Research Article

Automatic Identification and Location of Tunnel Lining Cracks

Pengyu Wang, Shuhong Wang, and Alipujiang Jierula

School of Resources and Civil Engineering, Northeastern University, Shenyang 110819, China

Correspondence should be addressed to Shuhong Wang; shwangneu@126.com and Alipujiang Jierula; alipu135@126.com

Received 21 August 2020; Revised 27 December 2020; Accepted 8 March 2021; Published 26 March 2021

Academic Editor: Zhijie Wen

Copyright © 2021 Pengyu Wang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The lining crack is common for the tunnel in the stage of operation, which has seriously influenced the service life and safety of tunnel engineering. It is a new trend to use computer vision to detect tunnel cracks over the past few years in China and foreign countries. By image processing technology and intelligent algorithm, the computer has a hominine visual perception system which understands, analyzes, and determines input image information, thus recognizing and detecting specific objectives. However, the effect of image recognition for tunnel crack now cannot satisfy the demands of practical engineering. SSD algorithm has been used when analyzing features of lining surface image, while comparison analysis has been made from image recognition results, error rate, and running time. The results indicate that the SSD algorithm can accurately and rapidly detect and mark the position of the tunnel crack. The tunnel information obtained from image recognition is subsequently imported into the team independently developed software GeoSMA-3D, which is useful for determining tunnel grade.

1. Introduction

Since the 21st century, China has seen the rapid development of underground engineering, especially tunneling construction. The water leakage, lining crack, and differential settlement are common for the tunnel in the stage of operation and maintenance, where lining crack is the most common structure damage [1]. The lining crack will break the stability of the tunnel, and continuous deterioration may result in instability of tunnel structure [2]. In fact, it is a process to see the change and deterioration of tunnel damage. Meanwhile, early detection and timely implementation of regulation measures can completely ensure tunnel safety in the stage of operation. Therefore, it is extremely necessary to monitor tunnel damage.

The tunnel structure damage is generally detected and recorded through traditional manual routing inspection which mainly depends on the level and subjective judgement of inspectors and therefore cannot completely and easily obtain accurate and detailed information [3]. Besides, a large number of labors and materials are required due to the low efficiency of detection (see Figure 1). With the change of demands for engineering detection, the method gradually develops from early inspection by naked eyes to advanced technologies like NDT and physical detection, while many evaluation methods and models have also been proposed. At the beginning of the 21st century, the photographic car was developed for scanning surface damage on tunnel lining, and the laser scanning method was used for completely scanning surface conditions of lining concrete structure. The image of lining crack can be generated through recognition and analysis [4, 5]. Literature works [6] have evaluated and studied the application of fiber optical sensors in practical engineering and their influences on tunnel safety. Besides, literature studies [7] have also developed a tunnel damage analysis system which can predict the status of the development of lining crack based on changes in lining strength.

Over the past few years, the computer visual inspection technology for collecting and recognizing the surface image of the tunnel structure has sprung up [8]. However, such inspection has brought a lot of image data about the surface of the tunnel structure. According to existing research, 200 GB inspection data may be generated for a tunnel with a length of 1 km. Therefore, it is urgent to develop new technologies for rapidly recognizing and accurately extracting geometrical information about crack damage. With the coming big data era, computer’s computing power and training data size have been improved dramatically,
while complex models representative of deep learning start to attract people’s attention. Due to excellent generalization and robustness, deep learning now has been applied to computer visual detection for civil engineering in China and foreign countries [9, 10].

Target recognition technology originated in the 1940s. Until the 1990s, artificial neural network technology boosted the rapid development of this technology. At present, target recognition technology has been widely used in license plate recognition, face recognition, and object detection. Early traditional target recognition technology is mainly based on shallow-level models, and there is a need to artificially preprocess a mass of images, so the researchers have begun to study deeper network models and extract image features using the model. A variety of deep learning models have been proposed, including DBN [11], CNN [12], RCNN [13], FCN [14], and YOLO [15].

The DBN network model has greatly contributed to the establishment of multiple hidden layers of artificial neural network. The DBN model has been successfully applied in many fields, but still in the early stage of development [16]. However, it lacks an effective parallel training method, which produces high computational costs, and low-efficiency computation hinders the wide application of DBN. The CNN model has been well applied in the field of small image classification, but with a low recognition efficiency of large-scale data. Until 2012, Krizhevsky et al. achieved good results relying on deeper CNN in large-scale visual recognition challenge (LSVRC). RCNN can raise the accuracy of target detection to a new level, consisting of 4 independent steps: candidate window generation, feature extraction, SVM classification, and window regression, resulting in low detection efficiency. Therefore, many scholars obtained the improved Fast-RCNN, through which the images were sent to deep network for once, instead of sending all the candidate windows to the network, which greatly improved the detection speed. As one of the best object detection algorithms, YOLO’s main advantage lies in detection efficiency. As a part of CNN, the difference among feature extraction, SVM classification, and window regression is blurred for direct and quick detection.

For images of tunnel lining diseases, Smith and Brady [17, 18] first extracted 17 low-level features from original images and then input these low-level features into the convolutional neural network to obtain high-level features and finally input these high-level features into a multilayer perceptron to achieve the identification of tunnel diseases. Sun et al. [19] compared and analyzed the image classification of pavement cracks through three methods, including deep convolutional network, support vector machine, and ensemble learning. It was found that the deep convolutional network had better detection effects than the other two methods. Huang et al. [20] used the deep convolutional network to study the identification of concrete cracks and detected images of any size with the combination of the sliding window method.

However, the above method has shown many disadvantages, for example, single size of extracted feature graph, bad detection effect for small target crack, long time for data processing, and large calculation space. For this reason, this paper has used a deep learning-based SSD method to recognize the tunnel crack rapidly and accurately. There are three steps: first, establish a sample dataset; second, build a network structure; third, evaluate the training grid and results.

2. Acquisition of Image Data

As an important subdiscipline of computer science, artificial intelligence aims to implement such technologies as natural language understanding, image recognition, and speech recognition through computer. Machine learning is a part of artificial intelligence and is a method of supporting machine learning and providing functions unrealizable through direct programming; in practice, machine learning is a method of collecting data, training the models based on data, and finally making predictions using the models. Deep learning is a subdiscipline of machine learning, and its core is to automatically
combine simple features into more complex features and solve problems based on these features.

Deep learning is to train the model by training data, continuously collect a large amount of information in data into the model by machine learning algorithm, and make specific processing for real-world similar data by the trained model, for example, image classification and target positioning. The image classification is to distinguish images with and without crack and only determine an image with or without crack except the detailed position of the crack. The target positioning is to determine the crack position with the frame in the tunnel image and then obtain the detailed position of the crack. Therefore, recognition and positioning for crack are required in this paper.

2.1. Acquisition of Tunnel Image. The construction of an image set in a variety of ways is the first step and a basic part of image recognition based on deep learning, including original photographing methods, collection of massive data based on modern crawler technology and big data concept, and direct use of the existing open-source database. Regardless of which method is adopted, its ultimate goal is to collect desired and sufficient image samples containing feature objects, with diverse backgrounds.

The image sample dataset is obtained through image acquisition for lining surfaces of 12 tunnels in Shenyang by fast mobile detection equipment for tunnel structure [21, 22]. The tunnel crack is shown in Figure 2, where the total length is 7358 m for five long tunnels, 3080 m for four medium tunnels, and 1138 m for three short tunnels. The number of cracks in long tunnels is 139, with a total length of 1169 m, including 93 circular cracks and 46 noncircular cracks, 27 longitudinal cracks among noncircular cracks. The number of cracks in medium tunnels is 211, with a total length of 2009 m, including 142 circular cracks with a total length of 1702 m and 69 noncircular cracks with a total length of 306 m, 45 longitudinal cracks among noncircular cracks with a total length of 267 m. The number of cracks in short tunnels is 39, with a total length of 1065 m, including one circular crack and 38 noncircular cracks (completely longitudinal cracks).

The image set for this research is independently shot and is quickly extended without damages by initially converting the images (including rotation, mirror, and scale) using general image processing technology. It can be deemed that original images and images extended through initial conversion have the same target features for deep learning training instead of the same images. Therefore, when original images are insufficient, initial conversion becomes a common method of quickly extending the image set.

A large number of sample images marked with features are required for image recognition based on deep learning to form the training set, of which the basic principle is to conduct deep learning training with images marked with target features, find and extract features of marketed target, and then implement recognition and detection based on training results after recording. In the image sample dataset, each original image corresponds to a ground truth which is a datum reference obtained through marking crack region in the image manually.

The labeling of original images is quite significant in target detections. It can label the locations of the target objects in original images and generate corresponding xml files for each image to represent the location of the target standard box. At present, the labeling tool used is simple and easy, but it cannot label multiple targets of the same category in the same image; besides, it can only generate a corresponding txt file after the labeling is completed, which needs to be converted into the corresponding xml file through certain tools. This paper used the LabelImg tool, which can label multiple categories and directly generate xml files easily.

The label category can be modified in file “pre-defined_classes.txt” under folder “data” of the menu directory. When loading image with shortcut key “ctrl + u,” you can designate a place for generated “xml” file with shortcut key “ctrl + r,” and then mark targets in image by category with rectangular box. The software will automatically pop up category information after you mark a target, and then you can double click the corresponding name in pop-up information. When all required targets for each category are marked in an image, you can click the button or use the shortcut key “ctrl + s” to save and generate position information of the corresponding “xml” file. After that, you can mark the next image. The crack mark is shown in Figure 3.

The prediction process is relatively simple. For each prediction box, you can firstly determine its category (with maximum confidence coefficient) and confidence coefficient by category confidence and filter prediction box in the background and then filter prediction box with a minimum threshold by the threshold of confidence coefficient (e.g., 0.5). The remaining prediction boxes are decoded, and then, its real position parameters are obtained upon prior box. After decoding, these parameters are generally ranked in descending order by confidence coefficient, and only top-k (e.g., 400) prediction boxes remain. Finally, NMS algorithm is used to filter prediction boxes with a high degree of overlap. The remaining boxes are final detection results.

3. Detection of Cracks

3.1. Development Environment. The real-time detection network developed based on computer vision for tunnel lining cracks is suitable for Windows operating platform and is developed based on 64-bit Windows 10 Professional operating platform. In terms of hardware, a computer is provided with Intel i7-9700F CPU (a key computer component), NVIDIA RTX2070 SUPER graphics card, and
3.2. Crack Detection Algorithm. The main idea of the SSD algorithm is to evenly conduct intensive sampling at different locations in images. Different scales and aspect ratios can be used when sampling. Then, the CNN is used to extract features and directly perform the classification and regression. The entire process can be finished in one step. The advantage of one-stage methods is fast speed. The SSD algorithm refers to Single Shot MultiBox Detector. Single shot indicates that the SSD algorithm belongs to one-stage methods, and MultiBox indicates that the SSD algorithm uses multibox predictions [24]. The basic structure of the SSD is shown in Figure 4. The SSD uses the CNN network to extract detection results from different feature maps directly, which does not need detection after the fully connected layers. The detection can be completed in one step.

After a growing number of deep learning models have been proposed and improved, Google, Microsoft, Facebook, and other companies have developed a series of deep learning frameworks, including Google’s TensorFlow framework, Facebook’s Torch framework, Microsoft’s CNTK framework, Fchollet’s Keras framework, DMLC’s MXNet framework, and Caffe framework jointly developed by Berkeley Vision and Learning Center and community contributors. These deep learning frameworks are mainly used in image feature recognition, image classification, speech recognition, and natural language processing. With the emergence and application of the above frameworks, deep learning technology has been rapidly developed.

Based on existing image training samples, target detection algorithms for deep learning were studied, compared, and screened, including comparison of several classic target recognition algorithms based on the convolutional neural network: RCNN, YOLO, and SSD, among which SSD was finally used.

The SSD algorithm uses feature maps of multiple scales for detection, and there are a large feature map and a small feature map, both used for detections. The advantage is that large-scale feature maps (front feature maps) can be used to detect small objects and small-scale feature maps (back feature maps) can be used to detect large objects. As shown in Figure 5(a), $8 \times 8$ feature map can be divided into more units while the priority boxes of each unit are relatively small.

Each unit in the SSD algorithm has prior boxes with different scales or aspect ratios. The predicted bounding boxes are set based on these prior boxes to reduce the training difficulty. Normally, each unit will have multiple prior boxes with different scales and aspect ratios. For each prior box of each unit, there is a set of independent detection values, which can be divided into two parts. The first part is the confidence or score of each category. The SSD algorithm also regards the background as a special category. If the detection target has $c$ categories, the SSD algorithm needs to predict $c+1$ confidence values and the first confidence is the score without targets or the score belonging to the background. In the prediction process, the category with the highest confidence is the category to which the bounding box belongs. Particularly, when the first confidence value is the highest, it indicates that the bounding box does not contain the target. The second part is the location of the bounding box, containing 4 values ($cx, cy, w, h$), representing the center coordinates, the width, and the height of the bounding box. However, the actual predicted value is the conversion value of the bounding box compared with the prior box. The location of the prior box is represented by $d = (d^{cx}, d^{cy}, d^{w}, d^{h})$ and its corresponding bounding box is represented by $b = (b^{cx}, b^{cy}, b^{w}, b^{h})$. Then, the predicted value $l$ of the bounding box is the conversion value of $b$ compared with $d$. 

Figure 2: Diagram of the tunnel crack damage.

Figure 3: Marking cracks.
The process above is the encoding of the bounding box, which should be reversed during the prediction, that is, decoding. The true location $b$ of the bounding box can be obtained from the predicted value $l$:

$$
\begin{align*}
    f^x &= (b^x - d^x)/d^x, \\
    f^y &= (b^y - d^y)/d^y, \\
    l^w &= \log(b^w/d^w), \\
    l^h &= \log(b^h/d^h).
\end{align*}
$$

(1)

The process above is the encoding of the bounding box, which should be reversed during the prediction, that is, decoding. The true location $b$ of the bounding box can be obtained from the predicted value $l$:

$$
\begin{align*}
    b^x &= d^x f^x + d^x, \\
    b^y &= d^y f^y + d^y, \\
    b^w &= d^w \exp(l^w), \\
    b^h &= d^h \exp(l^h).
\end{align*}
$$

(2)

In summary, for a $m \times n$ feature map, there are $mn$ units in total. The number of prior boxes set for each unit is recorded as $k$. Then, each unit requires $(c+4)k$ predicted values, and all units require $(c+4)kmn$ predicted values. Because the SSD uses convolutions for detections, a total of $(c+4)k$ convolution kernels were used to complete the detection of this feature map. The SSD algorithm uses prior boxes (default boxes) of different sizes and aspect ratios to overcome the disadvantages of difficult detections of small targets and inaccurate positioning, so this paper used the SSD algorithm to detect cracks [25].

3.3. Network Structure. The SSD algorithm uses VGG16 as the basic model and then adds convolutional layers based on VGG16 to obtain more feature maps for detections. The network structure of the SSD is shown in Figure 6. It can be clearly seen that the SSD uses multiscale feature maps for detections. The input image size of the model is $300 \times 300$.

In this paper, tunnel lining cracks were taken as the feature recognition target, and collected and constructed database of 20,000 pictures was used as a dataset for deep learning training. In the initial stage, it took about 6 hours to keep the loss value stable at about 1 in training this model under the above workstation configuration after deep learning training and recognition test. The verification of all the recognition and detection effects in this paper will not be further explained and is based on the above image set and its training results [25].

$300 \times 300$ images were input into SSD 300, and VGG16 convolutional layer was used to extract features. Two fully connected layers of VGG16 were converted into ordinary convolutional layers (conv6 and conv7 in the figure), and then multiple convolutional layers were connected for $1 \times 1$ output using a global average pool. It can be seen from the figure that the SSD connects conv4_3, conv7, conv8_2, conv9_2, conv10_2, and conv11_2 to the final detection/classification layer for regression.

Basic steps of SSD network prediction are as follows: input a picture ($300 \times 300$) into the pretrained classification network (modified traditional VGG16 network) to obtain feature maps of different sizes; extract feature maps of Conv4_3, Conv7, Conv8_2, Conv9_2, Conv10_2, and Conv11_2 layers, and then construct 6 default boxes of different sizes at each point of these feature map layers. Separately detect and classify the feature maps, and generate multiple default boxes that meet preliminary requirements; combine default boxes obtained from different feature maps, filter some overlapped or incorrect default boxes using NMS, and generate a final set of default boxes (i.e.,
these parameters, the training process can be adjusted to
(label_map.pbtxt), and other parameters. By adjusting
file (checkpoint), directory of identification label file
number of training steps (num_steps), path of checkpoint
(batch_size), learning rate (learning_rate), maximum
(num_classes), batch size for each training step
mainly including the number of detection target classes
containing training feature information for SSD model,
the config file is used as a model configuration file
role in deep learning. Under the TensorFlow framework,
training again in the original training directory.
deep learning training after interruption, you can start
training steps increases. If there is a need to continue
number of steps will increase as the total number of
are automatically saved every several steps, and the
training results are the data files of the TensorBoard
starting with events.out.tfevents in the folder for saving
results are the data files of the TensorBoard
visualization tool and can be called by TensorBoard for
training, during which the parameters such as loss value
affect the efficiency of deep learning and the accuracy of
final recognition.
During the training process, it is necessary to determine
the ground truth (real target) and the prior box used for
matching in the training image first. The bounding box
corresponding to the matched prior box will be responsible
for predicting it. There are two main principles for matching
the prior boxes of SSD with the ground truth. First, for each
ground truth in the image, the prior box with the largest IOU
should be found, which matches with the ground truth. In
this way, it can be ensured that each ground truth must
match with a certain prior box. Usually, the prior box that
matches with the ground truth is called a positive sample.
Conversely, if a prior box does not match with any ground
truth and can only match with the background, then it is
called a negative sample. There are few ground truths in an
image, but there are many prior boxes. If only the first
principle is followed, many prior boxes will be negative
samples, resulting in the extreme imbalance between posi-
tive and negative samples. Therefore, the second principle is
necessary. The second principle is that, for the remaining
unmatched prior boxes, if the IOU of a ground truth is
greater than a certain threshold (usually 0.5), then the prior
box also matches with this ground truth, indicating that a
certain ground truth may match with multiple prior boxes.
After the training samples are determined, then the loss
function is determined. The loss function is defined as the
weighted sum of the localization loss and the confidence
error loss:
\[
L(x, c, l, g) = \frac{1}{N} \left( L_{\text{conf}}(x, c) + a L_{\text{loc}}(x, l, g) \right),
\]
where \(N\) refers to the number of positive samples of the prior
boxes. Here \(x^p_i \in \{1, 0\}\) is an indicating parameter; when
\(x^p_i = 1\), it indicates that the \(i\)-th prior box matches with the \(j\)-th
ground truth that belongs to the \(p\) category. \(c\) is the
predicted value of the category confidence. \(l\) is the location
prediction value of the bounding box corresponding to the
prior box, and \(g\) is the location parameter of the ground
truth.
A variety of data enhancement methods are used for SSD
algorithm, including horizontal flipping, cropping, zooming
in, and zooming out. Data enhancement can significantly
improve the algorithm performance for the main purpose of
making the algorithm more robust to targets of different
sizes and shapes. It can be intuitively understood that the
number of training samples can be increased through data
enhancement, and more targets of different shapes and sizes
can be built. By inputting into the network, network learning
is more robust. The loss value of the deep learning model in
the training process with the number of iterations is shown in
Figure 7.

3.4. Training Process. The training methods of the Ten-
sorFlow framework include CPU-based training and
GPU-based training. GPU training is well adapted to
NVIDIA graphics cards. Despite the compatibility with
AMD graphics cards, the configuration is more cum-
bersome, and graphics cards cannot be fully used. In
order to improve training efficiency, a mass of data is
obtained and debugged, and GPU-based TensorFlow
is used for training. Direct files required for training in-
clude model configuration files, existing transfer learning
models, record files, and label files, and training can be
started after full preparation and configuration. After the
training starts, there are two ways to stop training. The
first is to reach the maximum number of training steps set
in the model configuration file, and the second is to
directly stop the training program. The training results
are automatically saved every several steps, and the
number of steps will increase as the total number of
training steps increases. If there is a need to continue
deep learning training after interruption, you can start
training again in the original training directory.
Parameter configuration of the model file plays a vital
role in deep learning. Under the TensorFlow framework,
the config file is used as a model configuration file
containing training feature information for SSD model,
mainly including the number of detection target classes
(num_classes), batch size for each training step
(batch_size), learning rate (learning_rate), maximum
number of training steps (num_steps), path of checkpoint
(file (checkpoint)), directory of identification label file
(label_map.pbtxt), and other parameters. By adjusting
these parameters, the training process can be adjusted to

3.5. Training Effect. After deep learning training, the files
starting with events.out.tfevents in the folder for saving
training results are the data files of the TensorBoard
visualization tool and can be called by TensorBoard for
training, during which the parameters such as loss value
and learning rate vary with actual changes in the number of training steps. During this research, real-time detection of tunnel lining cracks based on deep learning and computer vision can support static image recognition, video file recognition, and real-time camera-based recognition and detection.

Rock images containing cracks were input. The detection results are shown in Figure 8. The SSD algorithm can accurately identify the location of cracks.

The comparison results of the SSD algorithm and other detection algorithms are shown in Table 1. Basically, it can be found that SSD has an accuracy the same as Faster R-CNN and a detection speed the same as YOLO.

Tunnel patrol robots are networked for real-time detection and statistics of tunnel lining cracks in the future. The tunnel patrol robot is shown in Figure 9.
4. Conclusion

(1) The deep learning-based SSD algorithm has been used to extract advanced features of tunnel crack image, which can effectively avoid noise effect. Besides, recognition algorithm has been established for tunnel crack image.

(2) Compared to the traditional algorithm, SSD algorithm has made improvements in multiscale feature graph, detection by convolution, and setting of prior boxes. As a result, SSD algorithm has achieved higher accuracy and better detection effect for small targets.

(3) From the view of recognition for a crack in the image, the rate of accuracy in this paper is approximated to 90%, which is far higher than other algorithms. With respect to running time, the algorithm in this paper takes a shorter time, being 0.2 s on average. Therefore, methods in this paper can recognize tunnel crack image rapidly and accurately to satisfy the demands of practical engineering.

(4) The size of the sample database determines the accuracy of deep learning training and whether it can easily overfit during the training process. This contribution attempts to establish a sample library of tunnel image features, but the current number of images is far from ideal. The next step will be to collect and collate new tunnel samples, increasing the types of network model tunnel damage identification, so that the network is not limited to the tunnel crack damage.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

The authors would like to thank Northeastern University for providing the access to the software and other facilities. This work was conducted with supports from the National Natural Science Foundation of China (Grant nos. 51474050 and U1602232), the Fundamental Research Funds for the
Central Universities (Grant no. 17010829), Doctoral Scientific Research Foundation of Liaoning Province (Grant nos. 2017054034 and 20170520341), State Key Laboratory of Silicate Materials for Architectures (Wuhan University of Technology) (Grant no. SYSJJ2017-08), and the Research and Development Project of China Construction Stock Technology (Grant no. CSCEC-2016-Z-20-8) to Dr. Shuhong Wang.

References

[1] T. Asakura and Y. Kojima, "Tunnel maintenance in Japan," Tunnelling and Underground Space Technology, vol. 18, no. 2-3, pp. 161–169, 2003.

[2] K. Boonsivanon and A. Meesomboon, "IKDSIFT: an improved keypoint detection algorithm based on SIFT approach for non-uniform illumination," Procedia Computer Science, vol. 86, no. 2, pp. 269–272, 2016.

[3] S. Bianco, M. Buzzelli, R. Mazzini, and R. Schettini, "Deep learning for logo recognition," Neurocomputing, vol. 245, no. 1, pp. 23–30, 2017.

[4] Y. Yuan, X. Jiang, and X. Liu, "Predictive maintenance of shield tunnels," Tunnelling and Underground Space Technology, vol. 38, pp. 69–86, 2013.

[5] S. Y. Lee, S. H. Lee, D. I. Shin et al., "Development of an inspection system for cracks in a concrete tunnel lining," Canadian Journal of Civil Engineering, vol. 34, no. 34, pp. 966–975, 2007.

[6] R. Montero, J. G. Victores, S. Martínez, A. Jardón, and C. Balaguer, "Past, present and future of robotic tunnel inspection," Automation in Construction, vol. 59, pp. 99–112, 2015.

[7] W. Zhang, Z. Zhang, D. Qi et al., "Automatic crack detection and classification method for subway tunnel safety monitoring," Sensors, vol. 14, no. 10, pp. 307–319, 2014.

[8] M. Gong, H. Yang, and P. Zhang, "Feature learning and change feature classification based on deep learning for ternary change detection in SAR images," ISPRS Journal of Photogrammetry and Remote Sensing, vol. 129, no. 2, pp. 212–225, 2017.

[9] T. Kobayashi and M. H. BFO, "Feature extraction based on histograms of oriented pdf," Gradients for Image Classification, vol. 91, no. 9, pp. 747–754, 2013.

[10] S. Y. Lee, S. H. Lee, D. I. Shin et al., "Development of an inspection system for cracks in a concrete tunnel lining," Canadian Journal of Civil Engineering, vol. 34, no. 34, pp. 966–975, 2007.

[11] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," Science, vol. 313, no. 5786, pp. 504–507, 2006.

[12] Y. L. Cun, L. D. Jackel, B. Boser et al., "Handwritten digit recognition: applications of neural network chips and automatic learning," IEEE Communications Magazine, vol. 27, no. 11, pp. 41–46, 1989.

[13] R. Girshick, J. Donahue, T. Darrell et al., "Rich feature hierarchies for accurate object detection and semantic segmentation," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 580–587, Columbus, OH, USA, June 2014.

[14] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3431–3440, Boston, MA, USA, June 2015.

[15] Z. Zhang, "A flexible new technique for camera calibration," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 22, no. 11, pp. 1330–1334, 2000.

[16] L. Deng, M. L. Seltzer, D. Yu et al., "Binary coding of speech spectrograms using a deep autoencoder," in Proceedings of the 11th Annual Conference of the International Speech, pp. 1692–1695, Chiba, Japan, September 2010.

[17] S. M. Smith and J. M. Brady, "SUSAN—a new approach to low level image processing," International Journal of Computer Vision, vol. 41, no. 10, pp. 41–52, 2001.

[18] Q. Ai, Y. Yuan, and X. Bi, "Acquiring sectional profile of metro tunnels using charge-coupled device cameras," Structure and Infrastructure Engineering, vol. 12, no. 9, pp. 1065–1075, 2015.

[19] Y. Sun, Y. D. Xue, and H. W. Huang, "Inspection equipment study on subway tunnel defects," in Proceedings of the 2016 World Tunnel Congress, pp. 1–9, International Tunneling and Underground Space Association, San Francisco, CA, USA, April 2016.

[20] H. Huang, Y. Sun, Y. Xue, and F. Wang, "Inspection equipment study for subway tunnel defects by grey-scale image processing," Advanced Engineering Informatics, vol. 32, pp. 188–201, 2017.

[21] Y. Lecun, Y. Bengio, and G. Hinton, "Deep learning," Nature, vol. 521, no. 7553, pp. 436–444, 2015.

[22] E. Protopapadakis and N. Doulamis, "Image based approaches for tunnels defects recognition via robotic inspectors," in Proceedings of the 11th International Symposium on Visual Computing, pp. 706–716, Springer International Publishing, Las Vegas, NV, USA, 2015.

[23] L. Zhang, F. Yang, Y. D. Zhang et al., "Road crack detection using deep Convolutional neural network," in Proceedings of the 23rd International Conference on Image Processing, pp. 3708–3712, IEEE, Phoenix, AZ, USA, 2016.

[24] Y.-I. Cha, W. Choi, and O. Büyükoztürk, "Deep learning-based crack damage detection using convolutional neural networks," Computer-Aided Civil and Infrastructure Engineering, vol. 32, no. 5, pp. 361–378, 2017.

[25] W. Liu, D. Anguelov, D. Erhan et al., "SSD: single shot multibox detector," 2016, https://arxiv.org/abs/1512.02325.