machines

MDPI

Article

Pipeline Landmark Classification of Miniature Pipeline Robot π-II Based on Residual Network ResNet18

Jian Wang [1], Chuangeng Chen [1], Bingsheng Liu [1], Juezhe Wang [2] and Songtao Wang [3,*]

[1] School of Mechanical and Electrical Engineering, Guilin University of Electronic Technology, Guilin 541004, China; wangjianrt@guet.edu.cn (J.W.); 1955835175@mail.guet.edu.cn (C.C.); lbs527@mails.guet.edu.cn (B.L.)

[2] School of Information and Communication, Guilin University of Electronic Technology, Guilin 541004, China; wangjuezhe@guet.edu.cn

[3] School of Mechanical Engineering, Nanchang Institute of Technology, Nanchang 330099, China

* Correspondence: s.wang@nit.edu.cn

Abstract: A pipeline robot suitable for miniature pipeline detection, namely π-II, was proposed in this paper. It features six wheel-leg mobile mechanisms arranged in a staggered manner, with a monocular fisheye camera located at the center of the front end. The proposed robot can be used to capture images during detection in miniature pipes with an inner diameter of 120 mm. To efficiently identify the robot's status within the pipeline, such as navigating in straight pipes, curved pipes, or T-shaped pipes, it is necessary to recognize and classify these specific pipeline landmarks accurately. For this purpose, the residual network model ResNet18 was employed to learn from the images of various pipeline landmarks captured by the fisheye camera. A detailed analysis of image characteristics of some common pipeline landmarks was provided, and a dataset of approximately 908 images was created in this paper. After modifying the outputs of the network model, the ResNet18 was trained according to the proposed datasets, and the final test results indicate that this modified network has a high accuracy rate in classifying various pipeline landmarks, demonstrating a promising application prospect of image detection technology based on deep learning in miniature pipelines.

Keywords: miniature pipeline robot; pipeline landmarks; residual network ResNet18; hyper parameters

check for updates

Citation: Wang, J.; Chen, C.; Liu, B.; Wang, J.; Wang, S. Pipeline Landmark Classification of Miniature Pipeline Robot π-II Based on Residual Network ResNet18. *Machines* **2024**, *12*, 563. https://doi.org/10.3390/machines12080563

Academic Editors: Dan Zhang, Jozsef Sarosi, Peter Sarcevic and Akos Odry

Received: 9 June 2024
Revised: 9 August 2024
Accepted: 13 August 2024
Published: 16 August 2024

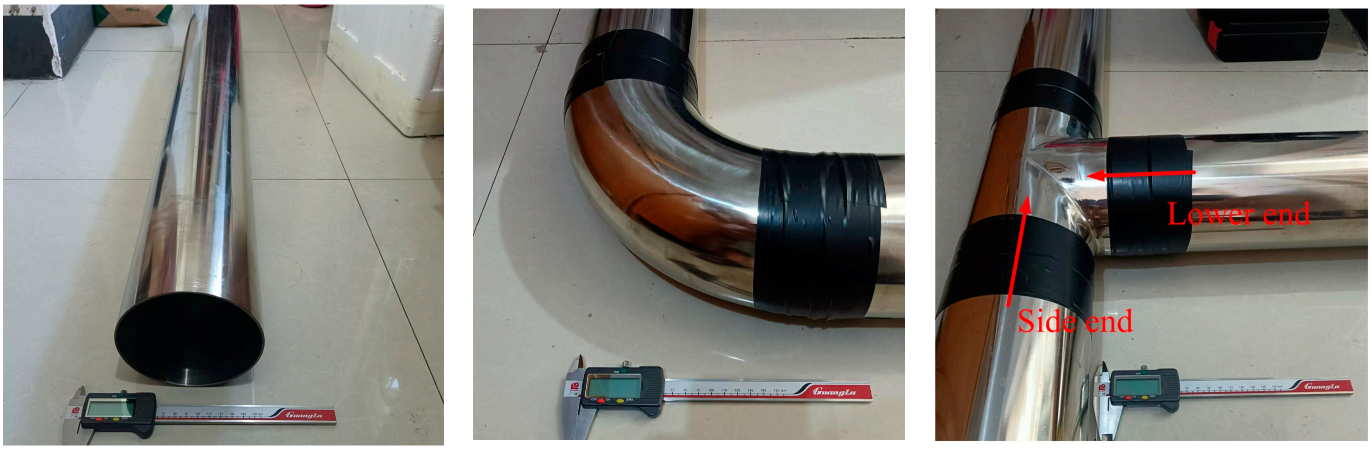
1. Introduction

Pipelines play a crucial role in the social and economic life of human beings. They can be used for the long-distance transportation of oil and natural gas, municipal water, and natural gas supply, as well as the conveyance of gases in the pharmaceutical and chemical industries. Over time, pipelines may experience aging, leaks, corrosion, and other issues that necessitate timely maintenance to prevent economic losses or safety hazards. Given that pipeline systems are typically buried underground or situated within large equipment, external inspection and maintenance are often impractical. Consequently, pipeline robots have emerged as a solution. By referring to existing, mature mobile mechanisms, scientists worldwide have developed various types of pipeline robots such as PIGs (Pipeline Inspection Gauges), wheeled-type pipeline robots, tracked-type pipeline robots, helical-type pipeline robots, snake-like pipeline robots, and inchworm-like pipeline robots. These robots are equipped with different sensors to facilitate inspection within the pipelines.

Wheeled and tracked pipeline robots are relatively mature in commercialization. For example, Jun [1] from the Harbin Institute of Technology in China developed a pipeline robot with six wheel legs arranged in a staggered configuration, suitable for pipes with a diameter of 195 mm. This robot uses a drag cable to supply power externally. Kim [2] from Sungkyunkwan University in South Korea proposed a multi-axis differential gear mechanism designed with a single motor as the power input and multiple sets of wheels as the output. This differential mechanism automatically distributes the driving force without any control force, making it highly efficient when traveling in straight and curved pipelines.

Kwon from Hanyang University in South Korea developed double-row wheeled [3] and triple-row wheeled [4] pipeline robots, which use a support mechanism (wall-pressing mechanism) with a large range of variation, suitable for pipelines with a diameter of 100 mm. The double-row wheel mechanism has one degree of freedom for steering, making it suitable for turning in T-shaped pipelines. Both robots also use drag cables for external power supply. Considering that the application in this paper is for a miniature pipeline system, a six-wheel-leg form of the mobile mechanism was proposed to be applied to miniature pipelines with an inner diameter of 120 mm.

In addition to straight pipes (as shown in Figure 1a), pipeline systems also include some transition structures, such as curved pipes (as shown in Figure 1b) and T-shaped pipes (as shown in Figure 1c), referred to as landmarks in the pipeline [5,6]. Generally, pipeline systems are composed of a limited variety of these landmark segments. Therefore, for a pipeline robot to effectively perceive its internal environment or its own state within the pipeline system, it must first efficiently identify and classify these landmarks using the sensors carried by the robot. For example, if the identified landmark is a straight pipe, the pipeline robot needs to move straight; if the identified landmark is a curved pipe, the robot needs to turn; if the identified landmark is a T-shaped pipe, i.e., a three-way structure, the robot needs to further determine whether it is entering from the side end or the lower end of the T-shaped pipe. Efficient identification and classification of pipeline landmarks are crucial for the autonomous navigation control of the pipeline robot within the pipeline and play a significant role in constructing the topological structure or mapping the entire pipeline system.



(**a**) (**b**) (**c**)

Figure 1. Common structures in the pipeline system (with an inner diameter of 120 mm): (**a**) straight pipe; (**b**) curved pipe; (**c**) T-shaped pipe.

Currently, the classification of pipeline landmarks, based on the principles of detection equipment, can be divided into two main categories: shadow-based and reflection-based methods.

Shadow-Based Method: Zhang [7] used a pipeline robot to detect landmarks of curved pipes with an inner diameter of 200 mm and a turning radius of 300 mm. The robot employs a monocular CCD camera as a visual sensor, utilizing the captured images to calculate movement invariant quadrature to identify the presence of bends, as these parameters are independent of lighting changes and vary with the turning of the reflective surfaces of the curved pipe. Ahrary [8] equipped the KANTARO sewer pipeline robot (suitable for 200–300 mm pipelines) with binocular vision devices to identify and classify the shadows of manhole (T-shaped pipe) and joint (welding connection) landmarks. They proposed a computationally efficient matching method, namely Linear Computation, to classify these landmarks and create 3D models. Lee [9] researched the pipeline robot MRINSPECT V to detect 8-inch (203 mm) natural gas pipelines. He designed a spotlight source with 64 high-brightness LEDs. The combination of spotlight and camera angles creates distinct shadow patterns on the pipeline landmarks (curved pipes and T-shaped pipes), which are then matched to determine the type of landmark. The study also discusses the differences in shadow patterns when detecting T-shaped pipes from the side end and lower end. Thielemann [10] used a time-of-flight camera (TOF Mesa SwissRanger SR-3000) to detect 400 mm sewer pipelines and scanned the 3D models of Y-shaped and T-shaped pipe junctions, classifying these landmarks.

Reflection-Based Method: Chand [11] used a pipeline robot equipped with 3D MEMs Lidar (Infinisoleil FX10 Lidar) to detect and map 4-inch (102 mm) firefighting pipelines, scanning curved pipe landmarks and creating 3D point cloud models. He proposed an ANV (average normal vector) method based on these models to determine the direction of the curved pipe. Choi [12] used the pipeline robot MRINSPECT VI to detect straight, curved, and T-shaped landmarks in 8-inch (203 mm) pipelines. He designed a detection device with nine PSDs (Position Sensitive Devices), with one at the front and eight evenly distributed around the sides, to measure distances to the pipeline walls in nine directions. The varying combinations of reflection distances are used to identify the corresponding landmarks.

These two methods can also be combined into a hybrid method: Lee [13–15] utilized pipeline robot MRINSPECT V with a combination of spotlight cameras and line lasers to scan straight, curved, and T-shaped pipes. The reflected images of lasers are captured by the camera, and the different patterns are used to identify various pipeline landmarks. Choi [16] used the pipeline robot MRINSPECT VI to detect 8-inch (203 mm) pipelines, combining the spotlight and camera system from [9] and the PSD system from [12]. The camera is used for distant detection, and the PSD system is used for close detection, identifying straight, curved, and T-shaped pipe landmarks.

Through the analysis of the aforementioned literature, it is evident that there are many effective methods for identifying and classifying various pipeline landmarks. Whether based on the shadows or reflections of the landmarks, cameras are used for reception. The classification of landmarks requires the development of corresponding algorithms based on different graphical characteristics. For miniature pipelines, such as those with diameters below 100 mm, binocular cameras, medium to large LIDARs, and complex sensor-based detection equipment are not very suitable due to size limitations. Using devices like PSDs requires close proximity to the landmarks to be effective, necessitating preliminary judgment by the camera system, which makes the mechanism bulky. Therefore, this paper proposes using a fisheye camera as the pipeline landmark detection device to identify and classify landmarks in stainless steel pipelines with an inner diameter of 120 mm.

Traditional methods for classifying pipeline landmark images primarily rely on the different shadow characteristics reflected by various pipeline landmarks to summarize corresponding patterns [6,9] (as shown in Figure 2). In a straight pipe, the near end close to the robot camera will have a very strong reflection, which becomes weaker with increasing depth of the pipeline, appearing as non-reflective and dark in the distance, forming a circular shadow in the center. At the bend of a curved pipe, light does not reflect back, creating a crescent-shaped shadow at the edge. For T-shaped pipes, there are two modes: entering from the side end or from the lower end. When entering from the side, it resembles a combination of a straight pipe and a curved pipe, resulting in a shadow pattern with a circular shape in the center and a crescent shape at the edge. When entering from the lower end, it is akin to two curved pipes joined together, resulting in a shadow pattern with two crescent shapes on opposite edges.

After establishing the shadow patterns for each pipeline landmark, the captured real landmark images can be compared to the standard landmark patterns to identify the type of real landmark. This is the standard procedure for traditional image matching. However, in actual classification, due to the influence of lighting and the complexity of the internal pipeline environment, the captured landmark shadow images can vary greatly and differ significantly from the ideal patterns. Various actual cases will be provided later to illustrate this situation. Therefore, traditional visual processing methods have certain limitations.

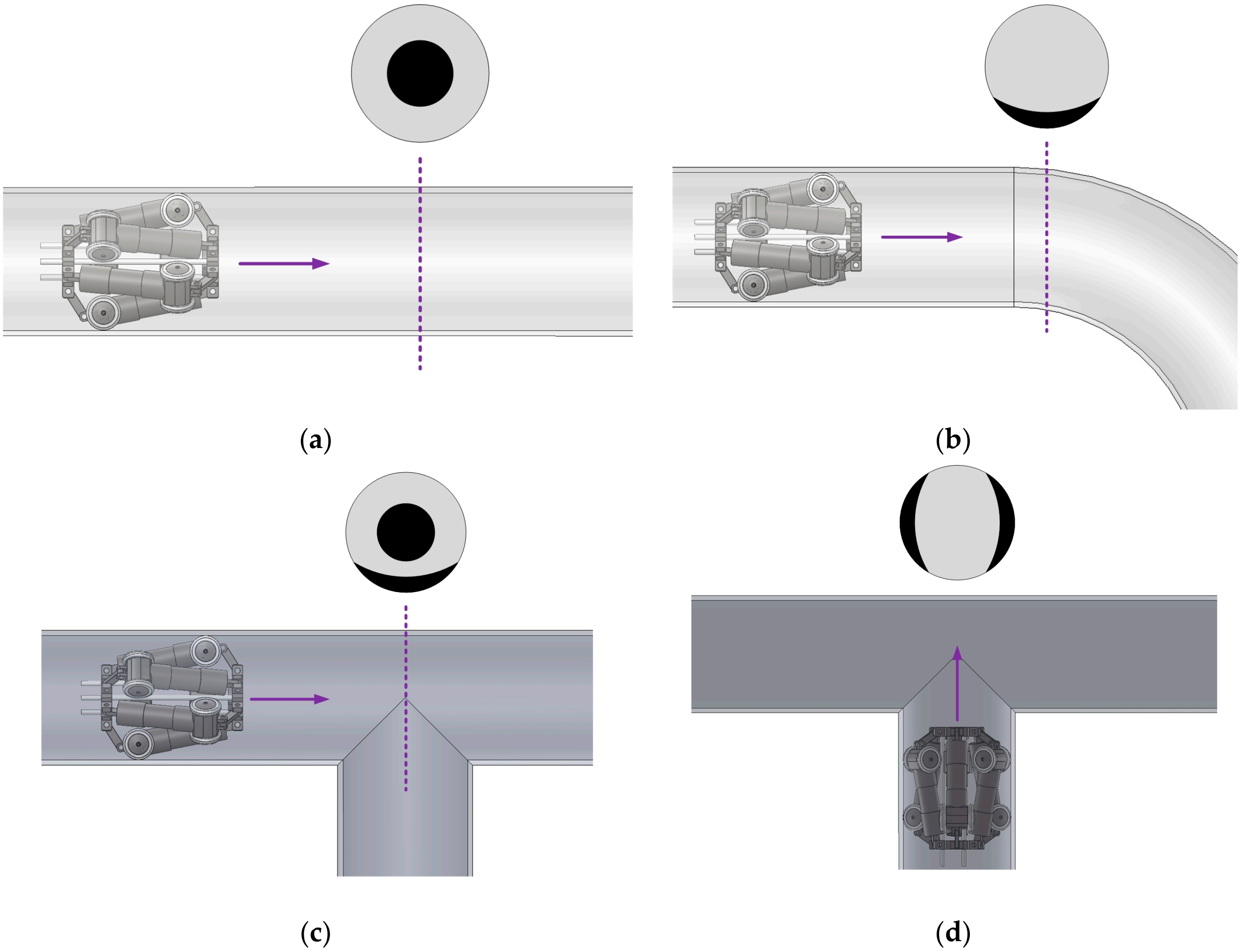


Figure 2. Pipeline landmark shadow patterns (stemmed from [5]): (**a**) straight pipe landmark; (**b**) curved pipe landmark; (**c**) T-shaped pipe-side entry landmark; (**d**) T-shaped pipe bottom entry landmark.

The existing literature has already applied deep learning methods based on CNNs (convolutional neural networks) to pipeline detection-related applications. For example, Haurum [17] constructed a dataset for sewer pipeline water level estimation using 11,558 CCTV sewer images provided by three Danish utility companies. Four types of deep neural network models—AlexNet, ResNet-18, ResNet-34, and ResNet-50—were trained for water level height classification, and the performance of each neural network model is evaluated using the F1 score. Patil [18] created a sewer blockage imagery classification dataset for 200 mm diameter PVC sewers. He used image enhancement techniques such as greyscaling, adding salt and pepper noise, random exposure adjustments, cutout, and mosaic, expanding the original dataset from 7040 images to 14,765 images. The YOLOX model was then trained to identify and classify three major sewer blockages: grease, plastic, and roots, achieving performance metrics above 90%. Zhao [19] used an improved YOLOv5 model to identify leakage points with a minimum diameter of 0.9 mm in DN100PE natural gas pipelines. He expanded the original dataset of 171 images to 1000 images using a DCGAN (Deep Convolutional Generative Adversarial Network), and the trained model achieved performance metrics above 95%.

From the above analysis, it is clear that if deep learning is to be used for the identification and classification of various pipeline landmarks, a dataset must first be established. The most significant contribution of this paper is the creation of a pipeline landmark dataset for deep learning training and testing. The pipeline robot π-type proposed in this paper is equipped with a fisheye camera to capture images of four types of landmarks: straight pipes, curved pipes, T-shaped pipes entering from the side end, and T-shaped pipes entering from the lower end. So, a residual network, i.e., ResNet18, for the classification of these pipeline landmarks can be used.

The arrangement of the paper is as follows: Section 2 mainly introduces the experimental platform of the proposed pipeline robot and its main detection modules; Section 3 covers the image processing of various pipeline landmarks and the establishment of the dataset; Section 4 discusses the training methods of ResNet18 model, loss functions, the training and testing process, and the presentation of metrics of the experimental results; Section 5 focuses on the influence of hyperparameter "batch_size"; finally, Section 6 draws a conclusion of the paper and gives a discussion on future research directions.

2. Experimental Platform and Its Detection System of Miniature Pipeline Robot π-II

After referencing existing mature mobile mechanisms, a pipeline robot suitable for an inner diameter of 120 mm, π-II type pipeline robot [20], was developed, as shown in Figure 3.

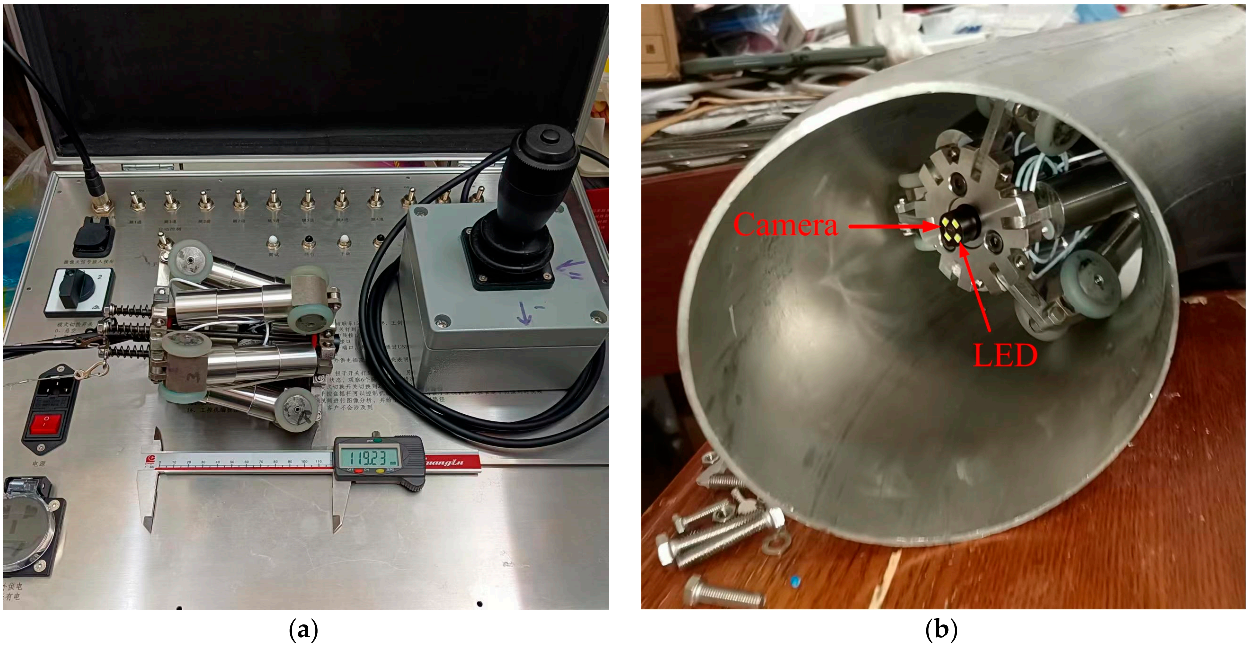


Figure 3. Miniature pipeline robot π-II experimental platform: (**a**) pipeline robot prototype and its control box; (**b**) pipeline robot equipped with a fisheye camera and LED light source.

To inspect the inner walls of the pipes, π-II type pipeline robot was equipped with a vision detection system, as shown in Figure 3b. The center of the mechanism was equipped with a monocular fisheye camera, which includes four LED patches inside the camera device, providing three levels of brightness to form the spotlight source. The main parameters of the fisheye camera are shown in Table 1. Due to size limitations, the selected camera had only 300,000 pixels. Using a higher resolution would increase the camera device's diameter. Additionally, the spotlight was not as meticulously designed as the 64 LED patches in [9], which require more space and a thicker power cable. In practical applications, there are no other light sources inside the pipeline system, so the brightness of the light source of the proposed π-II type pipeline robot is relatively weak. Consequently, the shadows cast on the pipeline landmarks will be complex and significantly different from the ideal conditions (as shown in Figure 2). These issues arise due to the size constraints of the pipeline.

Table 1. Performance parameters of the fisheye camera equipped on π-II pipeline robot.

Parameter	Value
Resolution	640 × 480
Frame Rate	30 fps
View Angle	70°
Waterproof Level	IP67
Photo Format	JPEG

Figure 4 shows the technical roadmap of this project. It consists mainly of two parts: creating a dataset of pipe landmarks that constitutes the majority of the work and training and testing the model based on the created dataset. When creating the dataset, photos can be saved into corresponding folders. For example, straight pipes can be classified as "0", so photos of straight pipes can be saved in a folder named "0". After each epoch of model training, immediate testing is conducted, which allows the performance of the model to be observed as it evolves over the epochs.

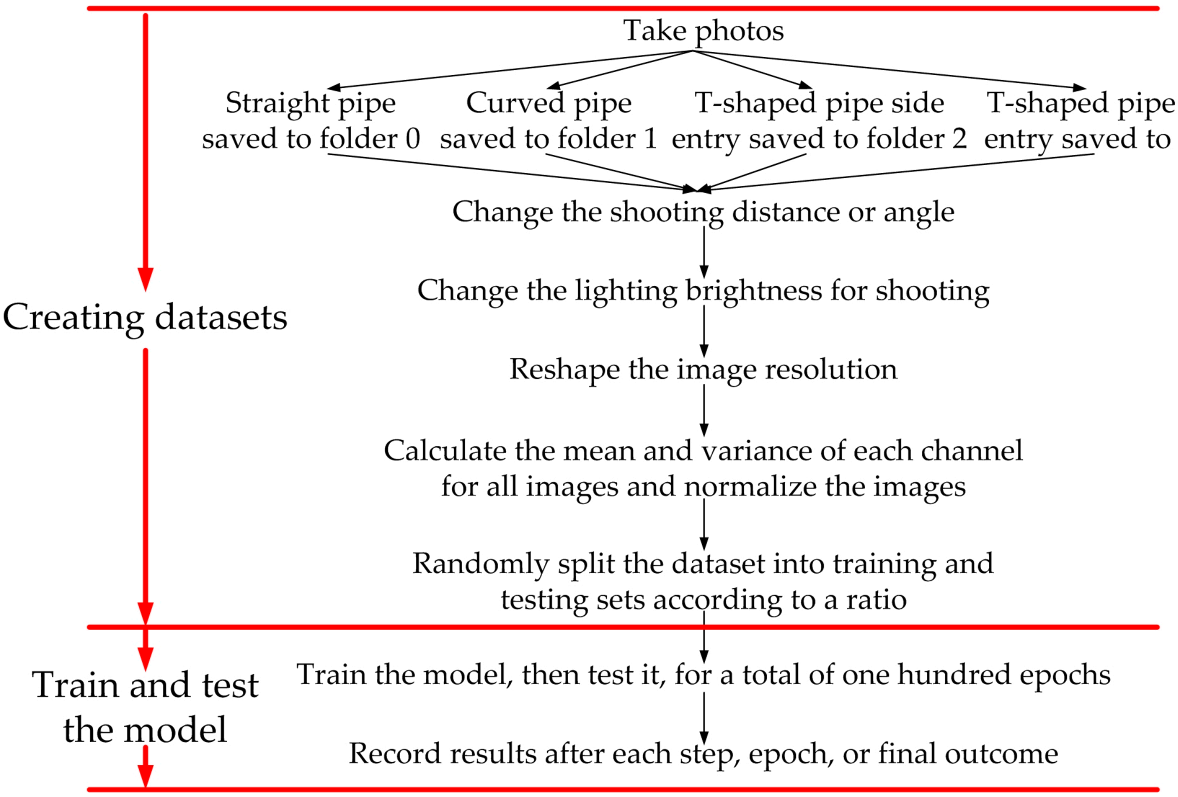

Take photos
Straight pipe saved to folder 0
Curved pipe saved to folder 1
T-shaped pipe side entry saved to folder 2
T-shaped pipe bottom entry saved to folder 3
Change the shooting distance or angle
Change the lighting brightness for shooting
Reshape the image resolution
Calculate the mean and variance of each channel for all images and normalize the images
Randomly split the dataset into training and testing sets according to a ratio
Train the model, then test it, for a total of one hundred epochs
Record results after each step, epoch, or final outcome
Creating datasets
Train and test the model

Figure 4. The technical roadmap of this project.

3. Construction of Typical Pipeline Landmark Dataset

There are currently no publicly available pipeline landmark datasets. Therefore, using the fisheye camera detection system of π-II type pipeline robot developed in this paper, images of various landmarks are captured to construct a usable dataset. The main landmarks consist of four categories: images taken by the robot in straight pipes, images taken in curved pipes, images taken at the side end of T-shaped pipes, and images taken at the lower end of T-shaped pipes.

3.1. Shadow Analysis of Typical Pipeline Landmarks

The fisheye camera equipped on pipeline robot π-II exhibits barrel distortion, a type of radial distortion, as shown in Figure 5. In this paper, in order to reduce computational load this distortion was not corrected. The final experimental results also indicate that this does not affect the classification performance of the deep learning network. The original resolution of the images captured by the fisheye camera was 640 × 480. Since the internal structure of the pipeline was generally symmetrical, only the central 480 × 480 portion of the original image was selected through programming.

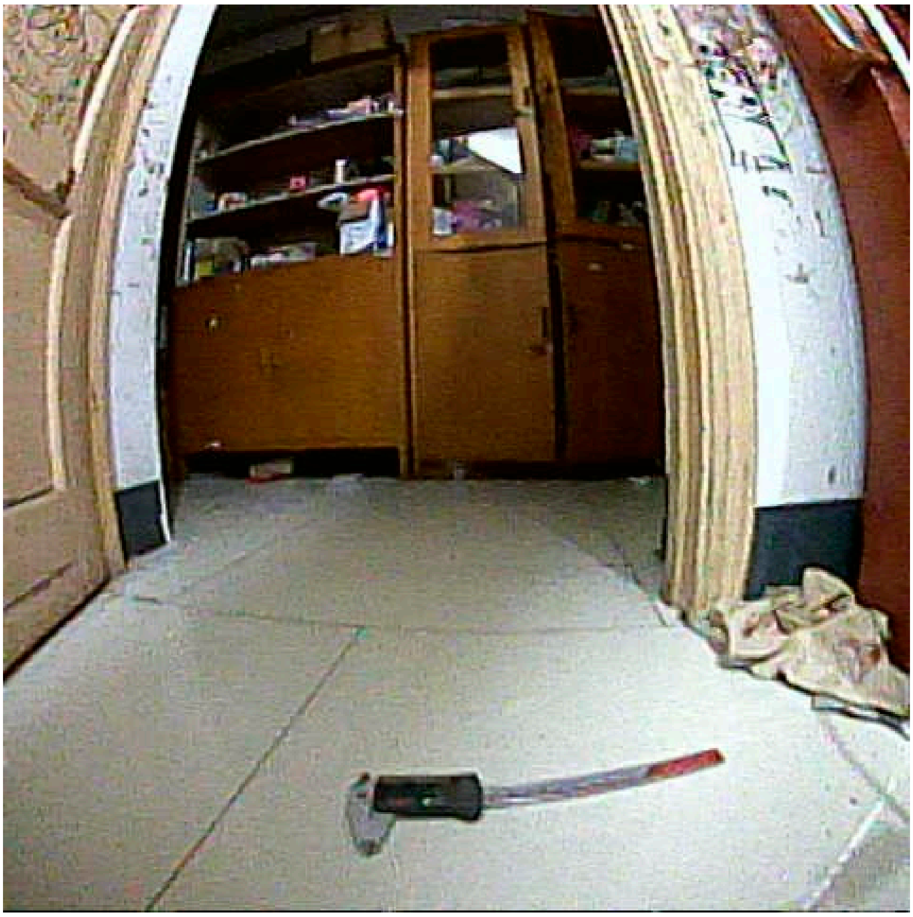


Figure 5. Distorted images captured by the fisheye camera on π-II pipeline robot.

3.1.1. Shadow Characteristics of Straight Pipes

In straight pipelines (as shown in Figure 1a), the image captured by the fisheye camera is relatively symmetrical. Ideally (Figure 2a), the spotlight causes reflections on the inner wall of the pipeline, with the brightness of the reflections decreasing as the depth of the pipeline increases. This creates a circular shadow at the center. In practice, this brightness decrease is gradual, without a clear shadow boundary (as shown in Figure 6a, taken inside a straight pipe of 4 m long). Additionally, if there is a pipeline connection structure not far ahead of the robot, reflections may occur at the center, resulting in an irregular shadow instead of a perfect circle (as shown in Figure 6b, with a connection structure 2 m ahead in the straight pipe). This makes traditional pattern matching ineffective for straight pipes with structures 2 m ahead. Furthermore, if the fisheye camera is angled slightly relative to the pipe axis, the brightness of the reflections on the inner wall at the same depth can vary greatly, as shown in Figure 6c, with large shadows present in the upper left and center. This is why deep learning is used instead of traditional machine vision methods in this paper. Additionally, the fisheye camera's LED light source provides three brightness levels. To increase the diversity of the dataset, shadow images of straight pipe landmarks at different brightness levels also need to be collected (as shown in Figure 6d,e).

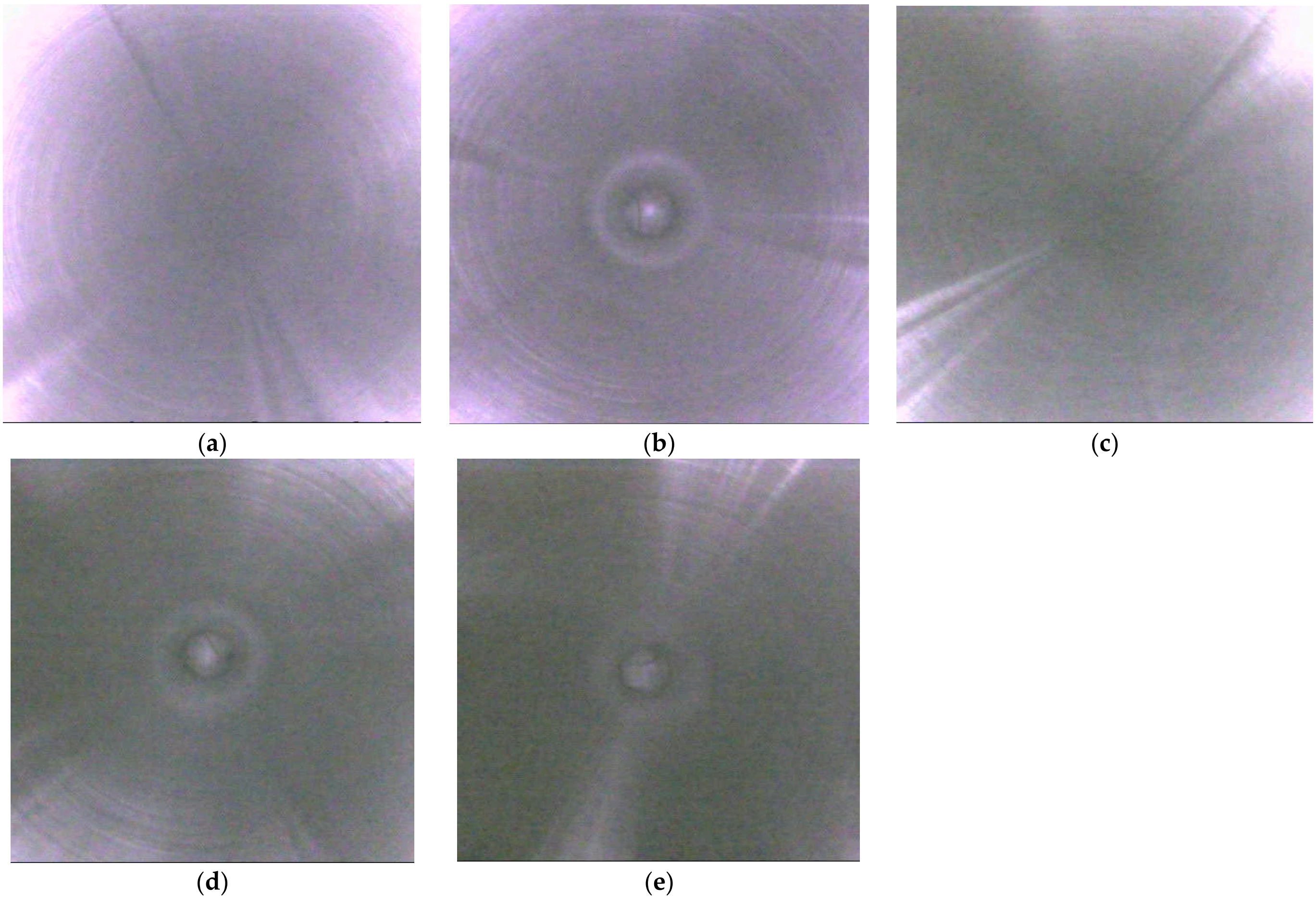


Figure 6. Shadow characteristics of straight pipe landmarks: (**a**) A 4m straight pipe with no structure nearby; (**b**) a straight pipe with a structure present2 m ahead; (**c**) uneven reflections caused by improper camera angle; (**d**) brightness of LED light system reduced by one level; (**e**) brightness of LED light system reduced by two levels.

The straight pipe landmark was labeled as class 0. A total of 224 photos were collected using the fisheye camera. To avoid conflicts with other landmarks, only the interior of straight pipes with structures beyond 2 m were photographed. If other pipeline landmarks were present within 2 m in a straight pipe, a specific determination was required, and it was no longer considered a straight pipe landmark.

3.1.2. Shadow Characteristics of Curved Pipes

To capture images of curved pipe landmarks, it is necessary to consider shadow patterns under three lighting intensities and three distance conditions, which is similar to straight pipe landmarks. Additionally, unlike straight pipe images, it is essential to account for the shooting angle of the fisheye camera. The shadow patterns in curved pipes are not symmetrical, and the brightness of reflections on the inner wall decreases along the direction of the bend, presenting a curved shape rather than the idealized crescent-shaped shadow shown in Figure 2b. Figure 7a,b show shadow patterns at different angles rotating around the axis of the pipe under brighter light sources for closer curved pipe landmarks. To enhance the diversity of the dataset, it is beneficial to vary the rotating angles as much as possible. Figure 7c,d depict images captured under lower brightness levels at closer distances, while Figure 7e shows an image taken at an intermediate distance, and Figure 7f was taken at the farthest distance.

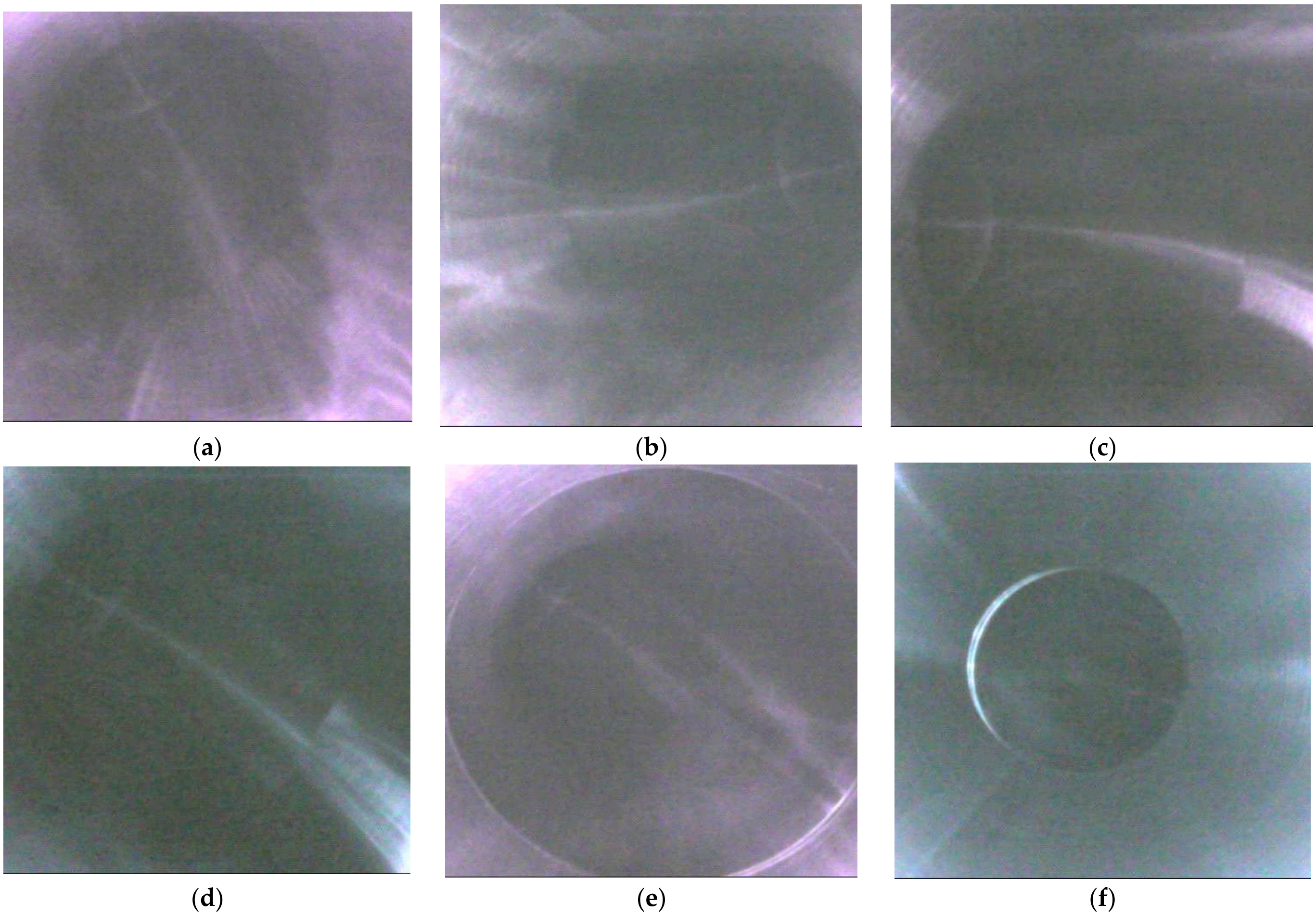


Figure 7. Images taken with a fisheye camera inside a curved pipe: (**a**) The near opening facing the upper left; (**b**) the near opening facing the right; (**c**) the near opening with brightness reduced by one level; (**d**) the near opening with brightness reduced by two levels; (**e**) taken at the middle distance; (**f**) taken at the farthest distance.

The curved pipe landmark was labeled as class 1. A total of 219 photos were collected using the fisheye camera. These photos were taken within 2 m toward the bent from different rotating angles along the axis of the pipe, under three different levels of brightness, and at various distances. The number of photos taken under different conditions was carefully considered to ensure the diversity and balance of the dataset.

3.1.3. Shadow Characteristics of T-Shaped Pipe Entering from the Side End

To capture images of T-shaped pipe side entry landmarks, it was necessary to consider shadow patterns under three different levels of brightness and three distance conditions. The captured shadow patterns were not symmetrical. At the lower end of the targeted T-shaped pipe, a crescent-shaped shadow appeared, while at the far end, in the direction of the light source, it resembled the shadow of a straight pipe. This closely matches the idealized shadow of the T-shaped pipe side entry landmark shown in Figure 2c. Figure 8a,b show shadow patterns at different rotating angles along the axis of the pipe under brighter light sources for closer T-shaped pipe side entry landmarks. To enhance the diversity of the dataset, varying the rotating angles as much as possible was beneficial. Figure 8c,d depict images captured under lower brightness levels at closer distances, while Figure 8e shows an image taken at an intermediate distance, and Figure 8f was taken at the farthest distance.

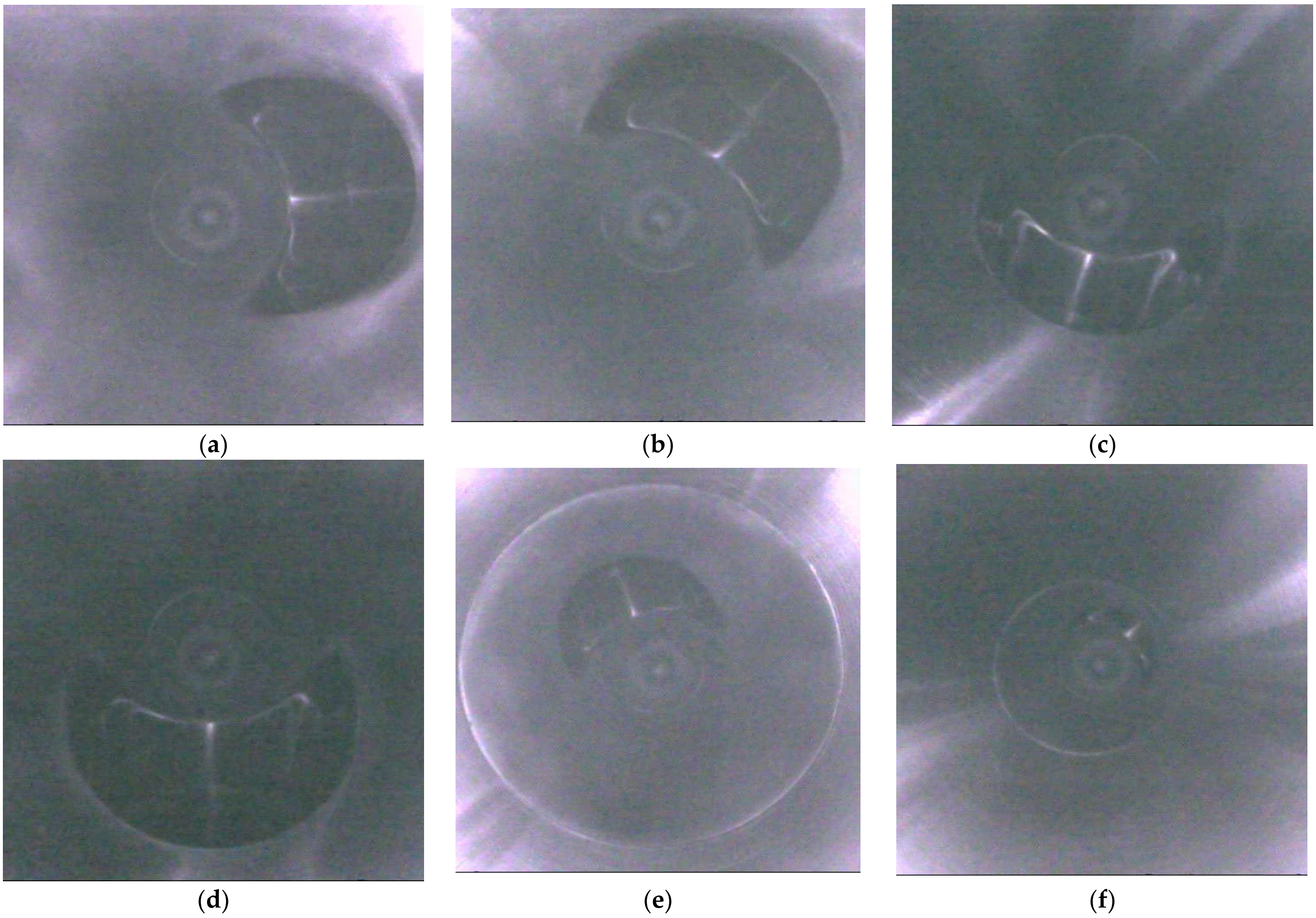
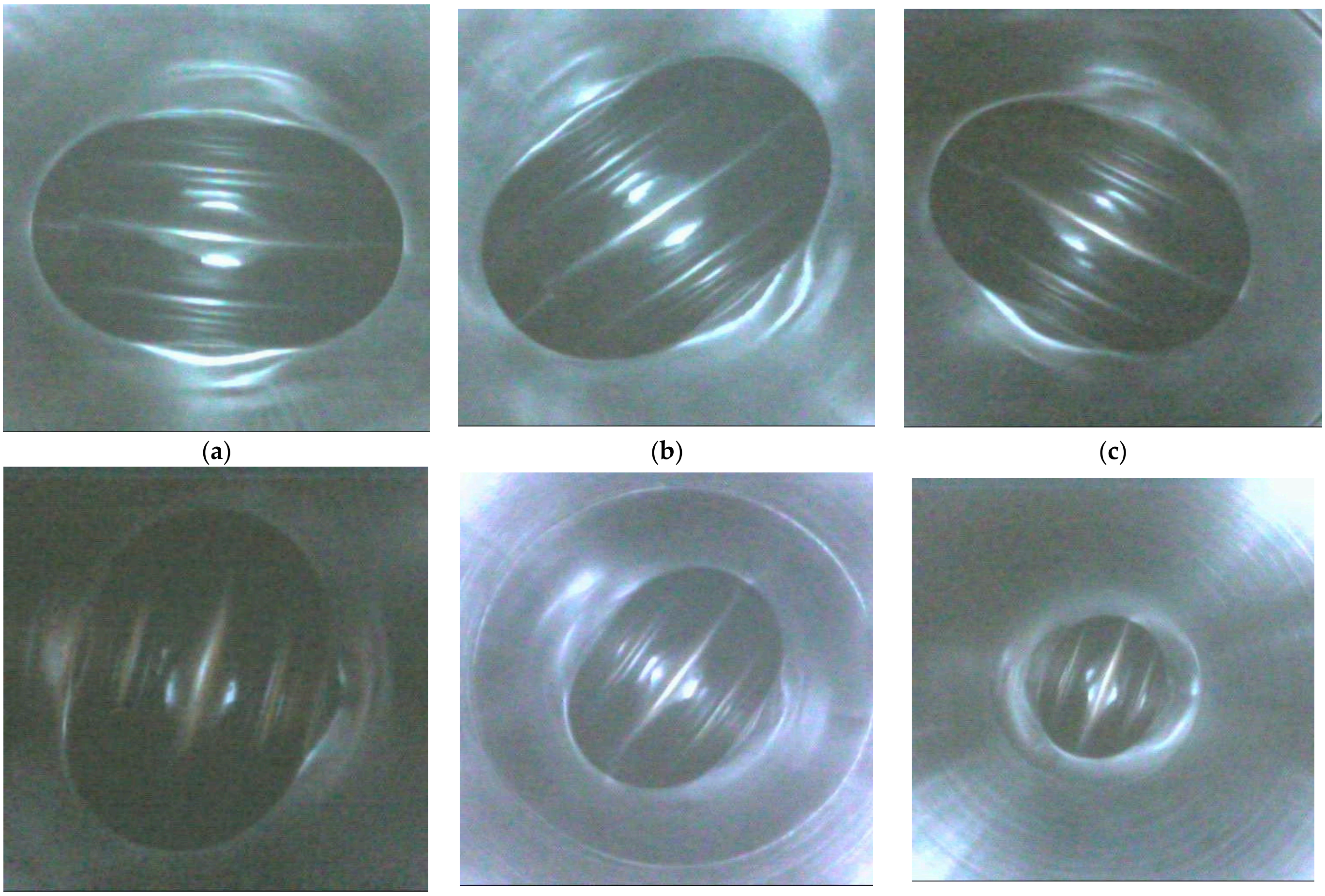


Figure 8. Images taken with a fisheye camera at the side end of a T-shaped pipe: (**a**) The near opening facing the right; (**b**) the near opening facing the upper right; (**c**) the near opening with brightness reduced by one level; (**d**) the near opening with brightness reduced by two levels; (**e**) taken at the middle distance; (**f**) taken at the farthest distance.

The T-shaped pipe side entry landmark was labeled as class 2. A total of 227 photos were collected using the fisheye camera. These photos were taken within 2 m toward the side end of a T-shaped pipe, at different rotating angles, with three different light source brightness levels, and at various distances. The number of photos taken under different conditions was carefully considered to ensure the diversity and balance of the dataset.

3.1.4. Shadow Characteristics of T-Shaped Pipe Entering from the Lower End

To capture images of T-shaped pipe bottom entry landmarks, it was necessary to consider shadow patterns under three different brightness levels and three distance conditions. The captured shadow patterns were not symmetrical, with the two side ends of the T-shaped pipe presenting two opposing, semi-circular shadows. This is similar to, but not entirely consistent with, the idealized shadow of the T-shaped pipe bottom entry landmark shown in Figure 2d. Figure 9a,b show shadow patterns at different rotating angles along the axis of the pipe under brighter light sources for closer T-shaped pipe bottom entry landmarks. To enhance the diversity of the dataset, varying the rotating angles as much as possible was beneficial. Figure 9c,d depict images captured under lower brightness levels at closer distances, while Figure 9e shows an image taken at an intermediate distance, and Figure 9f was taken at the farthest distance.



(**a**) (**b**) (**c**)

(**d**) (**e**) (**f**)

Figure 9. Images taken with a fisheye camera at the bottom end of a T-shaped pipe: (**a**) The near opening facing the left and right; (**b**) the near opening facing the bottom left and upper right; (**c**) the near opening with brightness reduced by one level; (**d**) the near opening with brightness reduced by two levels; (**e**) taken at the middle distance; (**f**) taken at the farthest distance.

The T-shaped pipe bottom entry landmark was labeled as class 3. A total of 238 photos were collected using the fisheye camera. These photos were taken within 2 m toward the bottom end of a T-shaped pipe, at different rotating angles, with three different light source brightness levels, and at various distances. The number of photos taken under different conditions was carefully considered to ensure the diversity and balance of the dataset.

3.2. Creation of the Pipeline Landmark Dataset

In this experiment, a total of 908 shadow images of four types of pipeline landmarks were collected. The original photos were BGR three-channel with a resolution of 480 × 480. Before importing each photo into the dataset, preprocessing was required, which involved first resetting the resolution and then normalizing the pixel values of each channel. The Python code (version 3.8.13) was as follows:

```
self.pip_trainsform=torchvision.transforms.Compose([
torchvision.transforms.ToTensor(),
torchvision.transforms.Resize((height, width)),
torchvision.transforms.Normalize(mean=[B_mean, G_mean, R_mean],
std=[B_std, G_std, R_std]),
])
```

In this context, "torchvision.transforms.Compose" is a composite function for various image preprocessing operations. "torchvision.transforms.Resize((height, width))" is used to reset the resolution of an image to the specified height and width. In this paper, height=200 and width=200. "torchvision.transforms.Normalize(mean=[B_mean, G_mean, R_mean],std=[B_std, G_std, R_std])" is used to normalize the pixel values of each channel of the image, defined as follows:

$$\text{Pixel\_Channel\_X\_Normalized} = \frac{\text{Pixel\_Channel\_X\_Origin} - \text{X\_mean}}{\text{X\_std}}, \tag{1}$$

where X can be B, G, or R, representing any channel. "mean=[B_mean, G_mean, R_mean]" are the mean pixel values for each channel across all images in the dataset, and "std=[B_std, G_std, R_std]" are the standard deviations. These values differ for the training and test datasets and must be calculated in advance. Figure 10 shows a comparison between the original images and the preprocessed images. The upper images are the original, and the lower part images are the preprocessed ones. Due to space limitations, the original images and the corresponding preprocessed images are scaled to the same size, with the upper right corner indicating the pipeline landmark categories: 0—straight pipe landmark, 1—curved pipe landmark, 2—T-shaped pipe side entry landmark, 3—T-shaped pipebottom entry landmark.

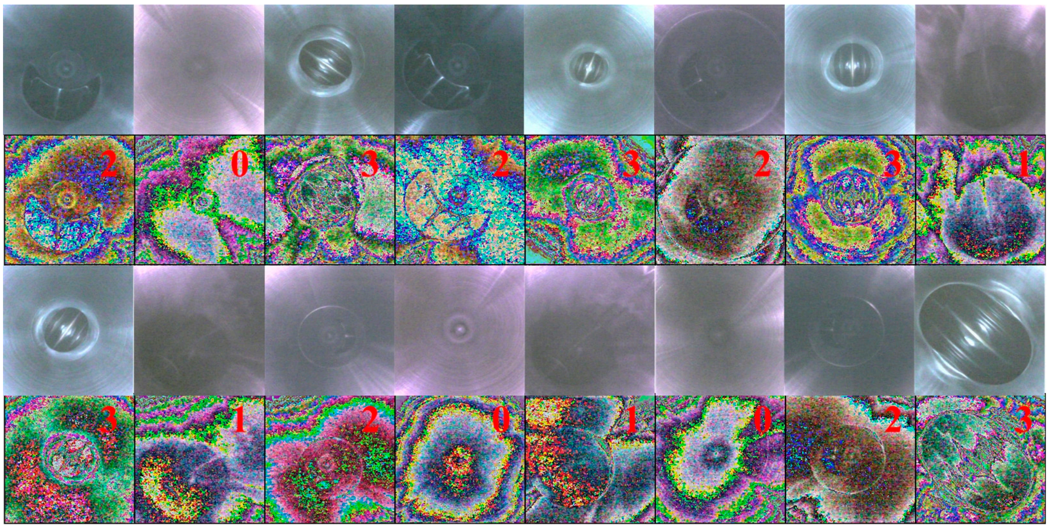


Figure 10. A comparison between original images and preprocessed ones.

After preprocessing the captured images, divide the dataset into training and test sets in an 8:2 ratio, ensuring that each pipeline landmark category is represented in both sets.

```
train_images,test_images,train_labels,test_labels=sklearn.model_selecti
on.train_test_split(images,labels,test_size=0.2,random_state=42)
```

In this context, "images" refers to the entire dataset of images, and "labels" refers to the corresponding pipeline landmark categories for all images. "train_images" and "train_labels" consist of 80% of the images randomly selected to form the training dataset and their corresponding landmark categories. "test_images" and "test_labels" consist of the remaining 20% of the images forming the test dataset and their corresponding landmark categories.

4. Pipeline Landmark Classification Based on the Deep Learning Mode ResNet18

Considering the need for a certain level of real-time performance for pipeline robots operating within pipelines, the deep learning model used to identify various pipeline landmarks must balance recognition performance and parameter size to avoid excessive runtime. The residual network model ResNet18, although relatively basic, offers strong classification performance. This paper primarily investigates the feasibility of using the

ResNet18 network model to classify four types of pipeline landmarks. The following context will detail the model's characteristics, the selected loss function and optimization method, performance metrics, as well as the training process and test results.

4.1. Choice and Modification of the Classification Model

ResNet18 is a typical residual network model, meaning it contains modules (as shown in Figure 11) that connect both the original signal and the residual signal. This design helps mitigate the degradation problem in deeper layers of abstraction. These smaller modules often stack to form a larger network. ResNet50 follows the same residual network approach but with a deeper stacking of modules, resulting in a larger network model with more parameters to train. Although ResNet50 can provide a more abstract understanding, it requires more computational resources. Additionally, for the pipeline landmark classification task in this paper, the dataset consisted of fewer than 1000 images, making ResNet50 more prone to overfit. Therefore, this paper prioritized the use of the ResNet18 network model. The computer hardware resources used for training the model are listed in Table 2.

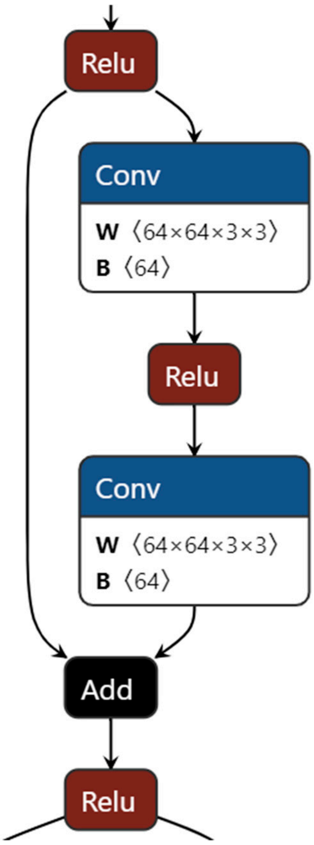


Figure 11. Residual network module of ResNet18.

Table 2. Computer hardware resources.

Device	Specification
CPU	Intel i5-7300HQ 2.50GHz
Memory	20 GB
GPU	NVIDIA GeForce GTX 1050
GPU Memory	4 GB

Secondly, for small sample size datasets, if all parameters of the network model ResNet18 are retrained, the final test performance metrics of the network may not be very good. Therefore, the original ResNet18 model can be modified, as shown in the following custom residual network class code (python version 3.8.13):

```
class PIPEResNet18(torch.nn.Module):
    def __init__(self):
        super(PIPEResNet18,self).__init__()
self.cnn_layers=torchvision.models.resnet18(pretrained=True)
num_ftrs=self.cnn_layers.fc.in_features
self.cnn_layers.fc=torch.nn.Linear(num_ftrs,4)
    def forward(self,x):
        out=self.cnn_layers(x)
        return out
```

In this context, "cnn_layers" is defined as the standard ResNet18 network, "pretrained=True" indicates that the ResNet18 parameters can be updated, "num_ftrs=self.cnn_layers.fc.in_features" means obtaining the final layer of ResNet18, the "fc" network layer, and "self.cnn_layers.fc=torch.nn.Linear(num_ftrs,4)" redefines the "fc" layer, changing the original ResNet18 network "fc" layer from 1000 outputs (classifying 1000 categories) to four outputs (classifying four pipeline landmark categories).

4.2. Selection of the Loss Function and Optimization Method during Training

The loss function is typically used as the objective function during the model training phase. It aims to iteratively optimize the neural network model parameters to minimize the objective function. Classification models use the standard categorical cross-entropy loss for evaluation. For multi-class classification problems, the following formula can be used:

$$Loss = -\frac{1}{N}\sum_{i=1}^{N}\sum_{c=1}^{M} y_{ic}\log(p_{ic}), \tag{2}$$

where N is the number of samples. If considering the loss for one step in training, the number of samples N is the value of "batch_size" preset in the training dataset. If considering the loss for one epoch, the number of samples N is the number of images in the training dataset. M is the total number of pipeline landmarks. In this paper, there are four types of pipeline landmarks, i.e., $M = 4$. If the class of the i-th sample is c, then $y_{ic} = 1$; if the class of the i-th sample is not c, then $y_{ic} = 0$. p_{ic} is the predicted probability that the i-th sample belongs to class c.

The Python code for the loss function is as follows:

```
criterion=torch.nn.CrossEntropyLoss(reduction='mean')
```

Moreover, for training the samples, the Adam optimization method can be used with a learning rate set to 0.001. The Python code is as follows:

```
optimizer=torch.optim.Adam(model.parameters(),lr=0.001)
```

4.3. Selection of Performance Metrics during Testing

To evaluate a trained model, some performance metrics need to be selected. Common metrics for classification problems include accuracy, precision, recall, and F1 score. For multi-class classification problems, micro-accuracy can be used. Its formula is the following:

$$\text{micro\_accuracy} = \frac{\sum_{c=1}^{M}\text{TP}_c}{N} \tag{3}$$

where N is the number of samples, and TP_c is the number of samples predicted to be class c, those are actually class c. If considering the performance metrics for each step in testing, the number of samples N is the "batch_size" preset in the test dataset. If considering the performance metrics for one epoch in testing, the number of samples N is the number of images in the test dataset. M is the total number of classes.

The Python code corresponding to the performance metrics function is as follows:

```
micro_accracy=precision_score(test_labels,test_outputs,average='micro')
```

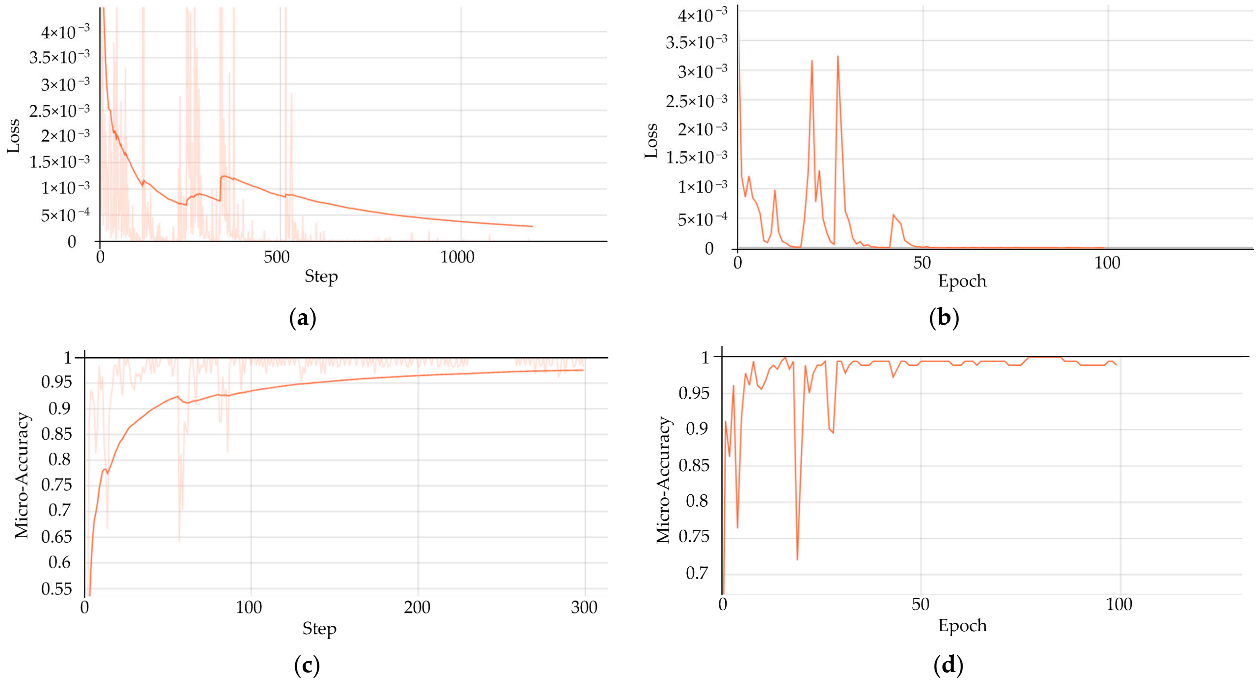
In this context, "test_labels" represents the category labels corresponding to the test samples, and "test_outputs" represents the predicted categories given by the model after inputting the sample images.

4.4. Training Process and Testing Results

The training process relies on the training dataset, while the testing process relies on the testing dataset. As shown in Table 3, the residual network model was trained for 100 epochs, where the hyperparameter "batch_size" was set to 64, and the value of the loss function was recorded at each step (as shown in Figure 12a), as well as at each epoch (as shown in Figure 12b), representing the micro and macro variations in the training process, respectively. After each epoch of the training process, the model was then used for testing, and the performance metric micro-accuracy was recorded at each step of the testing process (as shown in Figure 12c), as well as at each epoch (as shown in Figure 12d).

Table 3. Parameters of training process and test process.

Parameter	Value	
	Training	Test
Datasets	704	182
Resolution	200 × 200	200 × 200
Epochs	100	100
Batch size	64	64
Steps per epoch	11	3
Learning rate	0.001	---
Optimizer	Adam	---
Fps	142	100



(**a**)

(**b**)

(**c**)

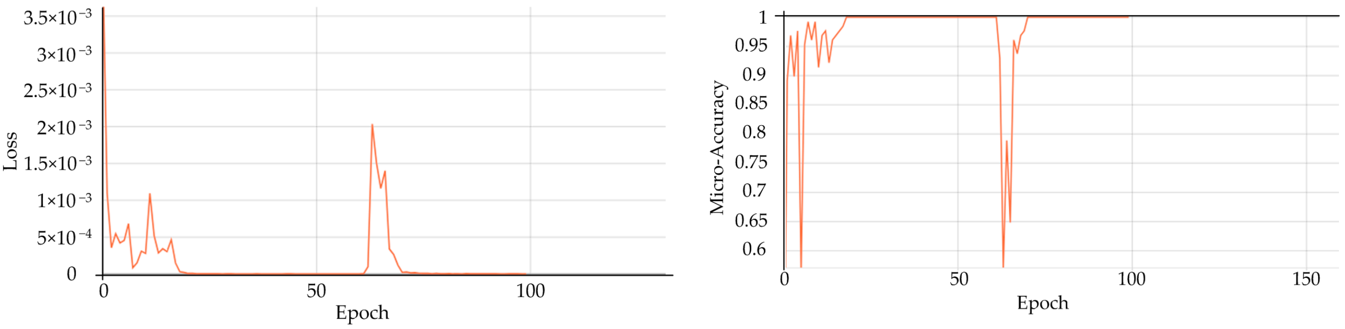
(**d**)

Figure 12. Training and testing results of the ResNet18 network model based on the proposed pipeline landmark datasets: (**a**) loss variation curve per step during training with smoothness of 0.999; (**b**) loss variation curve per epoch during training with smoothness of 0; (**c**) prediction accuracy variation curve per step during testing with smoothness of 0.999; (**d**) prediction accuracy variation curve per epoch during testing with smoothness of 0.

From Figure 12a,c, it can be observed that, overall, the model converges with training, and the predictions for pipeline landmarks become increasingly accurate. However, Figure 12b,d show that after 30 epochs, the model still occasionally produces low-accuracy predictions and does not achieve perfect accuracy. Analysis indicates that this is due to the existence of small batch sizes of images causing significant parameter changes during each epoch's training. Given the total dataset of 908 photos, the training set was formed by randomly selecting 80% consisting of 726 photos. With a "batch_size" of 64, there will be a step with only 22 photos (726−64 × 11 = 22). Due to the small number of samples in this step, the model cannot accurately learn all the characteristics of the pipeline landmarks, causing significant parameter fluctuations and resulting in drastic accuracy variations in the next test output. A simple improvement is to skip the training step with 22 samples. The code is as follows:

```
train_dataloader=DataLoader(train_ds,batch_size=64,shuffle=True,drop_la
st=True)
```

Using "drop_last=True" will skip the steps with fewer samples. The improved training loss curve for each epoch is shown in Figure 13a, and the testing accuracy curve for each epoch is shown in Figure 13b.



(**a**) (**b**)

Figure 13. Training and testing results after excluding small sample sizes: (**a**) loss variation curve per epoch during training with smoothness of 0; (**b**) prediction accuracy variation curve per epoch during testing with smoothness of 0.

As shown in Figure 13b, after skipping the steps with fewer samples to eliminate their periodic impact on each training epoch, the model's prediction accuracy stabilizes around 1 after approximately 20th epochs. This indicates an improvement. However, it does not completely eliminate prediction inaccuracies, as evidenced by a prediction failure around the 60th epoch, where the accuracy suddenly drops to about 0.6.

The training and testing results indicate that using the improved residual network model ResNet18 based on the pipeline landmark dataset proposed in this paper can achieve high-performance recognition and classification of various pipeline landmarks. The micro-accuracy exceeds 0.98, and if cases with small training sample sizes are excluded, the micro-accuracy can consistently reach 1. This validates the effectiveness of using deep learning networks for the recognition and classification of various joint structures in pipelines. The optimal model, such as the one trained up to the 40th generation, can also be selected, and predictions on all test set samples can then be made, as the confusion matrix obtained, as shown in Table 4, and the corresponding evaluation metrics, as shown in Table 5.

Table 4. Confusion matrix on the validation and test set.

	Straight Pipe	Curved Pipe	T-Shaped Pipe I	T-Shaped Pipe II
Straight pipe	43	0	0	0
Curved pipe	0	53	0	0
T-shaped pipe I	0	0	42	0
T-shaped pipe II	0	0	0	44

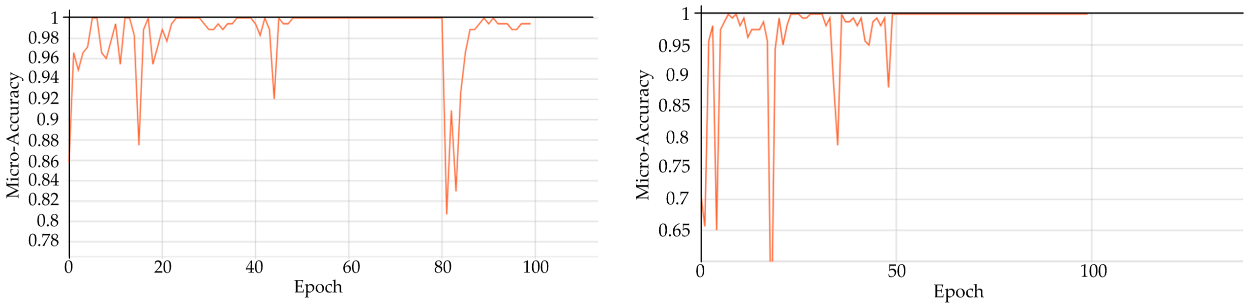
Table 5. Evaluation metrics on test set.

	Straight Pipe	Curved Pipe	T-Shaped Pipe I	T-Shaped Pipe II
TP	43	53	42	44
FN	0	0	0	0
FP	0	0	0	0
TN	139	129	140	138
Presicion	1	1	1	1
Recall	1	1	1	1
F1 score	1	1	1	1

5. Influence of HyperParameter "batch_size"

The hyperparameter "batch_size" has a significant impact on the training and testing results. For the pipeline landmark classification discussed in this paper, "batch_size" was set to 64, which can achieve the best testing result after many experiments. When the "batch_size" is smaller, the model cannot fully learn all the features of the pipeline landmarks in each step of training, leading to lower micro-accuracy after testing at each epoch. Conversely, if the "batch_size" is set larger, more computational resources are required for training.

As shown in Figure 14, the test accuracy curves for "batch_size" settings of 16 and 32, respectively, are displayed for both in the training process and testing process with "drop_last" set to "True" to discard small sample effects. As shown in Figure 14a, when the hyperparameter "batch_size" of training and testing processed is reduced to 32, the model cannot fully learn the dataset knowledge in each step. Therefore, it takes until around 50 epochs to accurately predict 1. Further reducing the "batch_size" to 16, as shown in Figure 14b, results in the model learning even less knowledge per step. Thus, even though the model can accurately predict to 1 by the end of 50 epochs, the periodic training with small sample sizes causes the model parameters to change, leading to inaccuracies in predictions around 80 epochs. So, according to Figure 13b, it is evident that for the pipeline landmark classification discussed in this paper, the better "batch_size" value can be selected as 64.



(**a**) (**b**)

Figure 14. Impact of different values of "batch_size" on test results: (**a**) prediction accuracy variation curve per epoch during testing with batch_size=32; (**b**) prediction accuracy variation curve per epoch during testing with batch_size=16.

6. Conclusions

Based on the proposed miniature pipeline robot system π-II, this paper explored the use of a monocular fisheye camera to capture the interior of four common categories of pipeline landmarks. The established dataset demonstrates that the lighting and inner wall environment inside the pipes during actual experiments is very complex, with significant differences between the captured landmark shadows and the ideally assumed shapes. This discrepancy makes it difficult to use traditional pattern-matching methods to identify corresponding pipeline landmarks. Therefore, a residual network model, ResNet18, which is more suitable for handling classification problems, was employed. Experiments showed that after each epoch of training, the micro-accuracy increased to above 0.98 when testing other samples. This paper delved into the impact of periodically occurring small sample sizes on training performance and discussed the influence of the hyperparameter "batch_size". The experiments demonstrated that the use of deep learning methods can accurately classify common pipeline landmarks, providing a foundation for subsequent research on pipeline mapping and autonomous navigation of pipeline robots within miniature pipelines.

The self-made pipeline landmark dataset used in this paper consists of 908 photos, which is relatively small in scale and specifically targets the stainless-steel pipeline shown in Figure 1. The types of pipeline landmarks are limited to four common types and do not cover all possible types. Therefore, to ensure that the detection system or neural network model developed has broader application value, it is necessary to create a larger and more comprehensive pipeline landmark dataset in future work. The success of the experiments in this paper indicates that this deep learning detection method is still feasible for broader applications.

Future work will focus on two aspects: One is focusing on more complex neural network models, not only classifying pipeline landmarks but also predicting the distance to the landmark and the direction of the landmark opening based on such monocular fisheye camera. The other is to apply this vision detection system to the autonomous control of robots navigating within pipelines. Especially for curved pipes and T-shaped pipes, only by accurately detecting the robot's distance and angle relative to these landmarks can the robot be controlled to navigate smoothly through such structures.

Author Contributions: Conceptualization, J.W. (Jian Wang) and S.W.; methodology, C.C.; software, J.W. (Jian Wang) and B.L.; validation, J.W. (Jian Wang), S.W. and J.W. (Juezhe Wang); formal analysis, J.W. (Jian Wang) and C.C.; investigation, B.L.; resources, J.W. (Jian Wang) and S.W.; data curation, B.L.; writing—original draft preparation, J.W. (Jian Wang); writing—review and editing, J.W. (Jian Wang) and S.W.; visualization, J.W. (Juezhe Wang); supervision, S.W.; project administration, J.W. (Jian Wang); funding acquisition, J.W. (Jian Wang). All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Specific Research Project of Guangxi for Research Bases and Talents, grant number AD20159061.

Data Availability Statement: The code and dataset used to support the findings of this study are available from https://github.com/445940846/pipelines (accessed on 14 July 2024).

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Jun, C.; Deng, Z.Q.; Jiang, S.Y. Study of locomotion control characteristics for six wheels driven in-pipe robot. In Proceedings of the 2004 IEEE International Conference on Robotics and Biomimetics, Shenyang, China, 22–26 August 2004; pp. 119–124. [CrossRef]
2. Kim, H.M.; Choi, Y.S.; Lee, Y.G.; Choi, H.R. Novel mechanism for in-pipe robot based on a multiaxial differential gear mechanism. *IEEE/ASME Trans. Mechatron.* **2017**, *22*, 227–235. [CrossRef]
3. Kwon, Y.S.; Lee, B.; Whang, I.C.; Kim, W.K.; Yi, B.J. A flat pipeline inspection robot with two wheel chains. In Proceedings of the 2011 IEEE International Conference on Robotics and Automation Shanghai International Conference Center, Shanghai, China, 9–13 May 2011; pp. 5141–5146. [CrossRef]

4. Kwon, Y.S.; Lee, B.; Whang, I.C.; Yi, B.J. A pipeline inspection robot with a linkage type mechanical clutch. In Proceedings of the 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems, Taipei, Taiwan, 18–22 October 2010; pp. 2850–2855. [CrossRef]
5. Wang, J.; Wu, H.; Wang, J.Z. Design of a small-type wheeled pipeline robot driven by monocular vision. In Proceedings of the 2023 IEEE 7th Information Technology and Mechatronics Engineering Conference, Chongqing, China, 15–17 September 2023; pp. 1133–1137. [CrossRef]
6. Chen, S.; Hu, H.; Guan, W.; Huang, C.; Gong, W.; Xu, X. Research on vision control system of pipeline robot. In Proceedings of the 2021 China Automation Congress, Beijing, China, 22–24 October 2021; pp. 5379–5384. [CrossRef]
7. Zhang, X.; Chen, H. Visual servo technique of autonomous mobile robot in bended pipe. In Proceedings of the 2002 IEEE International Conference on Industrial Technology, Bangkok, Thailand, 11–14 December 2002; pp. 588–593. [CrossRef]
8. Ahrary, A.; Tian, L.; Kamata, S.; Ishikawa, M. Anautonomous sewer robots navigation based on stereo camera information. In Proceedings of the 17th IEEE International Conference on Tools with Artificial Intelligence, HongKong, China, 14–16 November 2005. [CrossRef]
9. Lee, J.S.; Roh, S.G.; Kim, D.W.; Moon, H.; Choi, H.R. In-pipe robot navigation based on the landmark recognition system using shadow images. In Proceedings of the 2009 IEEE International Conference on Robotics and Automation, Kobe, Japan, 12–17 May 2009; pp. 1857–1862. [CrossRef]
10. Thielemann, J.T.; Breivik, G.M.; Berge, A. Pipeline landmark detection for autonomous robot navigation using time-of-flight imagery. In Proceedings of the 2008 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, Anchorage, AK, USA, 23–28 June 2008; pp. 1–7. [CrossRef]
11. Chand, A.N.; Zuhdi, N.; Mansor, A.; Iqbal, A.; Rustam, F.; Baur, W. An industrial robot for fire water piping inspection and mapping. In Proceedings of the 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems, Prague, Czech Republic, 27 September 2021; pp. 2337–2344. [CrossRef]
12. Choi, Y.S.; Kim, H.M.; Suh, J.S.; Mun, H.M.; Yang, S.U.; Park, C.M.; Choi, H.R. Recognition of inside pipeline geometry by using PSD sensors for autonomous navigation. In Proceedings of the 2014 IEEE/RSJ International Conference on Intelligent Robots and Systems, Chicago, IL, USA, 14–18 September 2014; pp. 5024–5029. [CrossRef]
13. Lee, D.H.; Moon, H.; Choi, H.R. Landmark detection of in-pipe working robot usingline-laser beam projection. In Proceedings of the ICCAS 2010, Gyeonggi-do, Republic of Korea, 27–30 October 2010; pp. 611–615. [CrossRef]
14. Lee, D.H.; Moon, H.; Koo, J.C.; Choi, H.R. Map building method for urban gas pipelines based on landmark detection. *Int. J. Control Autom. Syst.* **2013**, *11*, 127–135. [CrossRef]
15. Lee, D.H.; Moon, H.; Choi, H.R. Autonomous navigation of in-pipe working robot in unknown pipeline environment. In Proceedings of the 2011 IEEE International Conference on Robotics and Automation, Shanghai, China, 9–13 May 2011; pp. 1559–1564. [CrossRef]
16. Choi, Y.S.; Kim, H.M.; Mun, H.M.; Lee, Y.G.; Choi, H.R. Recognition of pipeline geometry by using monocular camera and PSD sensors. *Intel. Serv. Robot.* **2017**, *10*, 213–227. [CrossRef]
17. Haurum, J.B.; Bahnsen, C.H.; Pedersen, M.; Moeslund, T.B. Water level estimation in sewer pipes using deep convolutional neuralnetworks. *Water* **2020**, *12*, 3412. [CrossRef]
18. Patil, R.R.; Mustafa, M.Y.; Calay, R.K.; Ansari, S.M. S-BIRD: A novel critical multi-class imagery dataset for sewer monitoring and maintenance systems. *Sensors* **2023**, *23*, 2966. [CrossRef] [PubMed]
19. Zhao, Y.; Su, Z.; Zhao, H. Micro-leakage image recognition method for internal detection in small, buried gas pipelines. *Sensors* **2023**, *23*, 3956. [CrossRef] [PubMed]
20. Wang, J.; Mo, Z.; Cai, Y.; Wang, S. Kinematic analysis of a wheeled-leg small pipeline robot turning in curved pipes. *Electronics* **2024**, *13*, 2170. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.