

## Pipeline Defect Detection Cloud System Using Role Encryption and Hybrid Information

**Ce Li<sup>1, 2,\*</sup>, Xinyu Shang<sup>2</sup>, Liguo Zhang<sup>3, 4</sup>, Feng Yang<sup>1, 2</sup>, Jing Zheng<sup>1, 5</sup> and Xianlei Xu<sup>1</sup>**

**Abstract:** Pipeline defect detection systems collect the videos from cameras of pipeline robots, however the systems always analyzed these videos by offline systems or humans to detect the defects of potential security threats. The existing systems tend to reach the limit in terms of data access anywhere, access security and video processing on cloud. There is in need of studying on a pipeline defect detection cloud system for automatic pipeline inspection. In this paper, we deploy the framework of a cloud based pipeline defect detection system, including the user management module, pipeline robot control module, system service module, and defect detection module. In the system, we use a role encryption scheme for video collection, data uploading, and access security, and propose a hybrid information method for defect detection. The experimental results show that our approach is a scalable and efficient defection detection cloud system.

**Keywords:** Defect detection, cloud computing, data encryption, hybrid information.

### 1 Introduction

Traffic pipes are like blood vessels in a city, and the reliability of pipeline is more and more important. Once the city's pipeline system occurs fault, the normal life of citizens will be affected and the city will get plenty of losses. Thus, the pipeline defect detection and repair is an important problem of urban construction [Ng (2006)]. Most researchers get the pipeline video by pipeline robots with the closed circuit television (CCTV) hardware, and they always analyze the videos by off-line systems or humans to detect the defects of potential security threats. It takes a lot of time and costs a lot of money with low efficiency and low accuracy. The existing systems tend to reach the limit in terms of data access anywhere, access security and video processing on cloud. There is in need of studying on a pipeline defect detection cloud system for automatic pipeline inspection.

---

<sup>1</sup> State Key Laboratory of Coal Resources and Safe Mining, China University of Mining & Technology, Beijing, 100083, China.

<sup>2</sup> School of Mechanical Electronic & Information Engineering, China University of Mining & Technology, Beijing, 100083, China.

<sup>3</sup> Computer Science & Engineering, Harbin Engineering University, Harbin, 150001, China.

<sup>4</sup> Computer Science & Engineering, Hong Kong University of Science and Technology, Hong Kong.

<sup>5</sup> Stanford University, Palo Alto, CA 94305-6104, USA.

\* Corresponding Author: Ce Li. Email: celi@cumtb.edu.cn.

Some researchers have attempted defect detection by computer vision and cloud computing for pipeline inspection [Xie, He, Xie et al. (2018)], and some have argued that feature detection may not need a cloud approach [Shi, Zhang, Li et al. (2015); Anjum, Abdullah, Tariq et al. (2016); Boaretto and Centeno (2017)]. Nevertheless, with the development of cloud computing, multiple computers are parallelized through network, which increases the computing speed greatly for defect detection [Ali, Huda, Abawajy et al. (2017); Hawari, Alamin, Alkadour et al. (2018); Lin-Lin and Zhen (2009); Srivastava and Kumar (2015)]. In this paper, we propose a new method for pipeline defect detection and deploy a framework of cloud-based pipeline defect detection system.

The major work for Pipeline video processing includes image preprocessing, defect location and defect feature recognition [Wang, Chen, Qiao et al. (2018); Wan, Wei, Jiao et al. (2018)]. In this paper, we preprocess the pipeline image using image grayscale conversion, grayscale stretching, smoothing filter and canny edge detection, then extract the defect feature using histograms of oriented gradients (HOG) and visual geometry group network (VGGNet) [Zhou, Liang, Li et al. (2018)], and finally locate and recognize the defect with support vector machine (SVM) [Zhang, Li, Lu et al. (2016); Wiatowski and Bolcskei (2018)].

The rest of this paper is organized as following. Section 2 is a description about the related work, and Section 3 explains the framework of a cloud-based defect detection system of pipeline. Section 4 introduces the method for defect recognition in pipeline. In Section 5, the experimental result and analysis are given, and conclusion in Section 6.

## **2 Related work**

Defect detection of pipeline over cloud is a new research area. Performance and high availability have become increasingly important drivers. Cloud computing is the core technology of the next generation of network computing platform [Al Tayeb (2013)]. Jamkhedkar et al. studied the security on demand in cloud computing [Jamkhedkar, Szefer, Perezbotero et al. (2013)]. Garg et al. [Garg, Versteeg, Buyya et al. (2013)] shown a framework where customers' security requests can be expressed and enforced by leveraging the capabilities of servers with different security architectures. Valentini et al. [Valentini, Lassonde, Khan et al. (2013)] discussed the two major constraints for cloud computing: operational costs, and system reliability. These directions could be explored to improve the energy efficiency in cluster computing.

Pipeline image processing and defect detection algorithm are important to pipeline defect recognition. Literature review shows that there are many research results. Cosham et al. [Cosham, Hopkins, Macdonald et al. (2007)] proposed the best techniques currently available for the assessment of pipeline defects, and gave guidance on their use. An automatic image-based crack detection is proposed as a replacement by Mohan et al. [Mohan and Poobal (2018)]. Jiang et al. [Jiang, Tian, Farrell et al. (2017)] explored a method that combines machine learning and image systems simulation that automates the pipeline defect detection. Gao et al. [Gao, Liu, Ma et al. (2018)] studied the DIF and proposed cross-correlation via the differentiation and filtering process. Guo et al. [Guo, Soibelman, Garrett et al. (2009)] discussed the challenges of automated defect detection and proposed a general detection and recognition model for pipeline defect detection.

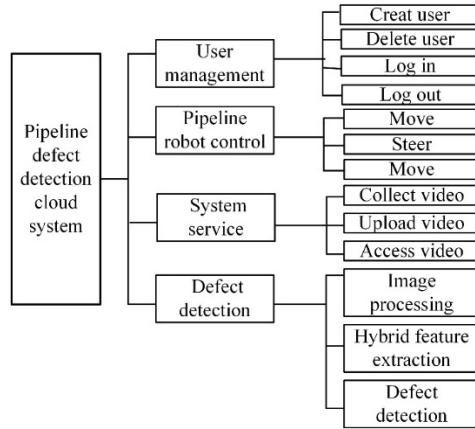
Chang et al. [Chang and Lin (2011)] introduced a library for support vector machine to defect classification. Levin et al. [Levin, Nadler, Durand et al. (2012)] studied the image restoration. They presented a law of diminishing return, namely that with increasing patch size, rare patches not only require a much larger dataset, but also gain little from it. The result shows that novel adaptive variable-sized patch schemes for image restoration. A simple and scalable detection algorithm was proposed by Girshick et al. [Girshick, Donahue, Darrell et al. (2014)], which combines two key insights: one can apply high-capacity CNNs to bottom-up region proposals to localize and segment objects and when labeled training data is lacking, supervised pre-training for an auxiliary task. And with fine-tuning, the performance will have a great progress. Wang et al. [Wang, Guo, Huang et al. (2015)] found that VGGNet is effective for object recognition in still images. The report describes the models trained by them achieved great performance on the dataset. Roy et al. [Roy and Chaudhuri (2016)] concentrated on the edge detection, which is a subset of segmentation to detect object of interest. The paper discussed the different performance measures of edge detection and proposed application on a hybrid filter using Bilateral and Canny.

### **3 Framework for the cloud-based defect detection system**

To build a cloud-based defect detection system, there are some issues need to be considered. First of all, we need to figure out what the system require. Then we have to take the system architecture design into account. Besides, we should think about the core system modules. The framework for the cloud-based defect detection system is described as follows.

#### **3.1 System analysis**

Through analyzing the cloud-based defect detection system, there are some requirements. The main requirements are listed in Fig. 1. User management, pipeline robot control, system service and defect detection. Creating and deleting system users, log in and log out compose the user management. With the heterogeneous devices, users can access to the system from anywhere. The user manager is very convenient for the users. When the abnormal event happened, the heterogeneous video providers should inform the users. As a result, the providers, networks and devices are necessary for the system. Through pipeline robot control, users are able to control robot movement and steering. System service including collecting video, uploading video and accessing video. Users can collect the real and effective pipeline defect video with the help of system service. Defect detection manager module is the image processing, hybrid feature extraction and defect detection management of cloud-based operations, helping the users detect the defect in the pipeline and assess the pipeline condition.



**Figure 1:** The chart of system analysis

### 3.2 System design

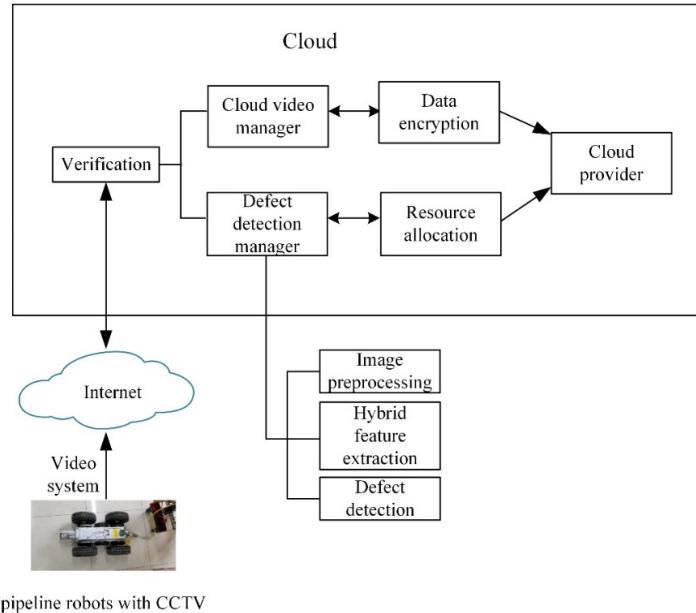
The architecture of the designed cloud-based defect detection framework is showed in Fig. 2. We can deploy the system on a private cloud. There are two major modules in the cloud, cloud video manager and defect detection manager. Notification is an important part of the system framework in the cloud. If the defect detection system finds some events interest, users will be informed. The system has great scalability. Lots of video providers and detection users are able to connect the defect detection system.

While the cloud system provides high efficiency and scalability, how to effectively prevent personal privacy disclosure, authorized access, cloud data security de-emphasis, and key update and privilege revocation issues also pose greater challenges for cloud service providers [Liu, Peng, Wang et al. (2018)].

More specifically, for storage, users encrypt the original data and upload the encrypted file from the collecting devices to the cloud [Xiong and Shi (2018)]. For access, users need to apply the cloud to obtain the encrypted file, then decrypt and download it. The encryption and decryption algorithms are six steps. First, the system creates a session key between a video owner and the cloud. Second, the video owner chooses a device and send the video along with a signing message. Third, the cloud verifies the authentication code via the signing message. Fourth, the cloud encrypts the identity and data after verification. Fifth, the owner uploads the encrypted files. Sixth, other users can have access to the data and decrypt it only under specific state. More details are given in the work [Yang, He and Lin et al. (2017)].

In Fig. 2, cloud video manager, defect detection manager, and data encryption are core system modules. In the cloud video manager module, the cloud providers connect different types of devices to collect, transmit, and store large amount of videos. In the module of defect detection manager, image preprocessing is employed to eliminate the effects of low resolution, different sizes and non-uniform illumination on each image of videos, which is implemented with grayscale conversion, grayscale stretching, spatial smooth filtering, and region segmentation. Inspired by the hybrid feature extraction as a kind of popular feature fusion method [Hao, Wang, Li et al. (2015)], we propose to

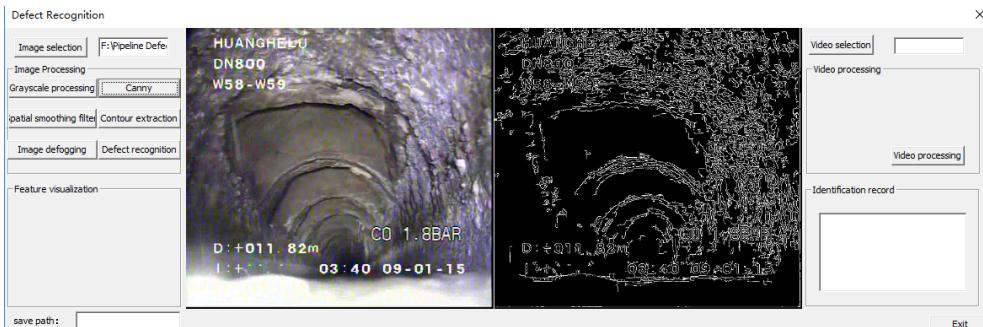
extract the hybrid information of defects using histogram of oriented gradient (HOG) and VGGNet descriptors. On the basis of hybrid feature, defect detection is finally implemented by SVM classifier. The data encryption module is adopted for security of data access and storage, in which the session key and digital signature are used for encryption and verification.



**Figure 2:** The designed architecture for a pipeline defect detection cloud system

### 3.3 System prototype development

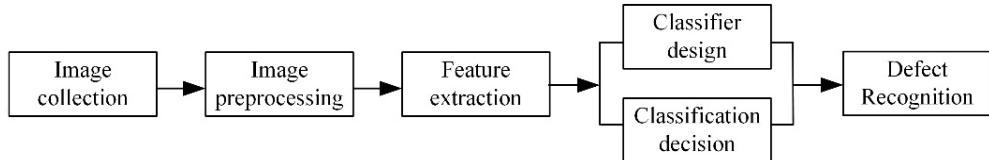
The prototype system is developed on a private cloud platform. The detection manager is deployed and CCTV system service is used to capture the image. Users can upload the pipeline image or pipeline video to the cloud. Then users are able to choose image processing method or defect recognize freely. In Fig. 3, there is an example of the user choose canny to process the image.



**Figure 3:** User choose canny to process the image in the system

#### 4 Proposed method for defect recognition in pipeline

Image recognition is the processing and analysis of various forms of information (shapes, colors, and textures) that characterize an image, and the process of describing, identifying, classifying and interpreting images. It is an important part of artificial intelligence and information science. Therefore, an image recognition system usually consists of four parts: image collection, image preprocessing, feature extraction, and image recognition. As shown in Fig. 4.



**Figure 4:** Image recognition system architecture

##### 4.1 Information collection system

Image collection is to upload the collected pipeline information to the cloud through the CCTV (closed circuit television) device. The video acquisition system is generally carried by the pipeline robot. The system is generally composed of a CCD camera, a transmission module, a display and a control unit. The camera is installed on the platform of the pipeline robot. By using the platform, the camera can achieve 360-degree rotation and axial 220-degree rotation. The image effect of the camera is determined by its resolution. In order to obtain clear pipeline image, this system adopts high resolution camera. The camera transmits the collected information to the cloud through the network transmission module. The acquisition system is shown in Fig. 5.



**Figure 5:** Acquisition system interface

##### 4.2 Image preprocessing

Due to the extremely complex environment of the pipeline, the features of the pipeline defect also have large differences, such as low resolution, different sizes and non-uniform illumination. These factors increase the difficulty of image segmentation processing and defect feature recognition, which causes inconvenience. Therefore, it is necessary to

preprocess the image before defect location and defect feature recognition. By preprocessing the image, the effect of improving the defect location and recognition accuracy can be achieved.

#### *4.2.1 Image grayscale conversion*

The grayscale image is different from the RGB image, and the color information of each pixel is no longer three components, but only one. The gray level of grayscale image is quantization level, such as 64 level, 256 level, etc. The gray level indicates the degree of gray level subdivision. The RGB image is converted to a grayscale image for the image segmentation. The formula for converting a color image into a grayscale image is

$$f(x, y) = 0.299R(x, y) + 0.587G(x, y) + 0.114B(x, y) \quad (1)$$

In Eq. (1),  $f(x, y)$  denotes the gray value of the pixel  $(x, y)$ . We calculate the brightness value of the pixel according to the RGB value of each pixel and the fixed weight. The weight of the formula, which is a generally accepted conversion formula, is determined by the sensitivity of the human eye to RGB three-color light. The used formula is the most consistent with true color.

#### *4.2.2 Image grayscale stretching*

In order to enhance the contrast between the interest area and the normal pipeline, the grayscale image needs to be grayscale stretched. Image grayscale stretching can make the brightness and darkness more vivid, the features are more obvious. And the image will be more conducive to the next image segmentation. In this paper, a simple processing method is adopted, in which the gray value  $f(x, y)$  of each pixel in the input image is the independent variable of the function, and the resulting dependent variable is the gray value  $g(x, y)$  of the output image, that is

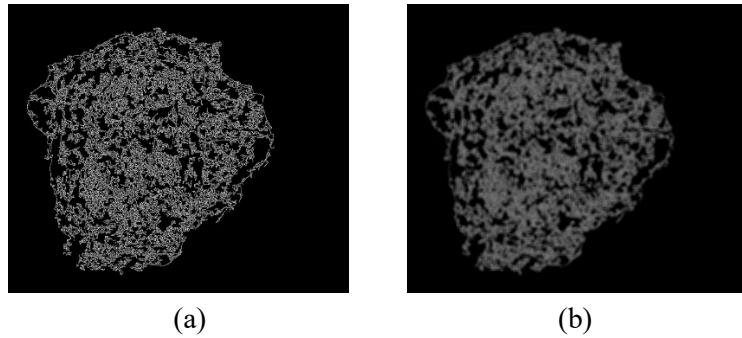
$$g(x, y) = T[f(x, y)] \quad (2)$$

According to different conditions and requirements, different mapping functions can be selected, such as proportional function, exponential function and so on. Such a method of inputting a point and obtaining a point of the output after a finite number of operations is called a point operation. In this paper, we use linear gradation transformation to enhance the contour recognition of suspected defect areas.

#### *4.2.3 Spatial smoothing filtering*

In any analog image, the image quality of the image is reduced during the process of being converted into digital image. That is the influence of noise during sampling, quantization and transmission. Due to various factors, such as uneven light caused by point light source, local highlight, metal reflection and so on. One value of the geometric feature is the perimeter of the suspected defect profile. Without the spatial smoothing filter enhancement, the contour we directly get will have a lot of serrated edges, which greatly increases the circumference and leads to experimental error. Fig. 6(a) and Fig. 6(b) show the binary image obtained by smoothing with a  $3 \times 3$  template and the binary image

without smoothing. It can be clearly observed in Fig. 6 that the smoothed contour has no significant serrated edge.



**Figure 6:** Spatial smoothing filter enhancement

#### 4.2.4 Image segmentation

Image segmentation is the technique and process of segmenting an image or scene into specific, unique parts or subsets and extracting the target of interest according to certain principles. In this experiment, our target of interest is the blocky area of suspected defection. In this paper, because the internal texture of the disease is also quite complex, and the target is only the outermost contour, the segmentation effect based on similarity is not satisfactory, so the segmentation based on discontinuity is used to find the edge of the disease. This paper mainly introduces Canny operator. The optimal edge contour is learned by Canny operator with four steps.

In the first step, Gaussian filtering is used to remove image noise. Gaussian filtering process is actually the process of weighted average calculation of the entire image. We use the size  $2N+1$  template (or convolution kernel, mask) to scan each pixel in the image and use the template to determine the neighborhood pixel weighted average grayscale to replace the template center pixel grayscale value. The one-dimensional Gaussian function is expressed as

$$G(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-(x-x_0)^2/2\sigma^2} \quad (3)$$

The Gaussian convolution kernel parameters of a given size and Sigma are calculated according to one-dimensional Gaussian function, and the calculated Gaussian convolution kernel is convolved with the gray image.

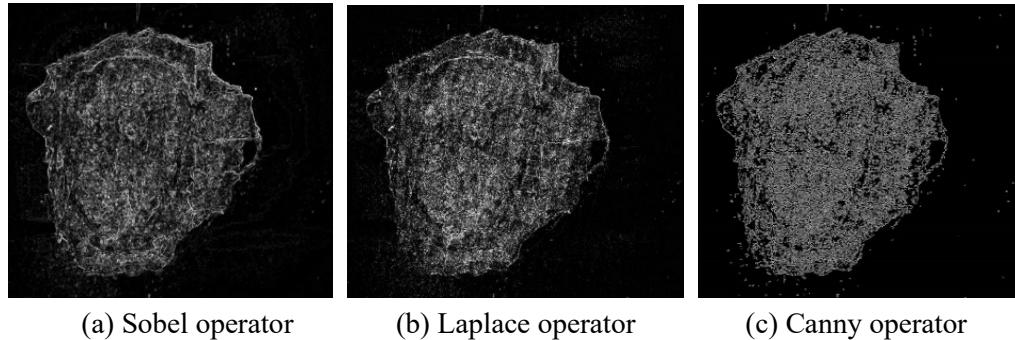
The second step is to calculate the gradient amplitude and direction. The horizontal and vertical templates are described as following.

$$\begin{aligned} P[i, j] &= (f[i+1, j] - f[i, j] + f[i+1, j+1] - f[i, j+1]) / 2 \\ Q[i, j] &= (f[i, j] - f[i, j+1] + f[i+1, j] - f[i+1, j+1]) / 2 \\ M[i, j] &= \sqrt{P[i, j]^2 + Q[i, j]^2} \\ \theta[i, j] &= \arctan(Q[i, j] / P[i, j]) \end{aligned} \quad (4)$$

In the Eq. (4),  $f(x, y)$  is the image gray value,  $P$  is the gradient amplitude in the  $X$  direction,  $Q$  is the gradient amplitude in the  $Y$  direction,  $M$  is the amplitude at this point, and  $\theta$  is the gradient Angle. After obtaining the gradient and gradient Angle in the  $X$  and  $Y$  directions, the gradient amplitude of the fusion of  $X$  and  $Y$  directions can be calculated. The calculation formula is

$$G = \sqrt{G_x^2 + G_y^2} \quad (5)$$

The third step, non-maximum suppression. Non-maximum suppression is applied for edge refinement. After gradient calculation, the edges extracted from the gradient values are still very blurry. We compare the edge intensity of the current pixel with the edge intensity of the pixel in the positive gradient direction and the negative gradient direction. This value will be preserved if the edge intensity of the current pixel is the largest compared to other pixels in the mask having the same direction. Otherwise, the value will be suppressed.



**Figure 7:** Comparison of Sobel, Laplace, Canny edge detection operators

The fourth step, double threshold and edge detection. After double threshold and edge detection, the canny image is obtained. We have compared the results of three operators for edge detection and connection as shown in Fig. 7. Note that the Canny operator is the empirically best choice in our experiment.

#### 4.3 Hybrid information

In the recognition system, the description of defect features directly affects the classification effect of the entire system. In the process of feature selection, the main features of the expression and understanding of the image information of thousands of images should be selected [Targ, Almeida and Lyman (2016); Meng, Rice, Wang et al. (2018)]. The selected features should be able to describe the defect reliably and independently. After experiment and comparison, the HOG characteristics of the image can better describe the morphological characteristics of the pipeline defect picture. At the same time, the transfer learning strategy was used to train the VGGNet model on the small-scale defect dataset. VGGNET can express the shape and texture characteristics of the image better. HOG and VGGNET are combined as classification features of classifiers, and good classification results can be obtained under small-scale data sets.

#### *4.3.1 Histogram of oriented gradient*

HOG (Histogram of Oriented Gradient) is a computer visual feature description for target detection. It forms features by counting and calculating histograms of gradient directions in local areas of the image. The HOG feature extraction algorithm is computed on the local grid element of the target image, so it can maintain better invariant characteristics than the geometric and optical changes of the target image, and the geometric and optical changes are only reflected in the larger space.

#### *4.3.2 VGGNet convolutional neural network*

The VGG convolutional neural network was developed from the Alex convolutional neural network. Compared to the Alex convolutional neural network, VGGNet has the following improvements:

All convolution operations in the network use a  $3 \times 3$  convolution kernel. The convolutional layer uses multiple small convolution kernels instead of one large convolution kernel to reduce the parameters of the entire network. On the other hand, it is equivalent to more nonlinear mapping, which increases the network's ability to fit and express. The structure alternating between multiple convolutional layers and non-linear active layers can extract deeper features than the single convolutional layer structure. It is better to select a convolution layer combination with a small filter rather than a convolution layer with a large filter. The former can represent more powerful functionality in the input data and use fewer parameters.

In the convolution structure of VGG network, the convolution kernel of  $1 \times 1$  was introduced. This operation did not affect the output dimension, but introduced non-linear transformation, which increased the expression capacity of VGG network and reduced the computational amount of the model.

In the model training stage, the A-level network of VGGNet with simple training level (with few layers) is adopted first, and then the weight of the network training is used to initialize the complex model, so as to accelerate the convergence speed of training

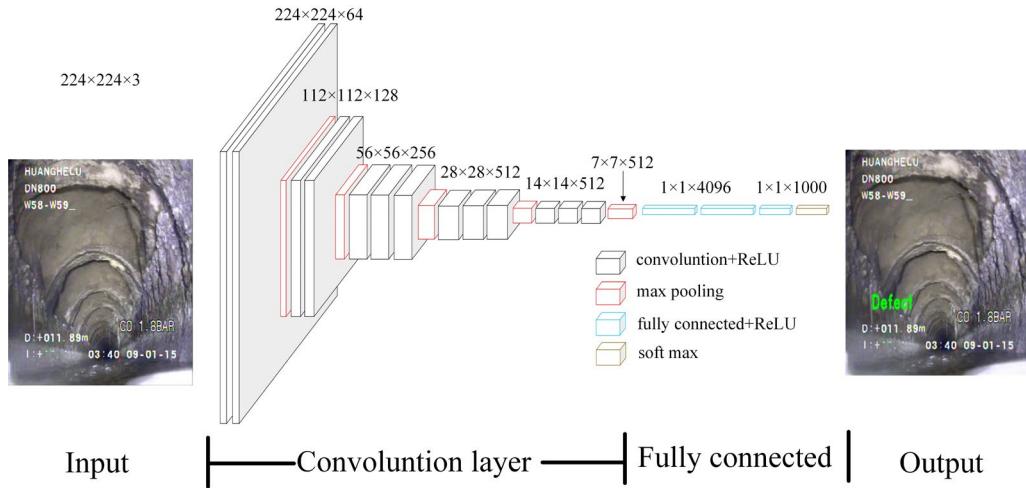
The multi-scale method is adopted in both the training and prediction of the model, which can increase the amount of training data, prevent the model from overfitting and improve the accuracy of prediction.

Fig. 8 shows the structure diagram of VGGNet-16 adopted in this paper. Although the convolutional network at each level from A to E gradually becomes deeper in the series of VGGNet, the overall number of parameters of the convolutional net-work does not increase significantly, because the number of thousands of parameters is mainly consumed in the last three full connection layers.

Transfer learning strategy, the most common obstacle to solving problems using deep learning techniques is the large amount of data required for training models. The reason for the need for large amounts of data is that a large number of parameters in the model are encountered during machine learning. When faced with a particular scenario, the data required to build the model may not be available. However, the relationship to some type of data acquisition in the model training task can also be easily applied to different problems in the same domain, which is called transfer learning. Using the transfer

learning strategy, we can directly use models that others have pre-trained. This model has been pre-trained through a large and easily available data set. Then find the reusable layer of the output. We can use the output of these layers as input to train networks with fewer, smaller scale data sets. Another major benefit of transfer learning technology is the ability to improve the “universality” of the model. Large models tend to over-fitting data. Because the transfer learning strategy allows the model to know different types of data, the model can learn better basic rules.

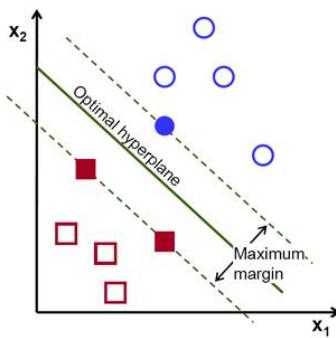
Feature extractor. The VGGNet network is very scalable, and the generalization ability to migrate to other image data is very good, so it can be used to extract image features. In this paper, the trained VGGNet feature extractor is used to extract features, then the HOG features are integrated, and the SVM classifier model is used for training and prediction.



**Figure 8:** The structure of VGGNet-16

#### 4.4 Defect recognition

After extracting the hybrid information of image, we introduce support vector machine (SVM) for defect recognition. Support vector machine is a new general learning method based on statistical theory and structural risk.



**Figure 9:** The schematic diagram of the SVM

For the case of linearly separable, it can be assumed as

$$\begin{cases} H_1 : (\omega^T x_i) + b \geq 1, y_i = 1 \\ H_2 : (\omega^T x_i) + b \leq -1, y_i = -1 \end{cases} \quad (6)$$

Normalized to

$$y_i(\omega^T x_i + b) \geq 1, i = 1, \dots, l \quad (7)$$

$\omega$  is the normal vector.  $b$  is the displacement term, which determines the distance between the hyperplane and the origin.  $x_i$  is the characteristic of each sample point.  $y_i$  is the category label of the sample point  $x_i$ . The distance between  $H_1$  and  $H_2$  to  $H$  is  $1/\|\omega\|$ . Naturally, the classification interval is  $2/\|\omega\|$ . The classification surface with the maximum and even minimum classification interval of  $2/\|\omega\|^2$  is called the optimal classification hyperplane, which is commonly referred to as the SVM classification decision plane. And the training sample points on  $H_1$  and  $H_2$  are called support vectors. Therefore, seeking the optimal  $(\omega, b)$  can be summarized as a quadratic programming problem:

$$\min \frac{1}{2} \|\omega\|^2 \quad (8)$$

$$\text{s.t. } y_i(\omega^T x_i + b) \geq 1, i = 1, 2, \dots, n$$

$n$  is the total number of sample points, the abbreviation s.t. means “Subject to”. The dual problem of the programming problem, that is, the solution of the maximized objective function can be obtained by introducing the Lagrangian optimization function. The corresponding classification decision function is

$$h(x) = \sum_{i=1}^n \alpha_i y_i x_i^T x + b \quad (9)$$

In the above formula,  $b$  is model parameter, and  $\alpha_i$  is a Lagrangian multiplier. For nonlinear cases, the basic idea of SVM is to map the vector to a high-dimensional feature space through a predetermined nonlinear mapping, and then construct an optimal hyperplane in this high-dimensional space.

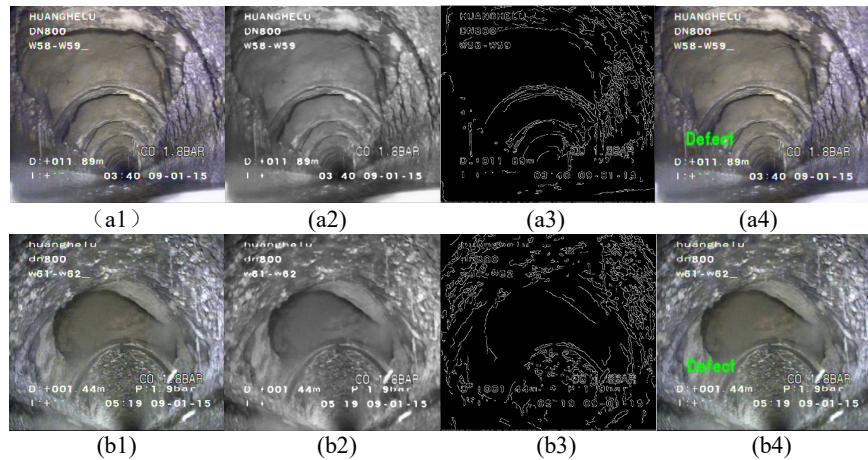
The advantage of SVM is that it is easy to obtain the nonlinear relationship between data and features in the small and medium sample size, which is highly interpretable and can solve high dimensional problems. SVM algorithm is not only simple, but also has good robustness. In this paper, we adopt SVM to identify pipeline defects using LibSVM toolbox.

## 5 Experimental result and analysis

In order to verify the effectiveness of the algorithm, experiments were carried out on video in real scene. They are carried out on the client of the developed system proto-type. We run the experiments on Core 4790K@4.0 GHz CPU unit, 4 CPU core, 8 GB memory, drive with 64-bit windows file system in Microsoft server. We used three videos to

validate the performance of pipeline defect recognition, which are collected from an actual pipeline with various corrosion in Zhengzhou province.

In Fig. 10, the 7470<sup>th</sup> frame and the 3270<sup>th</sup> frame in the original video are shown in Fig. 10(a1) and Fig. 10(b1), respectively. The smoothing filter enhanced images of the 7470<sup>th</sup> frame and the 3270<sup>th</sup> frame are displayed in Fig. 10(a2) and Fig. 10(b2). Two images detected by the canny operator are demonstrated in Fig. 10(a3) and Fig. 10(b3), and the results of our proposed defect identification are shown in Fig. 10(a4) and Fig. 10(b4). As shown in these subfigures, the noise in the images can be filtered out by gray processing and smoothing filter, and the images are segmented by the canny operator, then the defects features extracted using HOG and VGGNet are finally classified and identified by SVM. It is noted that the pipeline defects are different in the three videos and difficult to recognize considering of the lighting effects, texture characteristics, and the movement of pipeline robot. However, the system continues to perform well in this case, which verifies that the proposed cloud-based defect detection system can identify pipeline defects accurately. In short, the proposed method is efficient, convenient and highly accurate for cloud server to process pipeline videos and identify pipeline defects, which provides a new feasible way to replace the traditional manual maintenance method.



**Figure 10:** Pipeline defect recognition effect under real scene

## 6 Conclusion

In order to improve the efficiency and accuracy of pipeline detection, this paper develops a pipeline defect detection cloud system for automatic pipeline inspection. In this paper, a cloud based pipeline defect detection system is deployed with user management module, pipeline robot control module, system service module, and defect detection module. In this system, we use a role encryption scheme for video data collection, uploading, and access security. For defect detection, we introduce the hybrid information integrating VGGNet and HOG features and use SVM to identify pipeline defects. A prototype system is developed on the cloud architecture designed to verify the proposed method. Experimental results show that the system has high efficiency and accuracy in the

identification of pipeline defects, and it is a feasible method to replace artificial recognition. However, the effectiveness of defect recognition in different environments still needs to be improved, and the cost problem is still the key of the cloud-based defect recognition system. This motivates us to study on these application problems for the defect detection in the future.

**Acknowledgement:** The work was supported in part by the Fundamental Research Funds for the Central Universities (2016QJ04), Yue Qi Young Scholar Project of CUMTB, the State Key Laboratory of Coal Resources and Safe Mining (SKLCRSM16KFD04, SKLCRSM16KFD03), the Natural Science Foundation of China (61601466), the Natural Science Foundation of Beijing, China (8162035), the National Key R&D Program of China (2018YFC0807801), and the National Training Program of Innovation and Entrepreneurship for Undergraduates (C201804970).

## References

- Al Tayeb, A.; Alghatani, K.; El-Seoud, S.; El-Sofany, H.** (2013): The impact of cloud computing technologies in e-learning. *International Journal of Emerging Technologies in Learning*, vol. 8, no. 2013.
- Ali, M. M.; Huda, S.; Abawajy, J.; Alyahya, S.; Al-Dossari, H. et al.** (2017): A parallel framework for software defect detection and metric selection on cloud computing. *Cluster Computing*, vol. 20, no. 3, pp. 2267-2281.
- Anjum, A.; Abdullah, T.; Tariq, M.; Baltaci, Y.; Antonopoulos, N.** (2016): Video stream analysis in clouds: an object detection and classification framework for high performance video analytics. *IEEE Transactions on Cloud Computing*, pp. 1.
- Boaretto, N.; Centeno, T. M.** (2017): Automated detection of welding defects in pipelines from radiographic images DWDI. *NDT & E International*, vol. 86, pp. 7-13.
- Chang, C.; Lin, C.** (2011): LIBSVM: a library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, vol. 2, no. 3, pp. 27.
- Gao, Y.; Liu, Y.; Ma, Y.; Cheng, X.; Yang, J.** (2018): Application of the differentiation process into the correlation-based leak detection in urban pipeline networks. *Mechanical Systems and Signal Processing*, vol. 112, no. 2018, pp. 251-264.
- Garg, S. K.; Versteeg, S.; Buyya, R.** (2013): A framework for ranking of cloud computing services. *Future Generation Computer Systems*, vol. 29, no. 4, pp. 1012-1023.
- Girshick, R. B.; Donahue, J.; Darrell, T.; Malik, J.** (2014): Rich feature hierarchies for accurate object detection and semantic segmentation. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 580-587.
- Guo, W.; Soibelman, L.; Garrett, J. H.** (2009): Automated defect detection for sewer pipeline inspection and condition assessment. *Automation in Construction*, vol. 18, no. 5, pp. 587-596.
- Hao, P.; Wang, B.; Li, G.; Meng, Z.; Wang, L.** (2015): Hybrid framework for reliability-based design optimization of imperfect stiffened shells. *AIAA Journal*, vol. 53, no. 10, pp. 2878-2889.

- Hawari, A.; Alamin, M.; Alkadour, F.; Elmasry, M.; Zayed, T.** (2018): Automated defect detection tool for closed circuit television (CCTV) inspected sewer pipelines. *Automation in Construction*, vol. 89, pp. 99-109.
- Jamkhedkar, P. A.; Szefer, J.; Perezbotero, D.; Zhang, T.; Triolo, G. et al.** (2013): A framework for realizing security on demand in cloud computing. *IEEE International Conference on Cloud Computing Technology and Science*, vol. 1, pp. 371-378.
- Jiang, H.; Tian, Q.; Farrell, J. E.; Wandell, B. A.** (2017): Learning the image processing pipeline. *IEEE Transactions on Image Processing*, vol. 26, no. 10, pp. 5032-5042.
- Levin, A.; Nadler, B.; Durand, F.; Freeman, W. T.** (2012): Patch complexity, finite pixel correlations and optimal denoising. *European Conference on Computer Vision*, pp. 73-86.
- Lin-Lin, S.; Zhen, J. I.** (2009): Gabor wavelet selection and SVM classification for object recognition. *Acta Automatica Sinica*, vol. 35, no. 4, pp. 350-355.
- Liu, Y.; Peng, H.; Wang, J.** (2018): Verifiable diversity ranking search over encrypted outsourced data. *Computers, Materials & Continua*, vol. 55, no. 1, pp. 37.
- Meng, R.; Rice, S. G.; Wang, J.; Sun, X.** (2018): A fusion steganographic algorithm based on faster R-CNN. *Computers, Materials & Continua*, vol. 55, no. 1, pp. 1-16.
- Mohan, A.; Poobal, S.** (2018): Crack detection using image processing: a critical review and analysis. *Alexandria Engineering Journal*, vol. 57, no. 2, pp. 787-798.
- Ng, H. F.** (2006): Automatic thresholding for defect detection. *Pattern Recognition Letters*, vol. 27, no. 14, pp. 1644-1649.
- Roy, S.; Chaudhuri, S. S.** (2016): Error measurement & its impact on bilateral-canny edge detector-a hybrid filter. *International Journal of Modern Education and Computer Science*, vol. 8, no. 2, pp. 30-30.
- Shi, Y.; Zhang, C.; Li, R.; Cai, M.; Jia, G.** (2015): Theory and application of magnetic flux leakage pipeline detection. *Sensors*, vol. 15, no. 12, pp. 31036-31055.
- Srivastava, H.; Kumar, S. A.** (2015): Control framework for secure cloud computing. *Journal of Information Security*, vol. 6, no. 1, pp. 12.
- Targ, S.; Almeida, D.; Lyman, K.** (2016): ResNet in ResNet: generalizing residual architectures. arXiv preprint: 1603.08029.
- Valentini, G.; Lassonde, W.; Khan, S. U.; Minallah, N.; Madani, S. A. et al.** (2013): An overview of energy efficiency techniques in cluster computing systems. *Cluster Computing*, vol. 16, no. 1, pp. 3-15.
- Wan, F.; Wei, P.; Jiao, J.; Han, Z.; Ye, Q.** (2018): Min-entropy latent model for weakly supervised object detection. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1297-1306.
- Wang, T.; Chen, Y.; Qiao, M.; Snoussi, H.** (2018): A fast and robust convolutional neural network-based defect detection model in product quality control. *International Journal of Advanced Manufacturing Technology*, vol. 94, no. 9-12, pp. 3465-3471.
- Wang, L.; Guo, S.; Huang, W.; Qiao, Y.** (2015): Places205-VGGNet models for scene recognition. arXiv preprint: 1508.01667.

**Wiatowski, T.; Bolcskei, H.** (2018): A mathematical theory of deep convolutional neural networks for feature extraction. *IEEE Transactions on Information Theory*, vol. 64, no. 3, pp. 1845-1866.

**Xie, R.; He, C.; Xie, D.; Gao, C.; Zhang, X.** (2018): A secure ciphertext retrieval scheme against insider KGAs for mobile devices in cloud storage. *Security and Communication Networks*, vol. 2018.

**Xiong, L.; Shi, Y.** (2018): On the privacy-preserving outsourcing scheme of reversible data hiding over encrypted image data in cloud computing. *Computers, Materials & Continua*, vol. 55, no. 3, pp. 523-539.

**Yang, J.; He, S.; Lin, Y.; Lv, Z.** (2017): Multimedia cloud transmission and storage system based on internet of things. *Multimedia Tools and Applications*, vol. 76, no. 17, pp. 17735-17750.

**Zhou, S.; Liang, W.; Li, J.; Kim, K.** (2018): Improved VGG model for road traffic sign recognition. *Computers, Materials & Continua*, vol. 57, no. 1, pp. 11-24.