Received December 4, 2019, accepted December 20, 2019, date of publication December 26, 2019, date of current version January 6, 2020.

Digital Object Identifier 10.1109/ACCESS.2019.2962496

Intelligent Detection for Tunnel Shotcrete Spray Using Deep Learning and LiDAR

LUO CHUN-LEI, SHA HAO, (Student Member, IEEE),
LING CHUN-LAI, (Student Member, IEEE), AND LI JIN-YANG, (Student Member, IEEE)
College of Mechanical and Electrical Engineering, Central South University, Changsha 410000, China

Corresponding author: Sha Hao (csuelric95@gmail.com)

This work was supported by the Gengli Engineering Machinery Equipment co., Ltd., Henan.

ABSTRACT Shotcrete spray is an indispensable process in tunnel construction. At present, the construction of tunnels in China is mainly depend on labor or mobile concrete sprayer, which has lots problems like time-consuming, low precision, and labor intensive. An intelligent detection method for tunnel shotcrete spraying is proposed in this article. There are two main issues need to be solved, one is the modeling of tunnel in real-time to monitor the thickness of shotcrete and other is the detection of spraying area in the tunnel. The LiDAR can obtain a 3D model of tunnel after performing necessary preprocess on it in real-time. On the other hand, the spraying areas are usually divided by arches in the tunnel, so we can detect the position of arches to determine the spraying areas. Inspired by the YOLO algorithm, we proposed a novel neural network structure to detect the approximate bounding boxes of the arches and a line-detection algorithm is used to determine the final positions of the spraying area in the image. The size of the weight file of our neural network is only 2.57 MB after the use of some deep compression tricks, which means our model is device friendly. After that, the object detection results in the image will be projected to the point cloud data. The experimental results suggest that our method performed well in the detection for tunnel shotcrete spraying, and the mAP for spraying area detection was found to be 91.4%.

INDEX TERMS 3D point cloud, deep learning, object detection, tunnel shotcrete.

I. INTRODUCTION

Shotcrete support is commonly used in the construction of railways, highway tunnels, mine roadways, subways, and various underground buildings [1]. Tunnel shotcrete spraying is a process of spraying shotcrete at a very high speed onto a rock or concrete surface to prevent the collapse of the tunnel during excavation [2], [3]. To address the shortcomings of manual spraying such as a high rebound rate, large amount of dust, unstable construction quality, and slow construction progress [4], mobile concrete sprayer emerged. However, this method still requires an operator to manually operate the robot arm for spraying, which means that the quality of the shotcrete from this method depends largely on the experience of the operator. In order to liberate humans from heavy and dangerous work, the intelligent detection for tunnel shotcrete spray is necessary for the automated construction of tunnels. In analyzing the process of tunnel shotcrete spraying, the main tasks are the detection of tunnel profile and spraying area.

For the tunnel profile detection, some researchers have [5]–[7] used the LiDAR or RGB-D camera to build the 3D model of tunnel. In order to ensure the accuracy of 3D model of tunnel and make it sense in the subsequent process, the resolution of sensor is required to be higher, which means high costs. Our method does not process the point cloud data directly, so the performances of sensor such as the scanning angle and the density of data are not relatively important. Correspondingly, due to the sparsity of the point cloud, the data registration is necessary for surface reconstruction. The Iterative Closest Point (ICP) algorithm has been widely used in point cloud registration.

For the detection of the spraying area, the area is divided by two steel arches, and the arches are connected by steel bars. We can detect the spray area by extract the steel bar in the tunnel. However, there a few studies have focused on the detection of arches in a tunnel. The study of rock drilling fields is a more popular research area in tunnel construction, and is not needed the detection of arches. References [8], [9] introduce a method to detect an arch by manual division and Mah *et al.* [10] proposed a Principal Component Analysis (PCA)-based method to detect the mesh in

The associate editor coordinating the review of this manuscript and approving it for publication was Gustavo Olague.

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see http://creativecommons.org/licenses/by/4.0/

point clouds. However, these methods are not sufficiently automated or precise. Since Krizhevsky [11] proposed the use of a convolutional neural network to extract the feature of image in 2012, the field of objection detection in image has experienced great development [12]–[15]. The speed and accuracy are the main evaluation of model performance in object detection, and some deep learning algorithms such as Faster Region-based Convolutional Neural Network (Faster-RCNN), You Only Look Once (YOLO), and Single Shot MultiBox Detector (SSD) worked well in it. However, these algorithms are mainly used for RGB images, and the 3-dimensional point cloud is not suitable for directly applying these algorithms due to extradimensional data. The object detection for 3D point cloud datasets has been a vast, growing research area in recent times [16], [17]. Before the emergence of the deep learning method, 3D point cloud-based object detection already had a relatively mature process: point cloud filtering, point cloud clustering, feature extraction and classification [18], [19]. A Directed Edge Growing (DEG) method proposed by Zhang *et al.* [7] can effectively extract the points of tunnel arches from LiDAR directly, but this method require the point cloud data with high-resolution which means high costs. In addition to the method that extracting the feature on point cloud according to the traditional workflow, many object detection methods in point cloud based on deep learning have been proposed in recent years. Such methods [20]–[24] use deep neural networks to extract features in an end-to-end manner, but they are difficult to balance the speed and accuracy of detection.

To achieve intelligent detection for tunnel shotcrete spraying, a deep learning and LiDAR-based method that can monitor the tunnel profile and determine the shotcrete area in real-time is developed. This new approach performs well in our experiments and has a high resistance to interference. The rest of this paper is organized as follows. Section 2 gives the related papers in regarding intelligent detection. Section 3 introduces the research methodology. Section 4 evaluates the effect of this approach in the construction of tunnels. Finally, Section 5 provides the summaries and conclusions of this research.

II. LITERATURE SEARCH

A. OBJECT DETECTION IN IMAGE

In the early stage of the development of the object detection algorithm, most people built complex models and more complex multimodel integration based on low-level features to slowly improve detection accuracy [25]. DPM is one of these traditional algorithms[26]. When a convolutional neural network achieved great success in the 2012 ImageNet classification task Girshick *et al.* [11] proposed the Regional Convolutional Neural Network object detection Framework (R-CNN) in 2014 [27]. Since then, the field of object detection has begun to develop at an unprecedented rate. In recent years, the development of object detection algorithms based on deep learning has been mainly divided into two types. One type of these algorithms is based on object proposals, such as Fast-RCNN, Faster-RCNN and FPN, and the other type of algorithms is not proposal-based but is based on integrated convolutional networks, such as YOLO, SSD, and RetinaNet [13]–[15], [28], [29]. The comparison of the mAP, frame per second with the evolution of different object detection algorithms is shown in **Fig. 1**. The development of object detection algorithms has shown prospects in both speed and accuracy. Consequently, object detection algorithms based on deep learning have been applied to various industries [30]–[32].

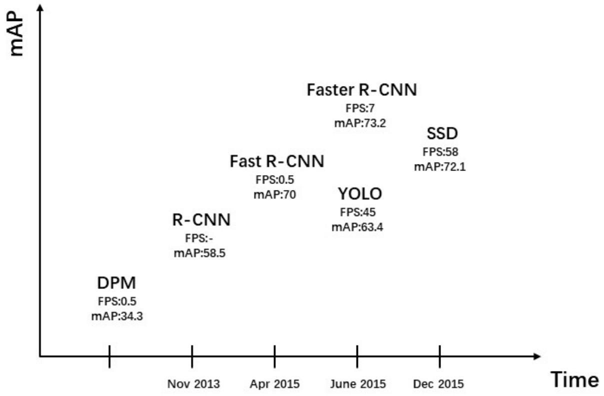


FIGURE 1. The comparison of different object algorithms.

B. OBJECT DETECTION IN A 3D POINT CLOUD

LiDAR-based 3D object detection is an inevitable requirement for the automation of tunnel shotcrete spraying because it is directly related to the understanding of the environment, thus laying the foundation for prediction and motion planning. Currently, there are three different deep learning methods in the field of 3D point cloud object detection [33], direct point cloud processing using multilayer perceptron, a method converting point clouds into voxels or image stacks by using CNN, and the combined fusion method. Some scholars have pushed the field forward. Engelcke *et al.* [34] proposed an approach to detecting objects natively in 3D point clouds using Vote3Deep net. Xiao *et al.* [35] proposed a novel hybrid CRF model to fuse the information from a camera and LiDAR for detecting a road. Caltagirone *et al.* [36] projected an unstructured and sparse point cloud to the camera image plane and then used fully convolutional neural networks to carry out road detection. Based on the literature survey, object detection in point cloud studies was reported less often.

C. INTELLIGENT DETECTION FOR TUNNEL SHOTCRETE SPRAY

In the process of tunnel excavation, the traditional methods mainly rely on the manual operation of the concrete sprayer for spraying. Studies on the automation of tunnel shotcrete spraying are relatively few in number. Some studies on automatically modeling and intelligently monitoring profile structures have been proposed [37], [38]. All of these

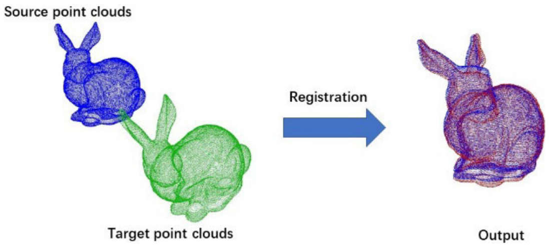


FIGURE 2. Registration of point cloud.

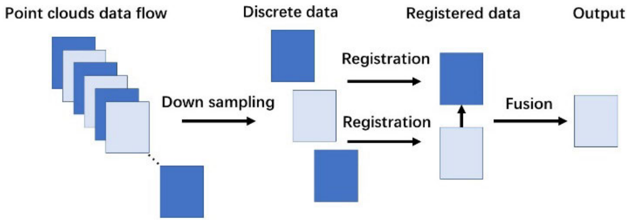


FIGURE 3. Surface reconstruction process based on continuous point cloud data.

studies use LiDAR to accomplish the goal of monitoring and have relatively better results. However, the use of only LiDAR is not sufficient to identify a shotcrete spraying area. Thus, we combined LiDAR with a camera to determine the shotcrete spraying areas by using deep learning and a novel line-detection algorithm.

III. RESEARCH METHODOLOGY

A. 3D MODELING OF A TUNNEL

Because of the limitations of LiDAR, it is difficult to complete the surface reconstruction of a tunnel directly through one frame of point cloud data. Before the 3D reconstruction of the tunnel, the mobile shotcrete sprayer would scan the whole tunnel by LiDAR. In this research, point cloud registration is used to expand the amount of data. The Iterative Closest Point (ICP) algorithm is a point set to point set registration method. As shown in **Fig. 2**, the source point clouds (blue) and the target point clouds (green) are two different point sets. The task of the ICP algorithm is to calculate the rotation and translation transformation matrix so that the target point clouds overlap with the source point clouds as much as possible.

Given two corresponding point sets $X = \{x_1, \ldots, x_n\}$, $P = \{p_1, \ldots, p_n\}$, registration minimizes the error between the transformed point clouds P' and the source point clouds X. In the ICP algorithm, registration uses least squares to represent the error. To minimize the **Equation (1)**, [39], [40] proposed a workflow to finish the registration of point cloud data.

$$E(R,t) = \frac{1}{N_p}\sum_{i=1}^{N_p} ||x_i - Rp_i - t||^2 \tag{1}$$

where x_i and p_i are corresponding points, R is the rotation matrix, t is the translation matrix, and N_p is the number of corresponding points. After the tunnel 3D scanning using LiDAR has finished, we propose a surface reconstruction process based on continuous point cloud data, as shown in **Fig. 3**. First, the continuous tunnel point cloud data collected by LiDAR in real time are down sampled at a certain time interval. Then the discretized point cloud data are paired and the ICP algorithm is used for registration. Then, we take the first registration data point as a reference and add the subsequently registered point cloud to it. Last, we combine all the registration point cloud data to obtain the final output.

A continuous contour model can be generated through surface reconstruction, which can express the tunnel conditions more intuitively. For numerical analysis and graphics, triangulation is an important preprocessing technique. References [40], [41] proposed a method that uses Delaunay triangulation to select a sample triangle as the initial surface, and continuously expands the boundary of the surface until all the points that conform to the geometric and topological conditions are connected, and finally form a complete triangle mesh surface. This algorithm is for fast triangulation of the original point cloud and has a relatively good surface reconstruction effect for point clouds with a smooth surface and uniform density [42].

B. DETECTION OF SPRAYING AREA

1) FUSION OF CAMERA AND LiDAR

Although some studie**s** use LiDAR for clustering to detect objects, the LiDAR we used is 16 lines, and its data are too sparse to cluster a distant object. Therefore, an RGB-camera is used for object detection, and then, the object is projected to the 3D point cloud to obtain the object position. To project the 2D bounding box coordinate information to the point cloud and obtain 3D bounding box coordinates of objects, the relationship between the two coordinate systems is required [43], [44]. The role of joint calibration is to fuse the camera (pixel) and LiDAR (point cloud) data. **Fig. 4** shows the relationship between the pixel coordinates and the world coordinates. The conversion relationship between the world coordinates and pixel coordinates can be expressed by **Equation (2)**:

$$\begin{aligned} s\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} &= \begin{bmatrix} \alpha_x & 0 & u_0 & 0 \\ 0 & \alpha_y & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} R & t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_w \\ y_w \\ z_w \\ 1 \end{bmatrix} \\ &= M_1 M_2 X_W = M X_W \end{aligned} \tag{2}$$

where $\alpha_x = 1/dx'$ and $a_y = f/dy'$, which are called the scale factors of the u and v axes, R is a $3*3$ rotation matrix, t is a $3*1$ translation matrix, $[x_w, y_w, z_w, 1]^T$ represents the coordinates under the world coordinates, M_1 is the intrinsic parameter matrix of the camera, M_2 is the extrinsic parameter matrix of the camera, and M is the projection matrix.

The intrinsic parameters of the camera should also include the five distortion parameters k_1, k_2, k_3, p_1, p_2 to eliminate the influence of distortion and the intrinsic parameter matrix

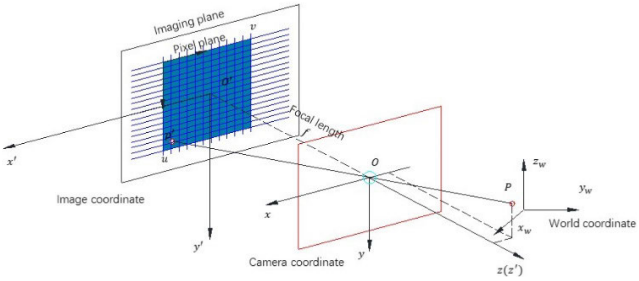


FIGURE 4. The relationships between the Pixel coordinate and the World coordinate.

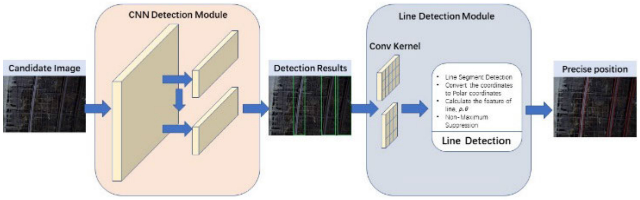


FIGURE 5. Flowchart for detecting spraying areas in the image.

M_1. The unknown variables for joint calibration include the 4 intrinsic parameters α_x, α_y, u_0, v_0, the extrinsic parameters $[R, t]$, and the 5 distortion parameters. We can determine the intrinsic parameters and extrinsic parameters of the camera by the corresponding position in the image and the three-dimensional coordinate system, and the corresponding position is identified by calibration targets such as a chessboard. The difference between our method and that of [45] is that the origin of the world coordinate system is fixed on the LiDAR rather than on corner of the chessboard.

2) DETECTION OF THE SPRAYING AREAS IN THE IMAGE

The detection of spraying areas in the image consists of: (1) a CNN detection module, which is used to detect the bounding boxes of arches and (2) a line detection module, which is used to extract the precise position of the arches. **Fig. 5** presents the workflow of the detection of spraying areas in the image.

YOLO [46] algorithm is a popular object detection algorithm. Different from other object detection algorithms, such as Faster-RCNN, it is a one-stage algorithm. The YOLO algorithm has been upgraded to three versions and greatly improved in speed and accuracy. In the YOLO-v3 [47] structure, there are some tricks such as multiscale prediction, darknet-53 backbone structure, and no pooling layer or full connection layer, to ensure the deeper layers and the higher accuracy of the network. However, there are 184 layers of darknet structure so that the parameters of the whole model can achieve 62.6 million, and the volume of the weight files is 239 MB, which means that it is difficult to apply the training results to a small controller. To reduce the redundancy, a simplified neural network is proposed to complete the object detection task, as shown in **Fig. 6**.

Inspired by YOLO-v3, this network uses successive 3 × 3 and 1 × 1 convolutional layers and has some residual units. The residual structure can accelerate the training speed of the network and make the network converge faster. Note that we reduce the depth of the darknet and the predicted dimensions. After the backbone network, darknet-23, the outputs of the network have 2 parts; each grid cell in the output feature map predicts B anchors and $N + 5$ parameters for those anchors, where B is the number of anchors and N is the numbers of classes. In addition to the position parameters of the anchors (x, y, w, h), there is also an additional predicted parameter C, which is the confidence of the anchors. The confidence comprehensively reflects the probability that the current bounding box contains the object $\Pr(object)$ and the accuracy of predicted bounding box IoU_{pred}^{truth}, as shown in **Equation(3)**. If the object is in the grid, $\Pr(object) = 1$.

$$Confidence = \Pr(object) * IoU_{pred}^{truth} \tag{3}$$

In addition, we also need to predict the probabilities of C classes for each grid cell. It represents the conditioned probability predicted by grid cell belongs to which category, which is $\Pr(class_i \,|\, object)$, $i = 1, 2, \ldots, C$. The class-specific confidence scores are shown as following:

$$\begin{aligned} Score &= \Pr(Class_i \,|\, object) * Confidence \\ &= \Pr(Class_i \,|\, object) * \Pr(object) \\ &\quad * IoU_{pred}^{truth} \end{aligned} \tag{4}$$

In addition to the simplification of the network structure, we discard the upsampling layers in YOLO-v3, because the object in tunnel is only an arch that is relatively fixed and large in the image, and the lower layers in the neural network contain more representational features. The lower layer features can be concatenated with a higher layer when the stride is set at two. After replacing the upsampling layers with down sampling DBL layers, the accuracy of our model improves.

The model size of our simplified structure is 57.8 MB, which is a dramatic reduction in model volume compared with that of the original YOLO-v3 model, and because some redundant layers are removed, the accuracy of our model is improved. To further reduce the parameter size of the model and make the model available in embedded systems, some deep compression methods are used in this paper. Han and Forrest N. Iandola *et al.* [48], [49] used pruning, trained quantization and weight sharing to compress some classical deep learning models many times. The pruning method is used to eliminate some unnecessary network parameters and retain those important to the network. Similar to [50], we evaluate the importance of filter channels in every convolutional layer using the average value of the normalized neuron weights for L1/L2 and the numbers of times when the output of the neurons is not 0. The flowchart is shown in **Fig. 7**.

In the pruning process, there is a ratio to control the degree of pruning of the model, and we can continually iterate the process of pruning to compress the model volume again. After comparing the iterative numbers, we find that the model can achieve a relatively better balance between the size and accuracy if the iterative number is two and the pruning ratio

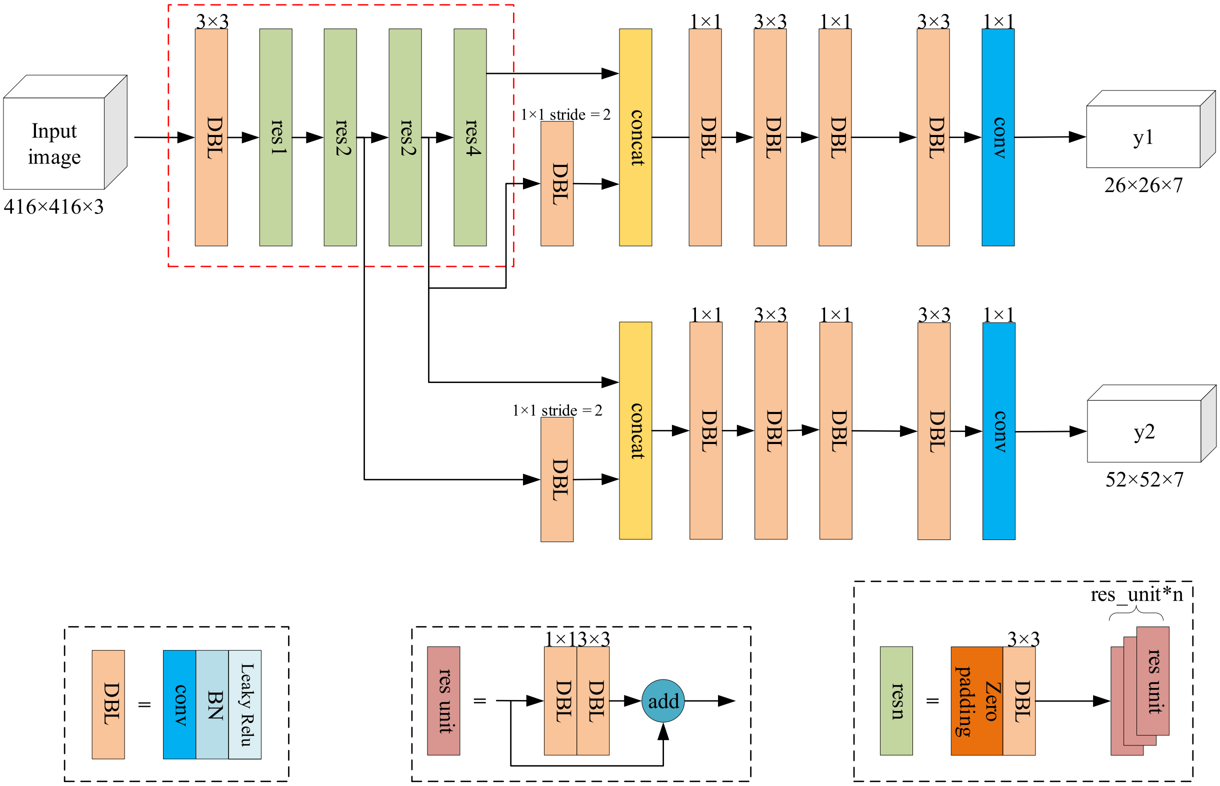


FIGURE 6. Overview of the proposed neural network.

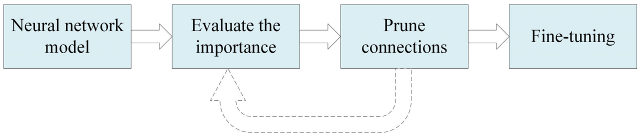


FIGURE 7. Flowchart for pruning.

is 0.9. After pruning, a fine-tuning process is used to recover the potential degradation in the performance, which is very important for pruning. In addition to the pruning method, other deep compression methods, such as trained quantization, transfer learning and low rank approximation are adopted to compress the model volume without affecting the accuracy. Based on the simplified neural network structure and these compression methods, we further present this new model with shallow structure and fewer trainable parameters than YOLO-v3. Moreover, due to the simplification of the parameters and model, our model can suppress overfitting to some extent.

Although the proposed neural network can detect the bounding boxes of the arch with high accuracy, there is still a problem with using the deep learning algorithm to detect the spraying area, and the results are affected by the shooting angle. As shown in **Fig. 8**, the areas of the bounding boxes of arches vary with the shooting angle. Object segmentation can be used, but this method is time-consuming and sensitive to the color in the image. In addition, the extraction of arches with image segmentation is irregular shape, which enhances the difficulty of the subsequent fusion with point cloud data. Considering the Line-Detection module can correct the bounding boxes position with less computes, it is used as the supplement of deep learning-based algorithm.

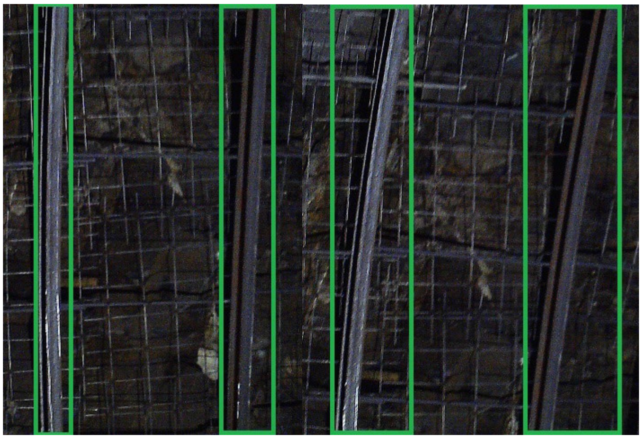
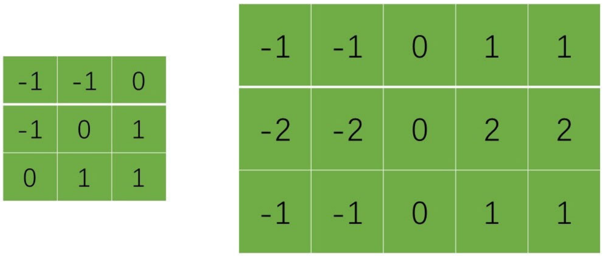


FIGURE 8. Different areas with different shooting angles.

The process of the line detection algorithm that we have adopted to extract the precise position of arches is presented as follows:

(i) Preprocess the image;

(ii) Conduct line segment detection;



-1	-1	0
-1	0	1
0	1	1

-1	-1	0	1	1
-2	-2	0	2	2
-1	-1	0	1	1

FIGURE 9. Two different types of convolutional kernel.

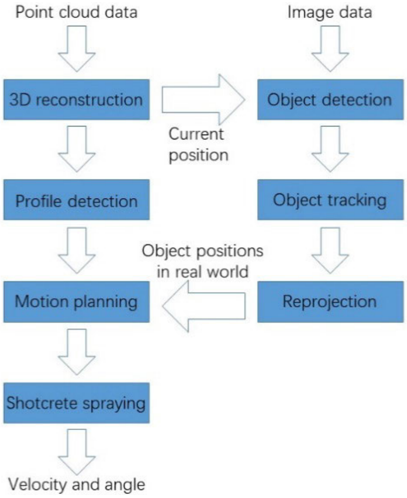


FIGURE 10. The flowchart of intelligent detection.

(iii) Convert the coordinates to polar coordinates and calculate the features of the line, ρ, θ; and

(iv) Conduct non-maximum suppression.

In 3.2.1, we have obtained the distorting parameters of camera by calibration, which can be used to offset the effect of camera distortion to some extent. During the construction of tunnel, there is always lack of sufficient light, and the hue (H) and Saturation (S) of image are not sensitive to light [51]. Therefore, in the line detection module, the bounding boxes of the arches predicted by the neural network algorithm, is converted to the HSV color model first. Since the arches are always vertical, two convolutional kernels are designed to extract the edge features of the image, as shown in **Fig. 9**.

The Line Segment Detector (LSD) algorithm can extract all the features of a line on a grayscale image in a short amount of time [52]. This algorithm calculates the angle between each pixel and the level-line to form a level-line field and merge the pixels in the neighboring regions with an approximate direction. These merged regions are called line support regions, which are the proposed line segments. When a line support region is particularly slender, it can be considered a straight-line segment. However, in the LSD algorithm, each pixel can only belong to one line support region, so if there is an intersection between two lines, at least one line is split into two parts. Moreover, the LSD algorithm is based on region growth, and long lines are often split into multiple lines because of occlusion, blur and so on.

TABLE 1. Object detection results using different algorithms.

Algorithm name	Backbone	mAP	Model Volume	FPS
YOLO	YOLOv3 tiny	55.2%	44.9 MB	64
	Darknet53	88.4%	239 MB	28
Faster-RCNN	VGG16	91.7%	349 MB	5
Complex-YOLO	Darknet19	73.7%	194 MB	13
Our proposed model	Darknet23	91.4%	57.8 MB	31
	Darknet23-0.9	89.3%	6.79 MB	32
	Darknet23-0.95	86.8%	2.57 MB	31
DPM	-	62.4%	-	0.8

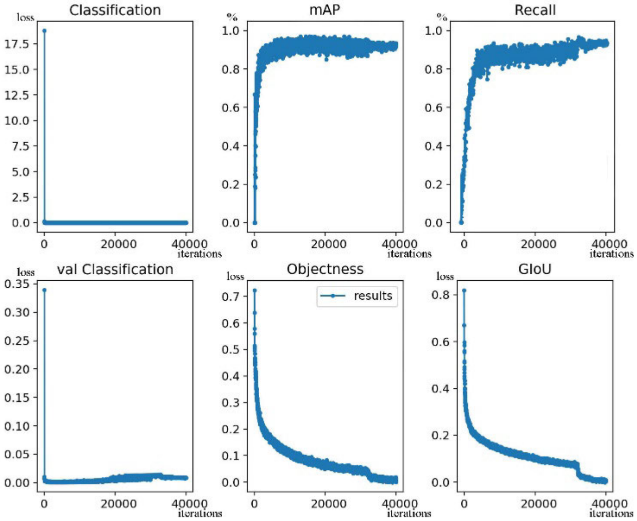


FIGURE 11. Training results for our proposed model.

Instead of describing a line segment by the end point and starting point coordinates, we can use ρ, θ to describe a line in the polar coordinate system, where ρ is the distance from the origin to the nearest point on the line and θ is the angle between the x-axis and ρ. Therefore, if the ρ and θ of the two lines detected by the LSD algorithm are very close, they are considered to be the same line and are merged.

The last step in line detection is Non-Maximum Suppression (NMS), and as the input is the bounding box of each arch, and each bounding box has only one arch, the output should be two lines. After NMS and line merging, there may still be multiple lines, and only 2 of the longest lines are used as the final output as the precise position of the arch.

3) CLUSTERING IN POINT CLOUDS

After obtaining the category and positions of objects, we can project this information to the point cloud, then the clustering algorithm is used to achieve the detection of the tunnel shotcrete spraying area [53]. The point cloud data are mainly a collection of a massive number of points that represent the surface of the object and do not contain the aggregate topology information of traditional entity grid data. The Kd-tree [54] is a data structure used in computer science to represent a set of points in k-dimensional space. This structure is a binary search tree with constraints. The Kd-tree is useful for interval and neighborhood searches [40]. We organize all the points using a binary tree. With the Kd-tree structure, the neighbors of a point can be searched only in their parent nodes and child nodes, which greatly reduces the calculation time for searching neighboring points. The process of Euclidean clustering based on the Kd-tree is as follows [55]:

(i) Create a Kd-tree representation for the input point cloud dataset P;

(ii) Set up an empty list of clusters C, and a queue of the points that need to be checked Q;

(iii) Then for every point $p_i \in P$, perform the following steps:

- Add p_i to the current queue Q;
- For every point $p_i \in Q$ do:
  - Search for the set P_k^i of point neighbors of p_i in a sphere with radius $r < d_{th}$
  - For every neighbor $p_i^k \in P_i^k$, check if the point has already been processed, and if not add it to Q;

(iv) Terminate the algorithm when all the points $p_i \in P$ have been processed and are now part of the list of point clusters C.

In this algorithm, the most important parameter is $d_{th} = \sqrt{(x-x_0)^2 + (y-y_0)^2 + (z-z_0)^2}$, which represents the radius threshold of clustering. Within this radius, all the points are clustered into a point cloud cluster. By clustering, we can detect the edges of objects and determine the minimum number of 3D bounding boxes in the point cloud that surround the objects

IV. EXPERIMENTAL ANALYSIS

To verify the effectiveness of this method of intelligent detection for tunnel shotcrete spraying, we selected a tunnel in China under construction as the test environment and set up a series of contrast experiments. These sensors can be integrated into a Robot Operating System (ROS), which is a robotic software platform that provides system-like functionality for heterogeneous computer clusters. We used a laptop to collect the raw point cloud and RGB image data through the bag recording function in the ROS for the subsequent processes. **Fig. 10** shows the basic flowchart of intelligent detection for tunnel shotcrete spraying.

A. EXPERIMENTAL EQUIPMENT AND DATASET

The dataset for verification was captured using a Velodyne 3D laser scanner and an RGB camera. We have tested that this LiDAR, VLP-16, can reflect the shape of the tunnel well in low light condition with little effect from the dispersed nature of light. The sampling frequency of the Velodyne VLP-16 is 10 Hz with a 30° vertical field of view (±15° up and down), and this scanner provides 16 equally spaced angular subdivisions (approximately 1.8°). The maximum detection distance is 120m. Valid values for azimuth range from 0 to 359.99 degree, which means each scan of LiDAR yields 640,000 points. To ensure the successful combination of different sensors, the frame rate of camera and LiDAR is fixed. The reconstruction of 3D tunnel model is achieved only by LiDAR, so the parameters of camera will not affect the accuracy of 3D model. We used the ROS in the Ubuntu system to record the real-time scanned 3D point cloud data in the tunnel for subsequent processing. In terms of deep learning, we used 1000 tunnel construction images as the dataset for the detection for tunnel shotcrete spraying. The dataset was divided into three parts randomly, 640 images for training, 160 images for validation, and 200 images for testing. All images were labeled by the tool in Matlab named 'Image Labeler'. Because of the limited condition, the data were collected from the same tunnel, which means our data are extremely conformed to the Independently identically distribution (IIP) condition, and there is only one class need to be detected in the part of CNN Detection Module. To prevent overfitting problem, the data were sampled with different angle and distance, so the numbers of arches at each image are not certain. Each image in training or testing data set has one to four steel arches. To ensure the definition of images, we used the mobile shotcrete sprayer to light the construction region.

B. MODEL EVALUATION

1) 3D MODELING AND PROFILE DETECTION

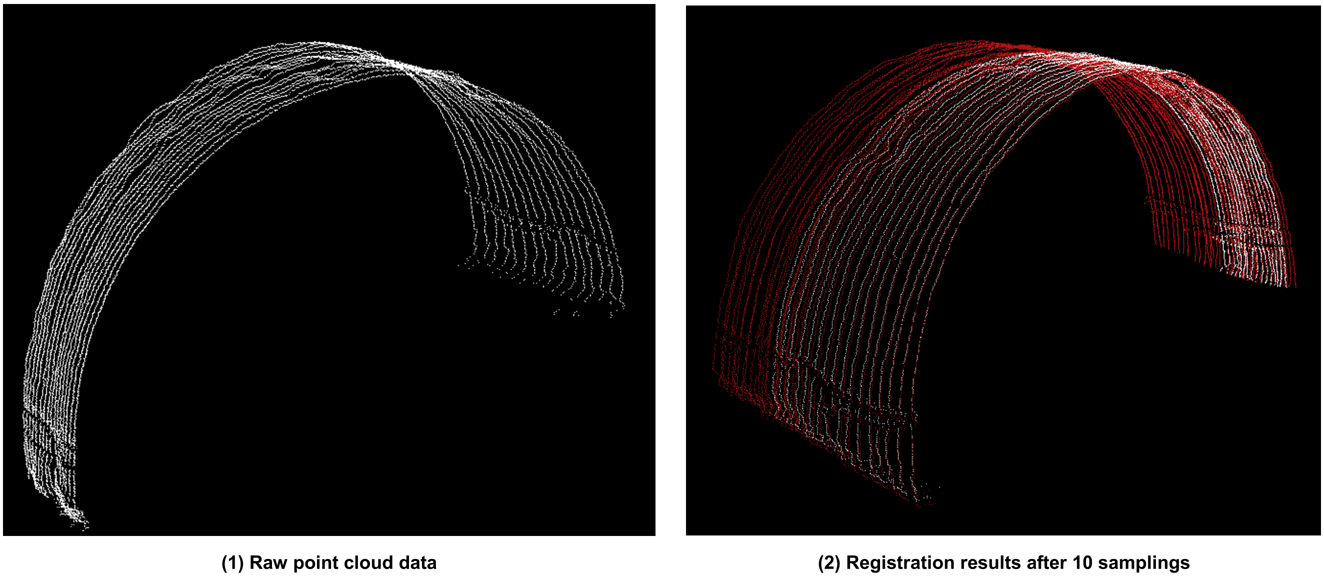
To reduce the interference of the environment, we filtered the input point cloud data by means such as removing the point cloud data where the distance is too close and within a certain range of angles. **Fig. 12** shows the performance of the point cloud registration using ICP. When the number of iterations is 5, a better registration result can be achieved. **Fig. 14** shows the results of the 3D reconstruction of the tunnel. Thereby, the stopping criterion for the spraying can be obtained.

2) MODEL TRAINING AND VERIFICATION

The hardware environments used for the training and testing were as follows:

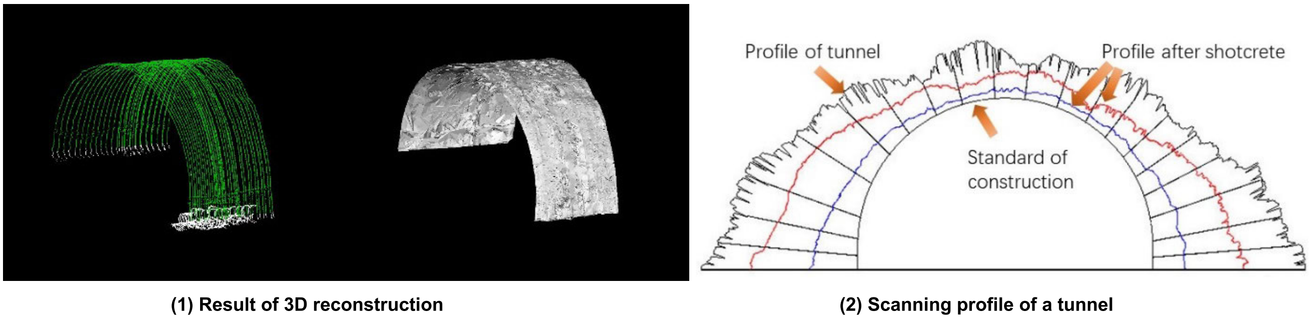
- OS: Ubuntu 16.04 LTS (64 bit);
- CPU: Intel(R) Core (TM) i7-9700k CPU @ 3.60 GHz
- CPU RAM: 16GB
- GPU: GeForce RTX 2070
- Graphics card RAM size: 8 GB

Table 1 presents the results of different object detection algorithms including YOLOv3, YOLOv3-tiny, Faster-RCNN with VGG16 net, Complex-YOLO, our proposed model without any compression, with a 0.9 compression ratio, and with a 0.95 compression ratio, and DPM. The Faster-RCNN, YOLO algorithms and our proposed model are deep learning based


(1) Raw point cloud data

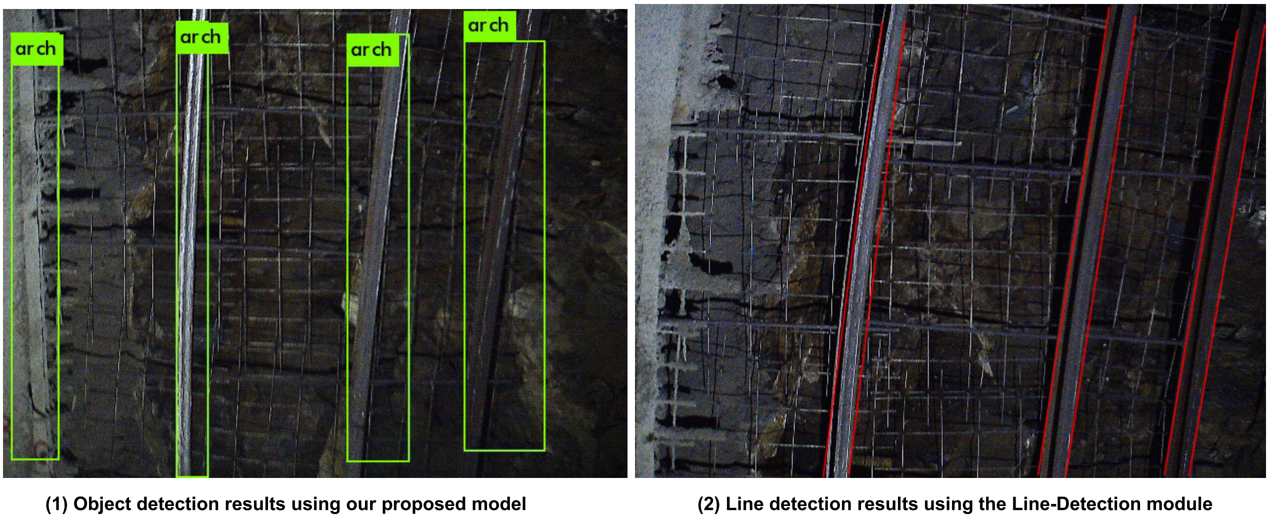
(2) Registration results after 10 samplings

FIGURE 12. Result of point cloud registration.


(1) Result of 3D reconstruction

(2) Scanning profile of a tunnel

FIGURE 13. Visualization of tunnel scanning data.


(1) Object detection results using our proposed model

(2) Line detection results using the Line-Detection module

FIGURE 14. Results of object detection in images.

and the DPM algorithm is a model-based method. In addition to these algorithms used in RGB image, we also evaluate the performance of Complex-YOLO which is a state-of-the-art real time 3D object detection network on point clouds only [56]. To train the compared networks for a fair comparison, a momentum of 0.9, a weight decay of 0.0005, a training rate of 0.001 and a number of iterations of 40,000 are used in the different deep learning methods of these networks.

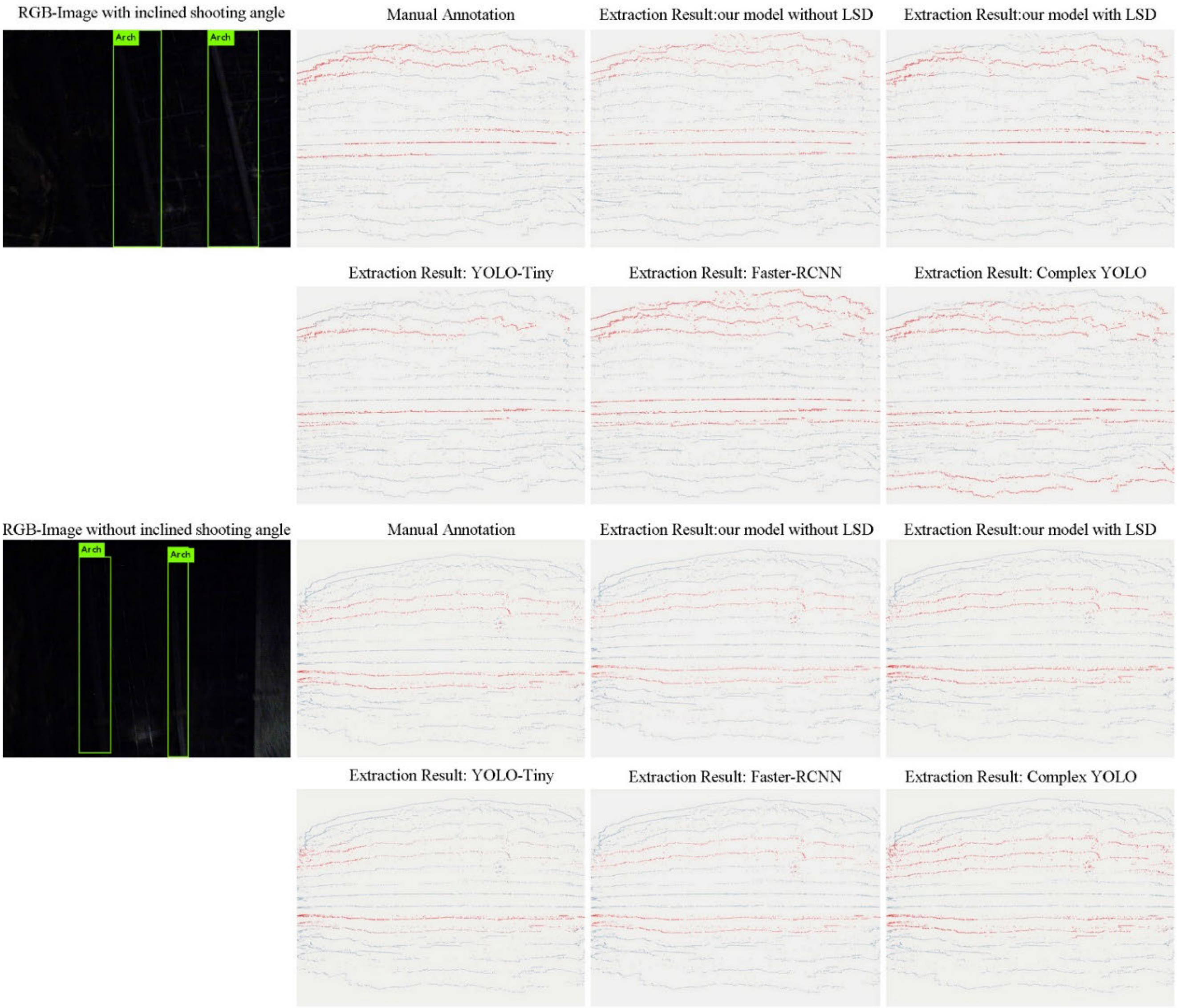


FIGURE 15. Results of object detection in point cloud.

TABLE 2. Object detection results in point cloud with inclined shooting angle.

Algorithm name	Our model without LSD	Our model with LSD	Projective method: YOLO-Tiny	Projective method: Faster-RCNN	Complex-YOLO
Pression	68.3%	85.4%	41.3%	68.5%	69.8%

TABLE 3. Object detection results in point cloud without inclined shooting angle.

Algorithm name	Our model without LSD	Our model with LSD	Projective method: YOLO-Tiny	Projective method: Faster-RCNN	Complex-YOLO
Pression	93.1%	91.8%	57.7%	94.5%	75.6%

The YOLO and our proposed model used the Darknet framework, and the Faster-RCNN and Complex-YOLO used the Tensorflow framework. As there is not available to calculate the mAP of Complex-YOLO in RGB image, we evaluate it according to the 3D bounding box overlap. Obviously, the mAP of the Faster-RCNN algorithm is better than those

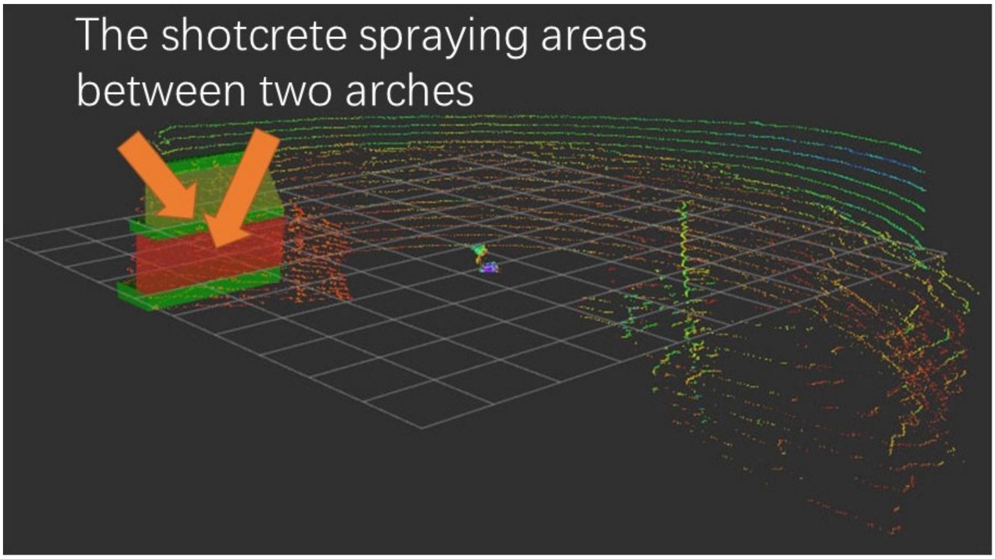


FIGURE 16. Detection of shotcrete spraying areas in 3D point cloud data.

of the series of the YOLO algorithms in general, and the DPM performed the worst. However, because the sampling frequency of our LiDAR is 10 Hz, the detection speed should be higher than this value to guarantee the fusion of the different data. Our proposed structure with a 0.9 compression ratio performs the best in the balance of accuracy, processing speed, and volume size, with the mAP reaching 90% while maintaining a processing speed of greater than 31 frames per second (FPS). In addition, the model volume is only 6.79 MB. The object detection results are shown in **Fig.11** and **Fig.14**.

3) FUSION OF CAMERA AND LiDAR DATA

During the construction of tunnel, workers do not spray the shotcrete on the entire surface of the tunnel but focus on the part of it. After this part of shotcrete solidifies, workers will spray the next part, which is not a continuous process. Our sensors are attached on the robotic arm, which allows the scope of scanning always match the spraying process. In total, the process of detection is dynamic, so even if the scanning scope after the combination of LiDAR and camera is a small part, it will not affect the result of our model. As the position relationship between LiDAR and car body is certain, we can transform the point cloud data obtained by LiDAR to the fixed coordinate system on mobile shotcrete sprayer. This transformation relation is a kinematics problem ant it is relatively easy to figure out once the sizes of each part of sprayer are known. The current collected tunnel point cloud contour will be aligned with the standard structure manually before the processing on it.

In addition to **Table 1**, we also evaluated the detection results of our method and other state-of-the-art algorithms in the point cloud data. The results are presented in **Fig. 15**, **Table 2**, and **Table 3**. Similar to the calculation of IoU, the overlap rate between the outputs of algorithm and the ground truth marked manually is used to represent the detection accuracy in point cloud, shown as follow. *TN* represents the number of predicted points in the sets of ground truth, and *FN* represents the number not in the sets.

$$overlap = \frac{TN}{TN + FN} \tag{5}$$

Our proposed model with LSD performed the best both in the pression and speed among these algorithms, which means our model is more robust. Other projective methods based on different backbone CNN structure can work better when the camera is facing to the steel arches. As the Complex-YOLO is trained on points cloud only, the differences of shooting angle will not greatly affect its performance, but its mAP is worse than other deep learning- based methods.

The 3D point cloud projection of the shotcrete spray area detection results using the neural network in the image is shown in **Fig. 16**. According to the point cloud clustering algorithm, the shotcrete spray area can be better identified. Therefore, we can sample the point cloud data of each line between this area in turn to direct the mobile shotcrete sprayer.

V. CONCLUSION

With the development of human society, more tunnels and underground structures need to be built, but the traditional process of spraying in a tunnel is time-consuming, labor-intensive, and error-prone. In this paper, we combine deep learning and LiDAR to realize the intelligent detection for tunnel shotcrete spraying. For the detection of the shotcrete spraying area, the basic idea of the method is to use a deep learning algorithm to detect the object in the image, then project the object into the corresponding point cloud region, and finally determine the spraying area using point cloud clustering. For the detection of shotcrete spraying quality, we use LiDAR to construct a 3D point cloud model of the tunnel and use ICP and the greedy projection triangulation algorithm to achieve 3-dimensional reconstruction and detection of the tunnel profile in real time. Combining deep learning algorithms and LiDAR can ensure the accuracy of object detection and reflect the physical characteristics of tunnels in the real world. The presented deep learning and LiDAR-based method can serve as a reference for subsequent research on the automation of tunneling equipment.

REFERENCES

[1] L. Luo, X. Li, M. Tao, and L. Dong, "Mechanical behavior of rock-shotcrete interface under static and dynamic tensile loads," *Tunnelling Underground Space Technol.*, vol. 65, pp. 215–224, May 2017.

[2] K.-I. Song and G.-C. Cho, "Bonding state evaluation of tunnel shotcrete applied onto hard rocks using the impact-echo method," *NDT & E Int.*, vol. 42, no. 6, pp. 487–500, Sep. 2009.

[3] N. Ginouse, M. Jolin, and B. Bissonnette, "Effect of equipment on spray velocity distribution in shotcrete applications," *Construct. Building Mater.*, vol. 70, pp. 362–369, Nov. 2014.

[4] N. Ginouse and M. Jolin, "Investigation of spray pattern in shotcrete applications," *Construction Building Mater.*, vol. 93, pp. 966–972, Sep. 2015.

[5] J.-Y. Han, J. Guo, and Y.-S. Jiang, "Monitoring tunnel profile by means of multi-epoch dispersed 3-D LiDAR point clouds," *Tunnelling Underground Space Technol.*, vol. 33, pp. 186–192, Jan. 2013.

[6] J.-Y. Han, J. Guo, and Y.-S. Jiang, "Monitoring tunnel deformations by means of multi-epoch dispersed 3D LiDAR point clouds: An improved approach," *Tunnelling Underground Space Technol.*, vol. 38, pp. 385–389, Sep. 2013.

[7] W. Zhang, W. Qiu, D. Song, and B. Xie, "Automatic tunnel steel arches extraction algorithm based on 3D LiDAR point cloud," *Sensors*, vol. 19, no. 18, p. 3972, Sep. 2019.

[8] P. H. Lin, H. Tserng, and C. C. Lin, "Automated construction of the Paghuashan tunnel for Taiwan high speed rail (THSR) project," *Autom. Construct.*, vol. 15, no. 5, pp. 627–639, Sep. 2006.

[9] M.-Y. Cheng, Y. Liang, C.-M. Wey, and J.-C. Chen, "Technological enhancement and creation of a computer-aided construction system for the shotcreting robot," *Autom. Construct.*, vol. 10, no. 4, pp. 517–526, May 2001.

[10] J. Mah, S. D. Mckinnon, C. Samson, and D. Thibodeau, "Wire mesh filtering in 3D image data of rock faces," *Tunnelling Underground Space Technol.*, vol. 52, pp. 111–118, Feb. 2016.

[11] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Commun. ACM*, vol. 60, no. 6, pp. 84–90, May 2017.

[12] J. R. R. Uijlings, K. E. A. Van De Sande, T. Gevers, and A. W. M. Smeulders, "Selective search for object recognition," *Int. J. Comput. Vis.*, vol. 104, no. 2, pp. 154–171, Sep. 2013.

[13] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R–CNN: Towards real–time object detection with region proposal networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1137–1149, Jun. 2017.

[14] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "SSD: Single shot MultiBox detector," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Amsterdam, The Netherlands, 2016, pp. 21–37.

[15] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Las Vegas, NV, USA, Jun. 2016, 779–788.

[16] A. Asvadi, L. Garrote, C. Premebida, P. Peixoto, and U. J. Nunes, "Multimodal vehicle detection: Fusing 3D-LIDAR and color camera data," *Pattern Recognit. Lett.*, vol. 115, pp. 20–29, Nov. 2018.

[17] Q. Wang and M.-K. Kim, "Applications of 3D point cloud data in the construction industry: A fifteen-year review from 2004 to 2018," *Adv. Eng. Inform.*, vol. 39, pp. 306–319, Jan. 2019.

[18] M. Himmelsbach, A. Müller, T. Luettel, and H.-J. Wünsche, "LIDAR-based 3D object perception," presented at the Proc. 1st Int. Workshop Cognition Tech. Syst., Munich, Germany, 2008.

[19] Z. Zhou and J. Gong, "Automated residential building detection from airborne LiDAR data with deep neural networks," *Adv. Eng. Inform.*, vol. 36, pp. 229–241, Apr. 2018.

[20] M. M. Rahman, Y. Tan, J. Xue, L. Shao, and K. Lu, "3D object detection: Learning 3D bounding boxes from scaled down 2D bounding boxes in RGB–D images," *Inf. Sci.*, vol. 476, pp. 147–158, Feb. 2019.

[21] A. Asvadi, C. Premebida, P. Peixoto, and U. Nunes, "3D Lidar-based static and moving obstacle detection in driving environments: An approach based on voxels and multi-region ground planes," *Robot. Auto. Syst.*, vol. 83, pp. 299–311, Sep. 2016.

[22] B. Wu, A. Wan, X. Yue, and K. Keutzer, "SqueezeSeg: Convolutional neural nets with recurrent CRF for real-time road-object segmentation from 3D LiDAR point cloud," presented at the IEEE Int. Conf. Robot. Automat. (ICRA), Brisbane, QLD, Australia, 2018.

[23] B. Leng, Y. Liu, K. Yu, X. Zhang, and Z. Xiong, "3D object understanding with 3D convolutional neural networks," *Inf. Sci.*, vol. 366, pp. 188–201, Oct. 2016.

[24] D. Maturana and S. Scherer, "VoxNet: A 3D convolutional neural network for real-time object recognition," presented at the IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS), Hamburg, Germany, 2015.

[25] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, May 2015.

[26] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part-based models," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 9, pp. 1627–1645, Sep. 2010.

[27] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," presented at the IEEE Conf. Comput. Vis. Pattern Recognit., Columbus, OH, USA, 2014.

[28] T.-Y. Lin, P. Dollar, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," presented at the IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Honolulu, HI, USA, 2017.

[29] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollar, "Focal loss for dense object detection," presented at the IEEE Int. Conf. Comput. Vis. (ICCV), Venice, Italy, 2017.

[30] W. Fang, L. Ding, B. Zhong, P. E. Love, and H. Luo, "Automated detection of workers and heavy equipment on construction sites: A convolutional neural network approach," *Adv. Eng. Inform.*, vol. 37, pp. 139–149, Aug. 2018.

[31] R. Zuo, Y. Xiong, J. Wang, and E. J. M. Carranza, "Deep learning and its application in geochemical mapping," *Earth-Sci. Rev.*, vol. 192, pp. 1–14, May 2019.

[32] Y. Tian, G. Yang, Z. Wang, H. Wang, E. Li, and Z. Liang, "Apple detection during different growth stages in orchards using the improved YOLO–V3 model," *Comput. Electron. Agricult.*, vol. 157, pp. 417–426, Feb. 2019.

[33] G. Zhao, X. Xiao, J. Yuan, and G. W. Ng, "Fusion of 3D-LIDAR and camera data for scene parsing," *J. Vis. Commun. Image Represent.*, vol. 25, no. 1, pp. 165–183, Jan. 2014.

[34] M. Engelcke, D. Rao, D. Z. Wang, C. H. Tong, and I. Posner, "Vote3Deep: Fast object detection in 3D point clouds using efficient convolutional neural networks," presented at the IEEE Int. Conf. Robot. Automat. (ICRA), Singapore, 2017.

[35] L. Xiao, R. Wang, B. Dai, Y. Fang, D. Liu, and T. Wu, "Hybrid conditional random field based camera-LIDAR fusion for road detection," *Inf. Sci.*, vol. 432, pp. 543–558, Mar. 2018.

[36] L. Caltagirone, M. Bellone, L. Svensson, and M. Wahde, "LIDAR-camera fusion for road detection using fully convolutional neural networks" *Robot. Auto. Syst.*, vol. 111, pp. 125–131, Jan. 2019.

[37] H. Yang, X. Xu, B. Kargoll, and I. Neumann, "An automatic and intelligent optimal surface modeling method for composite tunnel structures," *Compos. Struct.*, vol. 208, pp. 702–710, Jan. 2019.

[38] P. Lai and C. Samson, "Applications of mesh parameterization and deformation for unwrapping 3D images of rock tunnels," *Tunnelling Underground Space Technol.*, vol. 58, pp. 109–119, Sep. 2016.

[39] W. Li and P. Song, "A modified ICP algorithm based on dynamic adjustment factor for registration of point cloud and CAD model," *Pattern Recognit. Lett.*, vol. 65, pp. 88–94, Nov. 2015.

[40] D. Zhu, H. Guo, and W. Su, *Point Cloud Library PCL Learning Tutorial*. Beijing, China: Beihang Univ. Press, 2012, p. 402.

[41] S. R. Maurya and G. M. Magar, "Performance of greedy triangulation algorithm on reconstruction of coastal dune surface," presented at the 3rd Int. Conf. Converg. Technol. (I2CT), Pune, India, 2018.

[42] R. Zhang, G. Wang, J. Ma, Y. Wu, and G. Zhang, "Study of Huizhou architecture component point cloud in surface reconstruction," *IOP Conf. Ser., Earth Environ. Sci.*, vol. 69, Jun. 2017, Art. no. 012087.

[43] G. Bradski, A. Kaehler, and M. Loukides, Eds. *Learning OpenCV*. Newton, MA, USA: O'Reilly Media, 2008, p. 555.

[44] M. Velas, M. Spanel, Z. Materna, and A. Herout, "Calibration of RGB camera with velodyne LiDAR," presented at the 22nd Int. Conf. Central Eur. Comput. Graph., Vis. Comput. Visio, 2014.

[45] Z. Zhang, "Flexible camera calibration by viewing a plane from unknown orientations," presented at the 7th IEEE Int. Conf. Comput. Vis., Corfu, Greece, 1999.

[46] M. A. Al-masni, M. A. Al-antari, J.-M. Park, G. Gi, T.-Y. Kim, P. Rivera, E. Valarezo, M.-T. Choi, S.-M. Han, and T.-S. Kim, "Simultaneous detection and classification of breast masses in digital mammograms via a deep learning YOLO-based CAD system," *Comput. Methods Programs Biomed.*, vol. 157, pp. 85–94, Apr. 2018.

[47] J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement," Apr. 2018, *arXiv:1804.02767*. [Online]. Available: https://arxiv.org/abs/1804.02767

[48] F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally, and K. Keutzer, "SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5 MB model size," presented at the Int. Conf. Learn. Representations (ICLR), Toulon, France, 2016.

[49] S. Han, H. Mao, and W. J. Dally, "Deep compression: Compressing deep neural networks with pruning, trained quantization and Huffman coding," presented at the Int. Conf. Learn. Representations (ICLR), Toulon, France, 2016.

[50] P. Zhang, Y. Zhong, and X. Li, "SlimYOLOv3: Narrower, faster and better for real-time UAV applications," Jul. 2019.

[51] Y. Xu-qiang, F. Yong, and L. Hong-chen, "An object extraction method based on HSI color model," *Opt. Technique*, vol. 32, no. 2, pp. 290–292, 2006.

[52] R. Von Gioi, J. Jakubowicz, J.-M. Morel, and G. Randall, "LSD: A fast line segment detector with a false detection control," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 4, pp. 722–732, Apr. 2010.

[53] S. Kato, E. Takeuchi, Y. Ishiguro, Y. Ninomiya, K. Takeda, and T. Hamada, "An open approach to autonomous vehicles," *IEEE Micro*, vol. 35, no. 6, pp. 60–68, Nov./Dec. 2015.

[54] A. W. Moore, *An Intoductory Tutorial on Kd-Trees*. Cambridge, U.K.: Cambridge Univ. Press, 1991.

[55] H. Najdataei, Y. Nikolakopoulos, V. Gulisano, and M. Papatriantafilou, "Continuous and parallel LiDAR point-cloud clustering," presented at the IEEE 38th Int. Conf. Distrib. Comput. Syst. (ICDCS), Vienna, Austria, 2018.

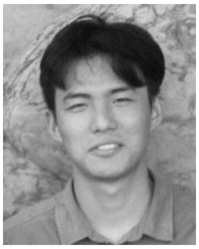
[56] M. Simon, S. Milz, K. Amende, and H.-M. Gross, "Complex-YOLO: An euler-region-proposal for real-time 3D object detection on point clouds," presented at the Eur. Conf. Comput. Vis. (ECCV), Cham, Switzerland, 2018.

LUO CHUN-LEI received the B.S., M.S., and Ph.D. degrees in mechanical and electrical engineering from Central South University, China, in 1990, 1995, and 2009, respectively.

He has been involved in the research and teaching of hydraulic machinery and control technology for 25 years. He has been involved in the research of electro-hydraulic system integration theory and technology for a long time such as the research and design of high-pressure large-flow electro-hydraulic proportional flow valve, proportional control systems, large-scale integrated hydraulic test bench, hydraulic test bench, and special equipment hydraulic system and components. He has published one academic monograph, more than 40 scientific research and teaching reform articles in SCI, EI, and CSCD journals and five identified results. He holds more than 20 patent applications (including nine invention patents), from 2010 to 2018. He has received three provincial and ministerial level scientific achievements (including the Provincial First Prize for Scientific and Technological Progress, in 2011).

SHA HAO (Student Member, IEEE) received the B.S. degree in mechanical engineering from the Luoyang Institute of Science and Technology, Henan, China, in 2017. He is currently pursuing the master's degree in mechanical engineering with the Mechanical and Electrical College, Central South University, Changsha, China. He joined Central South University, in August 2017. He has coauthored one conference publications. He is currently involved in the automation of equipment of engineering systems. His current research interests include multisensor fusion and deep learning.

LING CHUN-LAI (Student Member, IEEE) received the B.S. degree in mechanical engineering from Sichuan Agricultural University, China, in 2017. He is currently pursuing the master's degree in mechanical engineering with the Mechanical and Electrical College, Central South University, Changsha, China, where he joined Central South University, in August 2017. His current research interests include hydraulic simulation and intelligent detection.



LI JIN-YANG (Student Member, IEEE) is currently pursuing the master's degree with the College of Mechanical and Electrical Engineering, Central South University. His current research interests include hydraulic simulation and intelligent detection.