IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2017.DOI

Downhole Track Detection via Multi-scale Conditional Generative Adversarial Nets

XING WEI[1], JIA LI[2], GUOQIANG YANG[3], YANG LU[4]

[1]School of Computer and Information, Hefei University of Technology,Hefei 230009, China. (e-mail: weixing@hfut.edu.cn)
[2]School of Computer and Information, Hefei University of Technology,Hefei 230009, China. (e-mail: lijiajia@mail.hfut.edu.cn)
[3]School of Computer and Information, Hefei University of Technology,Hefei 230009, China.(e-mail: ygqhfut@163.com)
[4]School of Computer and Information, Hefei University of Technology,Hefei 230009, China.(e-mail: luyang.hf@126.com)

Corresponding author: Xing Wei (e-mail: weixing@hfut.edu.cn).

This work was supported by National Key R&D Program of China (201904d07020008) and Anhui Provincial Key R&D Program (2018YFC0604404).

ABSTRACT Frequent mine disasters cause a large number of casualties and property losses. Autonomous driving is a fundamental measure for solving this problem, and track detection is one of the key technologies for computer vision to achieve downhole automatic driving. The track detection result based on the existing models lacks the semantic detail of the tracks and relies too much on visual postprocessing technology. Therefore, this paper proposes a track detection model based on the multi-dimensional conditional generative adversarial network. First, the generator is decomposed into global and local parts using a multi-granularity structure. Second, a multi-scale shared convolution structure is introduced into the discriminator to further guide the generator. In addition, this paper proposes a penalty mechanism based on Monte Carlo search to enhance the semantic constraints in the image generation process. Compared with the state-of-the-art semantic segmentation algorithms, extensive experiments on the downhole scene dataset demonstrate proposed model achieved the best results in terms of pixel accuracy, intersection-over-union (IOU) and the track detection accuracy. This paper provides a new idea for track line detection. In the future, the model can also beapplied to other segmentation problems as well. Code and data will be shared.

INDEX TERMS Track detection, Conditional generative adversarial nets, Multi-scale information, Monte Carlo search, Automatic driving downhole

I. INTRODUCTION

In recent years, the frequent occurrence of large-scale mine accidents has caused a large number of casualties and property losses. The production and transportation in mining need to be developed in an unmanned and intelligent direction. As the unmanned research of ground scenes becomes more mature, there is a certain research basis for implementing automatic driving under the mine. The underground mine locomotive needs the track line as an aid in the safe underground driving process. Therefore, it is necessary to detect whether there are pedestrians or obstacles on the track in front of the currently running locomotive. If the above situation is encountered, it needs to be dealt with rapidly. Therefore, underground automatic driving provides a reliable method to ensure the safety of the lives and property of underground workers.

Track detection refers to recognizing the track area in a video or image by image processing technology, which shows the specific position of the track line. Track detection is one of the key technologies in computer vision for underground automatic driving. It can assist in the detection of pedestrians and obstacles and further improve the driving safety of underground locomotives. However, underground track detection is easily affected by complex environmental factors, such as light changes, water cover and cable interference. Thus, in recent years, track detection has become a challenging task in studying computer vision.

Track detection algorithms based on traditional image processing can be roughly divided into two categories: feature-based methods and model-based methods. Feature-based track detection technology [1], [2] mainly uses feature information such as track edge, texture, color, geometry and gray

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2019.2951860, IEEE Access

value to distinguish the track area from the surrounding environment. The track area is extracted, and the specific position information of the track in the image is obtained. However, this method relies too much on the underlying features of the image and the surrounding environment easily interferes, which creates considerable challenges for subsequent work and affects the final detection effect of the track. The basic principle of the model-based track detection method [3] is to transform the track detection problem into a problem of solving the track model parameters. According to the track pattern in the local area, the fitting of the track line is achieved by using a segmentation line, a parabola, a hyperbola or a spline curve to describe the model. However, a road model often cannot adapt to multiple road conditions at the same time, and the shape of the track varies widely. Thus, the track is difficult to detect with a linear model. The algorithm lacks the robustness and flexibility for any road shape.

Recently, deep neural networks have been used to replace hand-crafted features to achieve track line detection. The deep convolution neural network (DCNN) has been successfully applied to many computer vision tasks. The problem of track line detection is solved as an image segmentation task [4], [5]. The result of the final output of the network is the probability of each track pixel, i.e., the prediction of the pixel. The network predicts the pixels at the position of the track, then combines the pixels of the same track, and finally displays the position of the track line in the target image. However, the problem of downhole track detection scenes is not a direct classification task for track line pixels. Moreover, the prediction of the track line needs to preserve the structure or quality of the equivalent track, that is, the fineness and uniqueness of the track line. In addition, in the process of training, it is necessary to manually design a complex loss function that is suitable for improving the final detection effect. Finally, in the process of displaying the images, to retain better detection results, more postprocessing techniques are needed, which also increases the complexity of the application of such methods.

Another method for solving the above problem is to use a generative adversarial network (GAN) [6]. The GAN contains two opposing models: a generative model G for fitting the sample data distribution and a discriminative model D for judging the true and false data. However, one of the disadvantages of GANs is that the training is unstable; that is, the data generated by the generated model are random and uncontrollable.

The conditional generative adversarial network (CGAN) [7] adds an additional conditional y to generator G and discriminator D on the basis of a GAN. This condition is actually the label that is generated. The generator must generate a sample that matches the condition y. The discriminator must determine not only whether the image is true but also whether the image and the condition match. Some scholars have achieved good results in image generation via CGANs. This task is a type of visual and graphical problem in which the goal is to use paired images to train the network to learn the mapping between the input image and the output image [8]. For example, Isola et al. [9] proposed a network called the pix2pix framework for paired image transformation based on a CGAN. This method has achieved good results. However, the model is limited to generating low-resolution images, and images still lack texture and detail. Recently, Chen and Koltun [10] used modified perceptual loss [11], [13] to generate images. Although models can generate high-resolution images, generated images often lack fine detail and realistic texture.

In view of the shortcomings of previous works, this paper proposes a downhole track line detection model based on CGAN. We use the method of adversarial learning to solve the problem of artificially designing complex loss functions and introduce Monte Carlo search [14] technology into the generator network. Monte Carlo searches have been widely used in text generation tasks. In [15], [16], researchers used a Monte Carlo search to enhance the constraints on content and emotion in the process of text generation. This paper introduces a Monte Carlo search to solve the problem of generating image distortion and lack of precision. In addition, a convolution sharing layer is added to the discriminator network to facilitate learning the discriminator. In summary, the paper makes the following contributions:

- This paper proposes a downhole track detection model based on the multi-scale CGAN, which can generate high resolution (up to 2K) semantic segmentation images.
- This paper proposes a penalty mechanism based on Monte Carlo search to enhance the semantic constraints in the image generation process, which makes the network output to be more realistic or better structure-preserving, decreasing the dependency on potentially complex post-processing.
- To promote the fusion and learning of global information and local information, this paper introduces a multitask learning strategy based on parameter sharing in the discriminator network, which indirectly expands the storage capacity of the discriminator model and accelerates the model convergence.
- Experimental results demonstrate that compared with the state-of-the-arts, proposed model visually produces results more similar to the ground truth labels. The proposed model can also be flexibly applied to other segmentation problems as well.

The structure of the rest of the paper is as follows. The second section introduces the proposed model. The third section shows the evaluation indicators, network structure and implementation details of the experiments. The fourth section analyzes the experimental results, and the fifth section gives the conclusions and future work.

II. RELATED WORK

A. IMAGE-TO-IMAGE TRANSLATION

Many researchers have leveraged adversarial learning for image-to-image translation [9], which translates an input

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2019.2951860, IEEE Access

image from one domain to another domain given input-output image pairs as training data. CGANs aim to model the conditional distribution of real images given the input semantic label maps via the following minimax game:

$$\min_G \max_D \mathcal{L}_{GAN}(G, D) \tag{1}$$

Where G and D represent the generator and discriminator in CGAN, respectively. s and x represent the introduced auxiliary variables and inputs in CGAN. Where the objective function $\mathcal{L}_{GAN}(G, D)$ is given by:

$$E_x[\log D(s, x)] + E_{x,s}[\log(1 - D(s, G(x, s))] \tag{2}$$

Pix2Pix: Adversarial loss has become a popular choice for many image translation tasks because the discriminator can learn the trainable loss function and automatically adapt to the differences between the generated and real images in the target domain. The pix2pix method is a CGAN framework for image-to-image translation. The general flow of the model to solve image translation is shown in FIGURE 1.

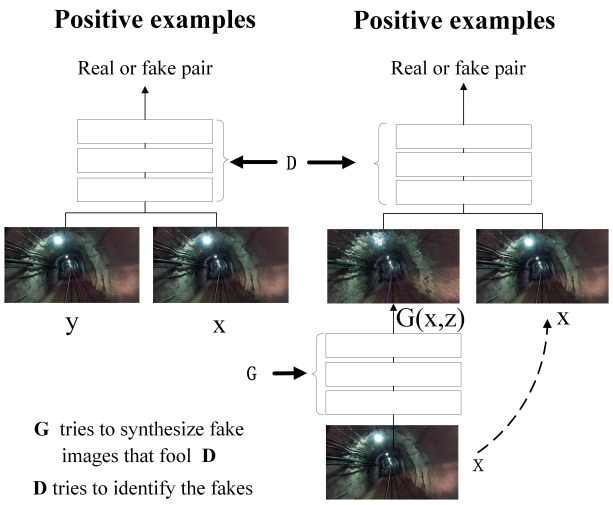

FIGURE 1: CGAN track detection process

In FIGURE 1, x represents the original image, G(x) represents the image generated by the generator containing the track line label, and y represents the true image containing the label. For the purpose of this paper, the goal is to input an image containing a downhole track line, and generator G generates an image that marks the existing track line. In other words, the training dataset is given as a set of pairs of corresponding images (s_i, x_i), where s_i is a semantic label map, and x_i is the corresponding natural image. However, in the application of downhole roadway scenes, the results generated by pix2pix are limited to low-resolution images. The generated image still has a large gap from the real sample, and the image still lacks texture and details.

Perceptual Loss: Several recent works [13], [17], [18] specifically targeting image super-resolution are based on the idea that pixel-level objective losses are often not sufficient to ensure high-level semantics of a generated image. Therefore, they suggest capturing higher-level representations of images from the representations of a separate network at a given layer. In image super-resolution, the corresponding ground truth label for a given low-resolution image is often available. Therefore, a difference measure between the high-level representations of the reconstructed and ground truth images is considered as an extra loss term. Our work is inspired by this idea. Similarly, we propose using the difference between the labels and predictions in a high-level embedding space.

B. TRACK LINE DETECTION

Since this paper's work focuses on track detection of downhole roadway scenes, the other related methods for this issue need to be discussed . In the feature-based approach, Quach et al. [1] proposed color and depth information recorded using a single RGB-D camera to better handle unfavorable factors such as lighting conditions and lane-like objects. However, since this method relies on the underlying features of the image, environmental factors easily interfere, making the algorithm less robust, and thus, the application effect is not good. Model-based methods, such as Bente et al. [3], proposed a lane detection method using the Hough transform and contour detection. They determine the corresponding model parameters by analyzing the target information in the road image. Therefore, it is robust to the presence of occlusion and interference in the lane line. However, a road model often cannot adapt to multiple road conditions at the same time. The algorithm lacks robustness and flexibility for any road shape.

Additionally, some scholars have proposed using DCNNs to detect lane lines. In [4], Pan et al. trained a spatial convolutional neural network (SCNN) for specific problems and added postprocessing techniques that rely on handcrafting. Another recent example is the work of Neven et al. [5], which first used a segmentation network to obtain a lane marker prediction map. The second network was then trained to perform a constrained perspective transformation, and finally, the network used curve fitting to obtain the final result. However, their method relies on more postprocessing techniques, which increases the complexity of the actual application of the model. The paper used a downhole scene dataset to train the models in [4] and [5] and compared them with the model proposed in this paper. The results of the comparison are shown in the experimental section.

III. MULTIDIMENSIONAL GENERATIVE ADVERSARIAL MODEL

A. MULTI-GRANULARITY GENERATOR

Referring to the hierarchical reinforcement learning in [19], we decompose the generator into two sub-generators G_1 and G_2, where G_1 is the global generator, G_2 is the local generator, and the overall structure of the generator $G = \{G_1, G_2\}$ is shown in FIGURE 2. The global generator is mainly used for the overall information construction of images. The local generator can effectively improve the resolution of the

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2019.2951860, IEEE Access

generated image. For example, an image with a resolution of 1024*512 is input into the generator, and the local generator output resolution is 2048*1024. The model proposed in [13] increases the resolution of the image to 512*512. The global generator of the model proposed in this paper is designed based on the above work. It consists of 3 components: a convolutional frontend $G_1^{(F)}$, a set of residual blocks $G_1^{(R)}$, and a transposed convolutional backend $G_1^{(B)}$. A semantic label map of resolution 1024*512 is passed through the 3 components sequentially to output an image of resolution 1024*512. The local enhancer network also consists of 3 components: a convolutional frontend $G_2^{(F)}$, a set of residual blocks $G_2^{(R)}$, and a transposed convolutional backend $G_2^{(B)}$. The resolution of the input image to G_2 is 2048*1024. Different from the global generator network, the input to the residual block $G_2^{(R)}$ is the elementwise sum of two feature maps: the output feature map of $G_2^{(F)}$ and the last feature map of the backend of the global generator network $G_1^{(B)}$, which helps to integrate the global information from G_1 to G_2.

In the experiment, we first downsample the original 2048*1024 image to obtain a low-resolution image of 1024*512. We use high-resolution images to train the local generator and low-resolution images to train the global generator. Finally jointly fine-tune all the networks together. Experimental results show that this hierarchical generator can effectively integrate global and local information to generate high-resolution images.

B. MULTI-SCALE SHARED CONVOLUTION DISCRIMINATOR

The structure of the discriminator is critical to generating high-resolution images. To differentiate high-resolution real and synthesized images, the discriminator needs to have a large receptive field, which requires either a deeper network or larger convolutional kernels. As both choices lead to increased network capacity, overfitting becomes more of a concern. Additionally, both choices require a larger memory footprint for training, which is already a scarce resource for high-resolution image generation.

TABLE 1: The network structure of the discriminators

Layer	Layer Information
Input Layer	CONV-(N64,K4x4,S2,P1), Leaky ReLU
Hidden Layer	CONV-(N128,K4x4,S2,P1), Leaky ReLU
Hidden Layer	CONV-(N256,K4x4,S2,P1), Leaky ReLU
Hidden Layer	CONV-(N512,K4x4,S2,P1), Leaky ReLU

To address this issue, The paper propose multiscale discriminators. For discriminator networks, we use 70*70 Patch-GAN. The network structure of the discriminator is shown in TABLE 1. After the last layer, the model applies a convolution to produce a 1 dimensional output. The model uses 3 discriminators that have an identical network structure but operate at different image scales. Because the three discriminators learn similarly, to promote the learning of each discriminator, the model introduces a multitask learning strategy based on parameter sharing [20]. The discriminator first extracts the primary features of the images through a shared convolutional layer and obtain the corresponding feature map. Then, the feature samples of the real sample and the generated sample are downsampled using 2 and 4 as sampling factors, respectively, so that three different scale images are obtained. The three discriminators D_1, D_2, and D_3, are also used to process three different scale images. Although the discriminators have the same network architecture, different discriminators can extract different information. A discriminator with a large input scale has a more global perception of the image and can guide the global generation of the image. A discriminator with a smaller input scale is better at guiding the details of the generated image to further improve the overall image, which also makes training the generator easier, since extending a low-resolution model to a higher resolution requires adding an additional discriminator at only the finest level, rather than retraining from scratch. In addition, the introduced multitask learning strategy greatly increases the storage capacity of the discriminator, so that the discriminator has more memory for learning how to discriminate the image and accelerate the convergence of the model. The specific process is as follows:

$$\min_{G} \max_{D_1,D_2,D_3} \sum_{k=1,2,3} \mathcal{L}_{GAN}(G, D_k) \tag{3}$$

where D_k represents one of the three discriminators.

C. OPTIMIZATION ALGORITHM BASED ON A MONTE CARLO SEARCH

We note that such a multiresolution pipeline is a well-established practice in computer vision [21], [22], and a two-scale pipeline is often enough. The experimental results show that although the performance improved, there are still many problems in the generated images. For example, the image generated in the complex scene is not detailed enough; the track lines of generated images have the disadvantages of blurring and ghosting because the generator does not obtain sufficient constraints when learning, resulting in a blurred portion of the generated track line, which cannot be accurately generated.

Thus, this paper introduces the Monte Carlo search technology to the model so that the generator can obtain guidance information rapidly when generating images. By searching the intermediate state of the generator multiple times and then sending the search results to the discriminator to calculate the penalty values, the generator constraints in the generation process are strengthened, and the quality of the generated image is further improved (such as resolution and detail). The search process is shown in FIGURE 2. First, the model uses G to perform a Monte Carlo search on the intermediate state of the generator. The specific process is as follows:

$$\{Y_{t+1}^1, ..., Y_{t+1}^N\} = MC^{G_\beta}(Y_t; N) \tag{4}$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2019.2951860, IEEE Access

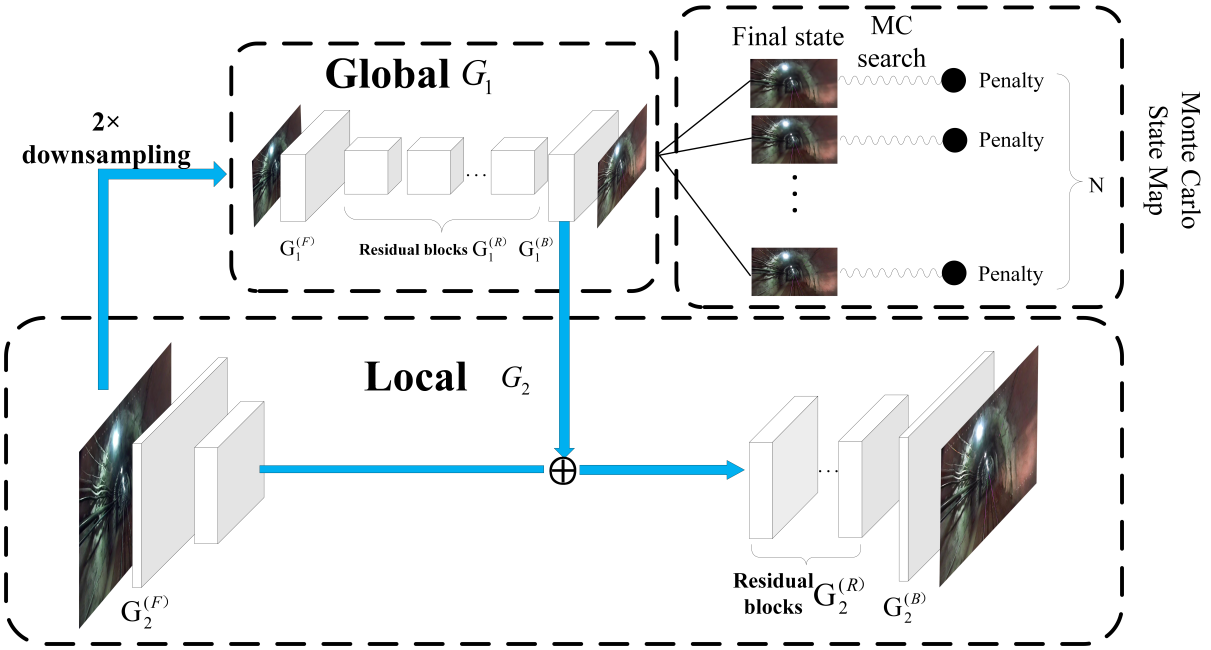

FIGURE 2: Multigranularity generator network structure

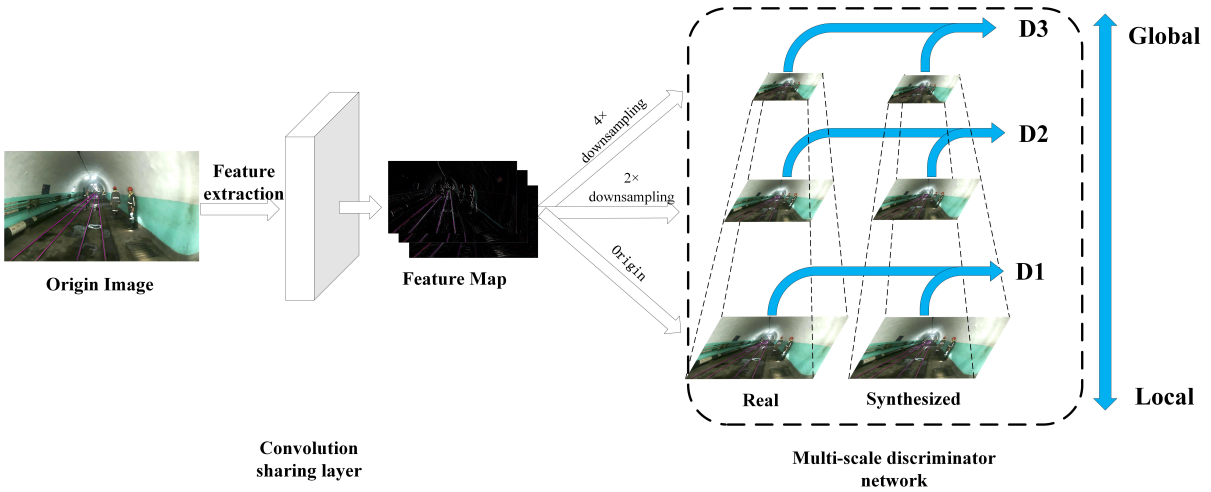

FIGURE 3: Multiscale shared convolution discriminator structure

where N represents the number of searches. MC^{G_β} represents the state of the simulation using the Monte Carlo search. G_β represents another generator virtualized by the Monte Carlo search technology, which has the same parameters as the actual generator. Y_t represents the intermediate state to be sampled. Y_{t+1}^i represents the final state obtained after sampling.

After obtaining the N sampling results, the final states are sent to the discriminator. To avoiding the mode collapse and generate higher quality images, we replace Eq. (1) with Wasserstein GAN objective with gradient penalty [23] defined as:

$$L_{GAN} = E_x[D_k(x,s)] - E_{x,s}[D_k(s,G(x,s)) + D_k(s,Y_{t+1}^i)] - \lambda_{gp} E_{\hat{x},\hat{s}}[(\|\nabla_{\hat{x},\hat{s}} D_k(\hat{x},\hat{s})\|_2 - 1)^2] \tag{5}$$

where D_k represents one of the three discriminators. $\hat{x}$ and $\hat{s}$ are sampled uniformly along a straight line between a pair of real and generated images. All experiments use $\lambda_{gp} = 10$.

To stabilize the training process and generate natural statistics at multiple scales, this paper refers to the perceptual loss in [11]–[13] and introduce a feature matching loss. The model extracts features from different layers of the discriminator and learns to match these intermediate states. Here, $D_k^{(i)}$ is defined to represent the i-th layer feature extractor

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2019.2951860, IEEE Access

of discriminator D_k (from input to the ith layer of D_k). The feature matching loss $\mathcal{L}_{FM}$ is then calculated as:

$$\mathcal{L}_{FM}(G, D_k) = E_{(s,x)} \sum_{i=1}^{T} \frac{1}{N_i} [\left\| D_k^{(i)}(s, x) - D_k^{(i)}(s, G(s)) \right\|_1] \tag{6}$$

where T represents the total number of network layers and N_i represents the number of elements per layer.

Our algorithm uses Monte Carlo search to calculate penalty values and introduce feature matching losses to improve the diversity of model and avoid collapse of mode. The Wasseratein loss is introduced to improve the stability in the training process and accelerate the model convergence. In summary, the final loss function combines both GAN loss and feature matching loss as:

$$\min_G((\max_{D_1,D_2,D_3} \sum_{k=1,2,3} \mathcal{L}_{GAN}(G, D_k)) + \lambda \sum_{k=1,2,3} \mathcal{L}_{FM}(G, D_k)) \tag{7}$$

where λ represents the manually set weighting factor. It is worth noting that in feature matching loss, D_k is used as a feature extractor and does not maximize loss.

IV. EXPERIMENTS PREPARATION

In this section, the datasets and evaluation indicators used in the experiments are explained, followed by the structure of the networks and implementation details.

A. DATASETS

Since there are currently no public datasets containing downhole track lines, we use video cameras fixed on mine locomotives to collect video data. The videos of various collected downhole scenes are divided into frames, and the resolution of the images is uniformly processed to 1280*720. The datasets include different scenarios from multiple mines. To better enhance the performance of the networks, it is also necessary to collect track images under the conditions of corners, multitracks and different illuminations in the downhole scene, which is beneficial for enhancing the generalization performance of the networks.

In actual data processing, finding that even for multiple downhole scenarios, the available datasets are lacking, which is very unfavorable for network training. To solve this problem, we use data enhancement technology to expand the datasets. The specific transformations include image rotation transformation, mirror transformation, flip image transformation and other methods to effectively expand the datasets. Finally we obtained approximately 2,500 images. The training set and the validation set are then divided in an 8:2 manner. That is, the training set is 2000 images, and the test set is 500 images. In terms of dataset labeling, using AutoCAD to mark the track line. To distinguish the track line from the surrounding environment, the color of the label is significantly different from the surrounding environment. The processing of the data set is shown in FIGURE 4. More data can be obtained at https://github.com/LJ2lijia/Downhole-track-line-dataset.

B. METRICS

The experiments use the official indicators on the ground [25], namely, Acc (accuracy), FP (false positive), and FN (false negative), which are defined as follows:

$$Acc = \sum_{im} \frac{C_{im}}{S_{im}} \tag{8}$$

where C_{im} is the number of correct prediction points generated during the test, and S_{im} is the number of ground truths. When the distance between the predicted point and the real point is less than the set threshold (here, the threshold is set to 3), the point is considered correct.

$$FP = \frac{F_{pred}}{N_{pred}} \tag{9}$$

$$FN = \frac{M_{pred}}{N_{gt}} \tag{10}$$

where F_{pred} is the erroneously predicted track line, N_{pred} is the track line that needs to be predicted, M_{pred} is the track line that is mistaken for the ground truth, and N_{gt} is the number of all track lines.

C. TRAINING DETAILS

The experiments in this paper are based on the Ubuntu 16.04, Linux 64-bit operating system, and the GPU is a 1080Ti. All the networks are trained from scratch using the Adam solver and a learning rate of 0.05. We keep the same learning rate for the first 100 epochs and linearly decay the rate to zero over the next 100 epochs. Weights are initialized from a Gaussian distribution with a mean of 0 and a standard deviation of 0.02. The number of Monte Carlo searches N is set to 5, and the specific gravity λ between the control feature matching loss and the discriminator loss function is set to 10. The implementation of model is based on the PyTorch[1] deep learning framework.

V. EXPERIMENTS

This section presents the experimental results and related analysis. The algorithm evaluates and analyzes the proposed model from both objective and subjective aspects.

A. AUTOMATIC EVALUATION

To quantify the results of the experiments, we perform semantic segmentation on the generated images and compare the degree of matching between the predicted segments and the input images. The principle involved is that if our model can generate a real image corresponding to the input label mapping, then the existing semantic segmentation model should be able to predict the real-world label of the ground. Among the models involved in the comparative experiment are pix2pix [9], cascaded refinement networks (CRN) [10]. The experimental results are shown in TABLE 1. The IOU in the table represents the intersection-over-union. To avoid the contingency of the experiment, the experiment was repeated

[1]https://pytorch.org/

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2019.2951860, IEEE Access

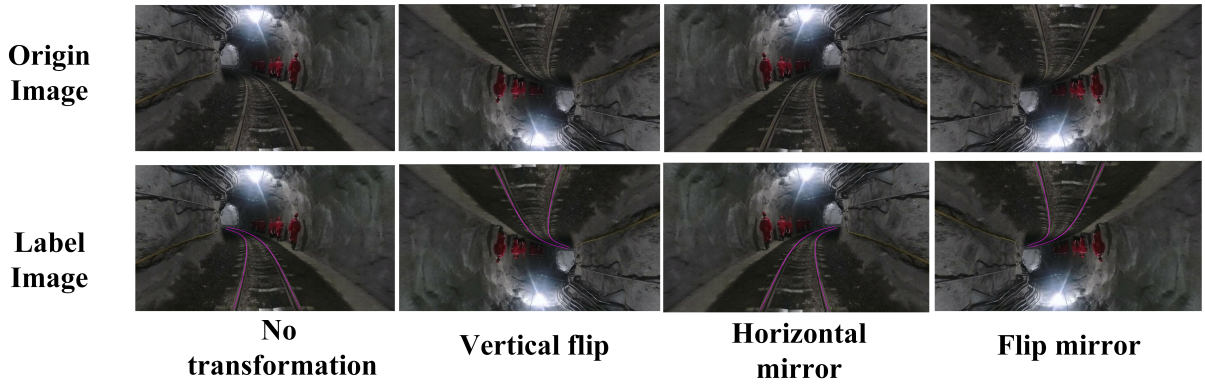

FIGURE 4: The process of data enhancement

several times, and the data in the table are the average of the experimental data.

TABLE 2: Automatic evaluation with image translation models

	pix2pix	CRN	Ours	Oracle
Pixel accuracy	77.56	70.13	**82.43**	83.25
IOU	0.3846	0.3375	**0.6218**	0.6763

As shown in TABLE 2, the model proposed in this paper obviously exceeds the previously existing models for this type of problem, both in terms of pixel precision and the IOU. The results of our model are closer to the original image, which proves the superiority of the algorithm in this paper.

To compare the model proposed in this paper with the existing state-of-the-arts lane detection algorithm, we transplant the lane detection algorithm to the underground for detecting the track line. The models involved in the experiment are pix2pix [9], Spatial CNN (SCNN) [4], LaneNet [5], and Segmentally Switchable Curves (SSC) [24].

The experimental result is shown in TABLE 3. From the experimental result, we can see that proposed model (Ours) outperforms all others methods, including SCNN and LaneNet. The accuracy achieved by paper's model is promisingly high, indicating that the framework with multi-granularity generator and multi-scale discriminator can detect the track lines in the images comprehensively and delicately.

However, our model does not achieve the best performance in inference speed, only second to SCNN and LaneNet models. This is mainly because multi-granularity generators introduce more computation. In fact, 22 FPS is fast enough for downhole track detection.

B. MANUAL EVALUATION

To further evaluate the proposed model, we adopts the method of manual evaluation. The existing platform for manual evaluation is MTurk[2] (Amazon Mechanical Turk), so we use a similar method to publish the results of the experiment online to a website and send them to volunteers through social platforms.

[2]https://www.mturk.com/

TABLE 3: Automatic evaluation with the lane detection algorithm on the ground

Method	Accuracy(%)	FP	FN	Inference Speed (FPS)
SCNN	93.26	0.0598	0.0269	24
LaneNet	92.87	0.0620	0.0312	**26**
SSC	89.64	0.0643	0.0393	18
pix2pix	90.89	0.0535	0.0297	19
Ours	**95.01**	**0.0401**	**0.0186**	22

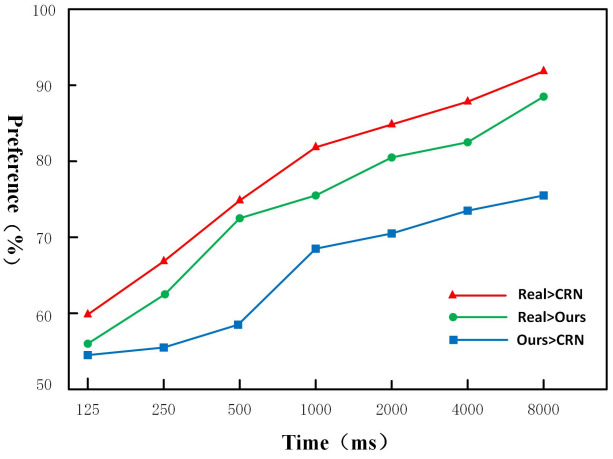

FIGURE 5: Preference-time fluctuation graph

For this task, based on the same input image, using the CRN model and the model proposed in this paper to generate two images and participate in the comparison with the real image. To better reflect fairness, we sent two of the three images to volunteers almost simultaneously. The volunteers are required to select the most accurate and texture-clear images within a limited time. The limited time is from 125 ms to 8000 ms. The comparison results are shown in FIGURE 5.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2019.2951860, IEEE Access

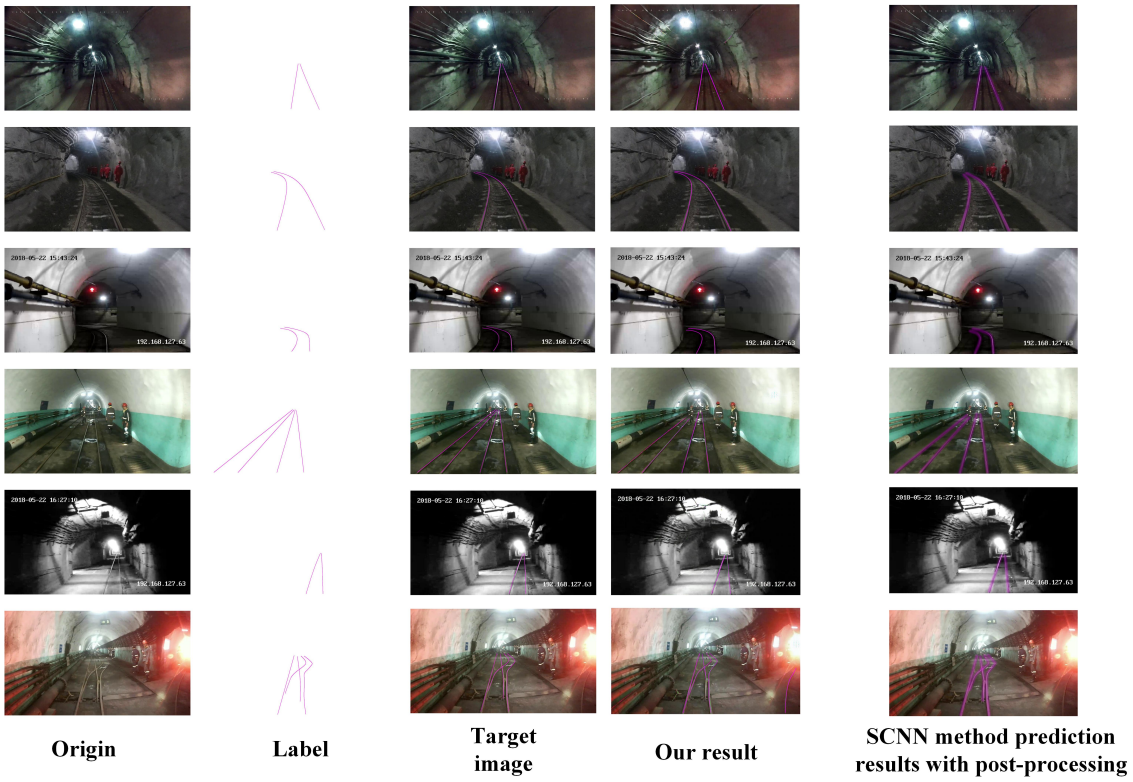

FIGURE 6: Comparison of our model and SCNN network test results

It can be seen from FIGURE 5 that as the limited time increases, the difference between the three images becomes more apparent. The final result shows that the model in this paper is obviously better than the CRN model, and the gap with the real images becomes increasingly smaller.

C. ABLATION ANALYSIS

To verify the validity of the Monte Carlo search, comparing the proposed model with the model without the Monte Carlo search and explore the impact of the number of searches on the Monte Carlo search on the performance of the model. Where Without MC represents the model without the Monte Carlo search, and N represents the number of Monte Carlo searches.

TABLE 4: Ablation analysis for proving the effectiveness of Monte Carlo

Method	Accuracy(%)	FP	FN	Average Time(s)
Without MC	91.25	0.1031	0.1002	0.2567
N=1	92.87	0.0901	0.0912	0.2678
N=3	93.09	0.0765	0.0703	0.2806
N=5	95.01	0.0401	0.0186	0.2962
N=7	95.68	0.0399	0.0176	0.3465
N=9	95.96	0.0365	0.0170	0.4031

The average time in TABLE 4 is the time when an image is generated during training. It can be seen from the experimental results that the introduction of the Monte Carlo search obviously significantly improves the accuracy, FP and FN of the final generated results. We found that with the increase in the number of searches, when N=5, both time-consumption and accuracy achieve better results. As N continues to increase, the accuracy, FP, and FN improve, but the increase is not large, and the average generation time of each image during training increased considerably because as the number of searches increases, the amout of calculations required to generate each image gradually increases, and the increase is nonlinear, so the number of Monte Carlo searches is set to 5 by default.

To verify the effect of multiscale discriminator and multi-task learning, we conduct a comparative experiment on the model proposed in this paper, the model with a multiscale discriminator network but no shared weights and the model using only a single-scale discriminator. The generator and loss functions are fixed during this experiment.

The experimental results are shown in TABLE 5. To avoid the contingency of the experiment, the experiment was repeated several times, and the results in the table are the average of the experimental results. Where Single D donets the model using only a single-scale discriminator and Multiscale Ds donets the model with a multiscale discriminator network but no shared weights.

From TABLE 5, the results find that multiscale discriminator and multitask learning strategy can significantly im-

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2019.2951860, IEEE Access

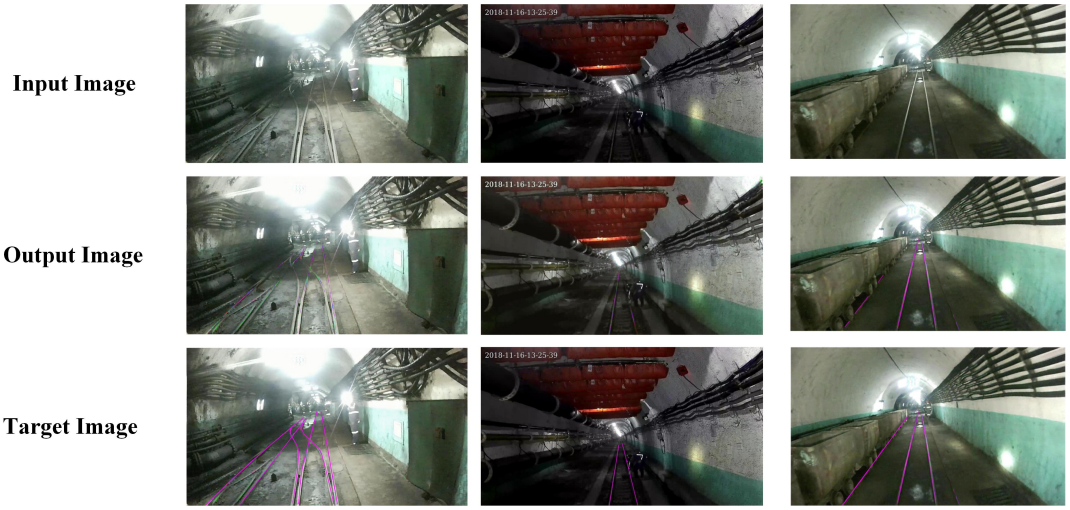

FIGURE 7: The test result of our model in the complex scenes

prove the pixel accuracy because the multiscale discriminator makes the judgment of the generated pixel points stricter during the network training. The multitasking learning strategy allows the discriminator to learn more features, giving the generator more accurate guidance.

TABLE 5: Ablation analysis for proving the effectiveness of multiscale discriminator and shared weights

	Single D	Multiscale Ds	Ours
Pixel accuracy	80.08	81.43	**82.68**
IOU	0.5125	0.5818	**0.6351**

D. CASE STUDY AND ERROR ANALYSIS

TABLE 6: IOU for different methods in specific conditions

Method	Single track	Multi-track	Curved track	Weak illumination
SCNN	0.5126	0.4983	0.5047	0.4835
LaneNet	0.4657	0.4528	0.4649	0.4476
SSC	0.3841	0.3487	0.3754	0.3564
pix2pix	0.3954	0.3689	0.3876	0.3701
Ours	**0.6295**	**0.6178**	**0.6267**	**0.6132**

The paper selected images of four specific scenes from the test set and compared them with state-of-the-art lane detection models. The experimental results are shown in TABLE 6. Experimental results show that the performance of proposed model is better than other models in four scenarios.It is worth noting that the previous models can not adapt to complex scenarios such as multi-track or curve and the performance of the models declines. However, the model has excellent detection accuracy in various scenarios, which also proves that paper's model has strong robustness.

To reflect the difference between the proposed model and the traditional CNN detection results, FIGURE 6 shows some of the test cases in the downhole roadway scenario. It can be seen from the experimental results that the images generated by our model are very detailed, and high-quality images can be generated for all the above scenarios. The proposed model can still accurately detect the tracks that are not marked in the training images, which fully demonstrates that the model has excellent robustness. The application of SCNN to the downhole scene also obtains good recognition results, but the robustness of the model is poor, and it cannot detect unmarked track lines. In addition, similar to the traditional models based on the convolutional neural network, the SCNN algorithm requires more postprocessing techniques, which improves the complexity of the visualization operation, and the detected results are not real. It can be seen from the above comparison experiments that paper's model has great advantages.

However, in the experiment, it is also found that if the scene in the image is very complicated (for example, more than four track lines and track lines are occluded), the result of the proposed model will be affected. The experimental results are shown in FIGURE 7. Because the track line recognition in complex scenes requires more adequate and accurate guidance information, but the repetitive search method for intermediate states does not effectively constrain the generation of generators for some complex scenes. Therefore, the future work is to achieve automated detection of track lines in complex scenarios.

VI. CONCLUSION AND FUTURE WORK

This paper proposes a downhole track line detection model based on a multi-scale conditional adversarial generation network. The model realizes the generation of realistic downhole track line detection images by decomposing the generator into global and local generators. The proposed penalty

mechanism based on Monte Carlo search helps the generator to focus more on perfecting the semantic structure of the images. Extensive qualitative and quantitative evaluations testify the effectiveness of proposed model. This paper provides a new idea for track line detection. In the future, the model can also be flexibly migrated to the image generation field, such as image translation. Using more complex images provided significant improvements in visual quality and added more details to synthesized images. The challenging nature of the problem leaves room for further improvements.

REFERENCES

[1] Cong Hoang Quach, Van Lien Tran, Duy Hung Nguyen, Viet Thang Nguyen, Minh Trien Pham, Manh Duong Phung. "Real-time lane marker detection using template matching with RGB-D camera." 2018 2nd International Conference on Recent Advances in Signal Processing, Telecommunications and Computing (SigTelCom). IEEE, 2018: 152-157.

[2] LI, Junyang, Lizuo JIN, and Shumin FEI. "Urban road detection based on multi-scale feature representation." Journal of Electronics and Information Technology 36.11 (2014): 2578-2585.

[3] Bente, Tamás Ferencz, Szilvia Szeghalmy, and Attila Fazekas. "Detection of lanes and traffic signs painted on road using on-board camera." 2018 IEEE International Conference on Future IoT Technologies (Future IoT). IEEE, 2018: 1-7.

[4] Xingang Pan, Jianping Shi, Ping Luo, Xiaogang Wang, Xiaoou Tang. "Spatial as deep: Spatial cnn for traffic scene understanding." Thirty-Second AAAI Conference on Artificial Intelligence. 2018.

[5] Davy Neven, Bert De Brabandere, Stamatios Georgoulis, Marc Proesmans, and Luc Van Gool. "Towards end-to-end lane detection: an instance segmentation approach." 2018 IEEE Intelligent Vehicles Symposium (IV). IEEE, 2018: 286-291.

[6] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville and Yoshua Bengio. "Generative adversarial nets." Advances in neural information processing systems. 2014: 2672-2680.

[7] Mirza, Mehdi, and Simon Osindero. "Conditional generative adversarial nets." arXiv preprint arXiv:1411.1784 (2014).

[8] Zhang, He, Vishwanath Sindagi, and Vishal M. Patel. "Image de-raining using a conditional generative adversarial network." IEEE Transactions on Circuits and Systems for Video Technology (2019).

[9] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, Alexei A. Efros. "Image-to-image translation with conditional adversarial networks." Proceedings of the IEEE conference on computer vision and pattern recognition. 2017.

[10] Chen, Qifeng, and Vladlen Koltun. "Photographic image synthesis with cascaded refinement networks." Proceedings of the IEEE International Conference on Computer Vision. 2017.

[11] Dosovitskiy, Alexey, and Thomas Brox. "Generating images with perceptual similarity metrics based on deep networks." Advances in neural information processing systems. 2016.

[12] Yongcheng Jing, Yezhou Yang, Zunlei Feng, Jingwen Ye, Yizhou Yu and Mingli Song. "Neural style transfer: A review." IEEE transactions on visualization and computer graphics (2019).

[13] Johnson, Justin, Alexandre Alahi, and Li Fei-Fei. "Perceptual losses for real-time style transfer and super-resolution." European conference on computer vision. Springer, Cham, 2016: 694-711.

[14] Florian Maire, Nial Friel, Antonietta Mira and Adrian E. Raftery. "Adaptive Incremental Mixture Markov chain Monte Carlo." Journal of Computational and Graphical Statistics (2019): 1-15.

[15] Lantao Yu, Weinan Zhang, Jun Wang, Yong Yu. "Seqgan: Sequence generative adversarial nets with policy gradient." Thirty-First AAAI Conference on Artificial Intelligence. 2017.

[16] Wang, Ke, and Xiaojun Wan. "SentiGAN: Generating Sentimental Texts via Mixture Adversarial Networks." IJCAI. 2018: 4446-4452.

[17] Sajjadi, Mehdi SM, Bernhard Scholkopf, and Michael Hirsch. "Enhancenet: Single image super-resolution through automated texture synthesis." Proceedings of the IEEE International Conference on Computer Vision. 2017: 4491-4500.

[18] Dosovitskiy, Alexey, and Thomas Brox. "Generating images with perceptual similarity metrics based on deep networks." Advances in neural information processing systems. 2016: 658-666.

[19] Alexander Sasha Vezhnevets, Simon Osindero, Tom Schaul, Nicolas Heess, Max Jaderberg, David Silver and Koray Kavukcuoglu. "Feudal networks for hierarchical reinforcement learning." Proceedings of the 34th International Conference on Machine Learning-Volume 70. JMLR. org, 2017: 3540-3549.

[20] Liu, Shikun, Edward Johns, and Andrew J. Davison. "End-to-end multi-task learning with attention." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2019: 1871-1880.

[21] Hao Dong, Simiao Yu, Chao Wu, Yike Guo. "Semantic image synthesis via adversarial learning." Proceedings of the IEEE International Conference on Computer Vision. 2017: 5706-5714.

[22] Guibas, John T., Tejpal S. Virdi, and Peter S. Li. "Synthetic medical images from dual generative adversarial networks." arXiv preprint arXiv:1709.01872 (2017).

[23] Arjovsky, Martin, Soumith Chintala, and Léon Bottou. "Wasserstein generative adversarial networks." International conference on machine learning. 2017: 214-223.

[24] Guo B; Dong Y. "Railway track detection algorithm based on piecewise curve model." Journal of Railway Science and Engineering, 2016.

XING WEI Associate professor at Hefei University of Technology. His research interests include deep learning and Internet of things engineering, driverless solutions and so on.Corresponding author of this paper.

JIA LI 2016 undergraduate student, School of Computer and Information, Hefei University of Technology.The main research direction is natural language processing and emotional dialogue generation.

GUOQIANG YANG Yang Guoqiang, male, master student, the main research direction is image processing and computer vision.

YANG LU was born in 1967, received his Ph.D. degree from HefeiUniversity of Technology in 2002. Now he is a professor and a doctoralsupervisor in Hefei University of Technology. His main research interestsinclude IOT (Internet of Things) engineering and distributed control system.