of corrosion on steel structures using automated image processing," in *Developments in the Built Environment*. Elsevier, 2020, pp. 1–12.
- [32] F. F. Feliciano, F. R. Leta, and F. B. Mainier, "Texture digital analysis for corrosion monitoring," in *Corrosion Science*, 2015, pp. 138–147.
- [33] F. Bonnin-Pascual and A. Ortiz, "Corrosion detection for automated visual inspection," in *Development in Corrosion Protection*. INTECH, 2014, pp. 619–632.
- [34] "Iso3691," <https://www.iso.org/obp/ui/#iso:std:iso:3691:-4:ed-1:v1:en/>.
- [35] "Iec60034," [https://global.ihs.com/doc\\_detail.cfm?document\\_name=IEC%2060034%2D1](https://global.ihs.com/doc_detail.cfm?document_name=IEC%2060034%2D1).
- [36] S. T. Nguyen, H. Nguyen, S. T. Bui, V. A. Ho, T. D. Ngo, and H. M. La, "An agile bicycle-like robot for complex steel structure inspection," in *2022 International Conference on Robotics and Automation (ICRA)*, 2022, pp. 157–163.
- [37] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, "Ros: an open-source robot operating system," in *ICRA workshop on open source software*, vol. 3. Kobe, Japan, 2009, p. 5.
- [38] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional neural networks for biomedical image segmentation," in *arXiv preprint: 1505.04597*. arXiv, 2015, pp. 1–8.
- [39] A. Chaurasia and E. Culurciello, "Linknet: Exploiting encoder representations for efficient semantic segmentation," in *arXiv preprint: 1707.03718*. arXiv, 2017, pp. 1–8.
- [40] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. Yuille, "Deepplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs," in *arXiv preprint: 1606.00915*. arXiv, 2017, pp. 1–8.
- [41] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2016, pp. 770–778.
- [42] M. Tan and Q. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," in *arXiv preprint: 1707.03718*. arXiv, 2017, pp. 1–8.
- [43] R. Ilija, R. P. Kosaraju, R. Girshick, K. He, and P. Dollar, "Designing network design spaces," in *arXiv preprint: 2003.13678*. arXiv, 2020, pp. 1–8.