Exploration of Vision-based Railway Turnout Recognition and Application

Chenglin Chen
Thrust of Intelligent Transportation
The Hong Kong University of Science and Technology (Guangzhou)
Guangzhou, China
cchen363@connect.hkust-gz.edu.cn

Huixiong Qin
Center for Advanced Infrastructure and Transportation
Rutgers University
New Brunswick, United States
hq63@soe.rutgers.edu

Yun Bai
Thrust of Intelligent Transportation
The Hong Kong University of Science and Technology (Guangzhou)
Guangzhou, China
yunbai@hkust-gz.edu.cn

***Abstract*—Railway turnouts are crucial components of railroad transportation, responsible for changing the direction of trains. Turnouts can aid in train positioning systems based on railway geographical information by accurately recognizing and classifying turnouts and predict the path of the train in advance, thereby enhancing the safety and efficiency of railway transit. However, the present turnout recognition algorithms perform poorly in real-world applications since there are few turnout-related datasets and the features between different classes are highly similar. This work systematically explores the problems that turnout recognition may face and corresponding improvement solutions. We customized a dataset and based on this, we conduct extensive experiments to explore the impact of factors such as data augmentation, image resolution, and training size on turnout recognition performance. The best solution can achieve 89.03% Top-1 accuracy and 93 FPS speed, enabling high accuracy while achieving real-time performance. Our work provides great promise for improving the performance of train environment perception and positioning systems and has the potential to be widely used in real-world rail transit.**

Keywords—railway, turnout recognition, computer-aided positioning, image classification, computer vision

I. INTRODUCTION

In contemporary society, railway transportation is widely used around the world as an efficient and reliable public transportation. The timely and precise acquisition of train location information is crucial to ensuring the safety and efficient operation of trains. Currently, one of the most mainstream train positioning technologies worldwide is positioning based on the Global Positioning System (GPS) [1]. The GPS receiver carried on trains can calculate the train's position by receiving satellite signals. The positioning accuracy of GPS typically ranges between 5 to 15 meters. In comparison to other modes of transportation, such as cars or airplanes, trains usually travel on relatively fixed paths. Trains travel on tracks, which are planned, laid and fixed in advance, thereby constraining the driving path of the train within the rail network. This characteristic renders the movement of trains relatively predictable. Therefore, more accurate train position information can be obtained by fusing the train position obtained by GPS with the track geographical information [2]. In other words, the coordinates obtained from the GPS can be mapped onto the track of the geographical map, thereby enabling the refinement of the train position. This method provides high accuracy in most cases but fails in turnout areas. There are often multiple paths near the turnout and current GPS often struggle to precisely determine the exact track on which a train is located in turnout regions.

In railway networks, turnouts, as crucial devices determine the direction of train travel, need to be accurately detected and classified to ensure the precision of train positioning. Through precise recognition of turnouts, we can know which path the train will follow, thereby providing crucial auxiliary information for GPS-based train positioning. Track circuits and vibration sensors are two commonly used for turnout recognition. When a train passes the turnout, the current in the track circuit changes. Likewise, vibration sensors mounted on tracks or turnouts can detect vibrations caused by a passing train. By monitoring these changes, we can determine the turnout direction. Some researchers have made several attempts. [3] detected the train type by accelerometer sensors placed around the turnouts and crossings. However, these methods are not suitable for non-electrical turnouts and can only detect signals when the train passes, lacking the ability for advance prediction. Additionally, the maintenance costs for these sensor facilities are high. With the rapid development of deep learning (DL) and computer vision technology, scholars have widely customized vision-based algorithms for the railway industry. For instance, [4] utilized semantics segmentation to detect trespassing incidents along the right-of-way, [5] monitored grade-crossing violations via object detection and object tracking, and [6] predicted track geometry degradation using DL-based time-series analysis. However, the safety-centric nature of traditional railway industry data is relatively closed, leading to a limited amount of research on turnout detection or classification.

The work [7] defined the maximum lateral deviation between the center of the current track and the center of the train. It first extracts the track through the edge detector and sets the track tracker, and then determines the turnout direction by the position of the current track center point. Similar work [8] was also carried out by detecting rail turnout through image processing on railway line. They use canny edge extraction and Hough transform to get the track line. The turnout crossing zone is determined by the intersection of the two track lines. Nevertheless, this method requires hand-crafted hyper-parameters and is limited to processing scenes with relatively simple backgrounds and exhibits poor robustness when confronted with complex environments. [9] constructed a semantic railway scene understanding dataset collected from on-board cameras from different countries. The image classification and detection tasks it builds include the switch category. They use densenet161 [10] pre-trained on ImageNet for fine-tuning to obtain better classification performance. The possibility of transfer learning in improving the accuracy of small dataset-based turnout detection was verified in the work of [11]. Reference [12] explored the effect of bounding box size on turnout detection performance by introducing three hyper-parameters over the boxes.

The first and third authors were funded by Guangzhou Municipal Science and Technology Project under Grant 2023A03J0011, excluding second author.

979-8-3503-4929-0/24/$31.00 ©2024 IEEE

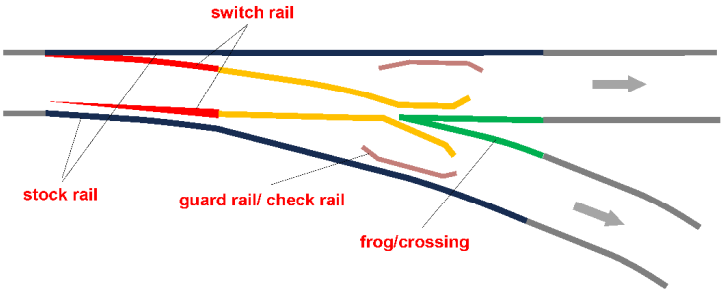

Fig. 1. The components of turnout.

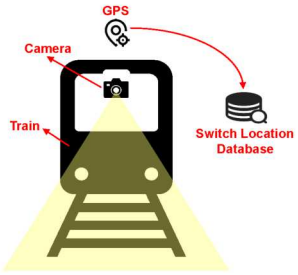

Fig. 2. Data acquisition.

The core concept behind vision-based turnout recognition is to find the feature relationships of different components within the turnout area (e.g., guide rails, switch blades, diamond crossovers, etc.) to determine their driving direction. Fig. 1 illustrates the components of the turnout. However, due to the minimal differences among different types of turnouts and the relative scarcity of turnout-related data, existing turnout detection algorithms exhibit poor performance in terms of accuracy and robustness. It may be possible to improve its accuracy by acquiring high-resolution images and using strategies such as data augmentation. However, to the best of our knowledge, there is currently no literature that has studied these in depth in the field of railway turnout recognition. Against this backdrop, this paper extensively explores the critical issues in turnout recognition and conducts in-depth analyses of a series of possible optimization solutions. Our goal is to provide new perspectives and innovative solutions for the development of GPS-based train positioning systems. The main contributions of this paper are as follows:

1) We conduct a comprehensive analysis of the challenges encountered in vision-based turnout recognition tasks and proposed a series of possible solutions.

2) We explore the effect of pre-processing of turnout recognition in terms of data augmentation, the choice of resolutions and scaling strategy on training, providing some empirical prior knowledge for the turnout recognition task.

3) We further explore how to deploy turnout recognition approaches on CPU platforms and how turnout recognition can assist train GPS positioning.

II. Case Study

A. Algorithm Customization

In recent years, the booming of deep learning has propelled the advancement of intelligence in different industries. Nevertheless, despite the significant success of these advanced algorithms in many industries, their application is somewhat restricted in safety-centric closed rail transit industries. Deep learning methods usually rely on extensive high-quality datasets for training. Unfortunately, the rail transit sector faces challenges in development of smart algorithms due to the limited availability of public datasets. In this paper, we focus on vision-based turnout recognition in the field of rail transit. We construct a customized turnout dataset and use it as the foundation for our study and discussions. The challenges that vision-based turnout recognition algorithms confront will then be discussed in detail, along with a few possible solutions.

Data Acquisition. The turnout data is captured by onboard forward cameras on the train. We have knowledge of the geographic location information of the turnouts. When the position of the train obtained by GPS is near the turnout, the image sequence at this time is collected. See Fig. 2. To ensure data diversity and explore the impact of image resolution on the images, we conducted data collection on two lines using cameras with resolutions of 640×360 and 1920×1024 respectively.

Challenge 1: Insufficient Training Data. Turnouts in the field of rail transit are considered special structures. Compared to straight tracks, the number of turnouts is relatively small, and the data collection process is more complex. In addition, the difference between adjacent frames in the video is small and can be ignored. Generally, one frame is selected at a fixed interval, thereby reducing the data redundancy but also shrinking the overall samples. These factors jointly lead to insufficient training data, limiting the performance of data-driven turnout recognition algorithms.

Challenge 2: Low Data Quality. The quality of turnout images collected by cameras may be affected by many factors. Firstly, the camera is sensitive to lighting changes, and the image quality is poor in strong light or dark light conditions. Secondly, cameras may be subject to interference from train vibrations and high-speed motion, leading to motion blur and distortion. Additionally, adverse weather conditions such as rain, snow, and strong winds can also affect image clarity. These factors collectively result in issues such as noise, blurring, and distortion in the turnout images captured by the camera, posing challenges to the accuracy and stability of turnout recognition algorithms.

Challenge 3: Low Inter-class Variance. The position of the switch rail in turnout plays a decisive role in the direction of track traversal. From a geometric perspective, trains typically travel in the direction where the gap between the switch rail and the stock rail is larger. There is a high similarity in geometric features between left and right switch. In the case of double locomotives, the train usually does not need to turn around when returning, but directly uses the locomotive on the other side to travel. However, in images captured in reverse travel, the position of the switch rail, which originally determines the direction, changes. This can lead to confusion

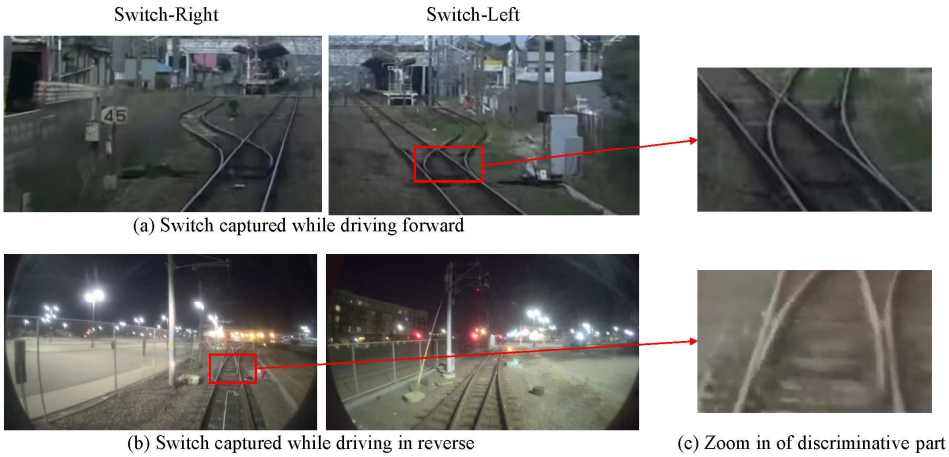

Fig. 3. Visualization of partial turnout images.

between the left and right switch when compared to images captured in forward travel, significantly increasing the difficulty of turnout identification. We displayed the turnout images with high similarity, as shown in Fig. 3. In order to observe the similarity of different turnout classes more intuitively, we use t-distributed Stochastic Neighbor Embedding (t-SNE) [13] to reduce the dimensionality of the original training image data to a 2-dimensional space, and reflect the similarity relationships through the relative positions of different turnout classes. Fig. 4 shows the tSNE distribution of the training data. As you can see, there are no clear boundaries between different classes.

Challenge 4: Data Imbalance [14]**.** Critical turnout components, such as switch tongues and switch points, often occupy relatively small areas in the image, with the majority of the image containing background information such as tracks and rails. This imbalance may lead the model to overly focus on the background, resulting in suboptimal performance in recognizing key turnout components. Additionally, certain turnout categories may appear less frequently in real-world scenarios, exacerbating the problem of imbalanced class occurrences. The above are some challenges of vision-based turnout recognition algorithms. We also have some strategies to deal with these issues, which are listed below.

S1: Data Augmentation [15]. Utilizing data augmentation techniques such as geometric or pixel transformations to expand the dataset can enhance the model's ability to generalize across different scenarios. However, the effectiveness of specific augmentation strategies remains to be further investigated.

S2: Transfer Learning [16]. Leveraging models pre-trained on large-scale datasets and fine-tuning them for turnout recognition tasks can accelerate model convergence and enhance performance.

S3: Oversampling and Under-sampling. Oversampling examples in the minority class and under-sampling examples in the majority class can be adopted to alleviate data imbalance in training to ensure that the model can better learn features from all classes.

S4: Class Weights. By assigning different weights to different classes, the model can focus more on classes with few samples, thereby improving the ability to recognize classes with few samples.

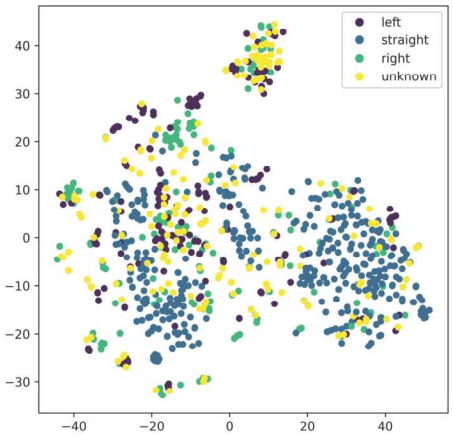

Fig. 4. t-SNE visualization of training data.

S5: Ensemble Learning. The robustness of the system can be enhanced by integrating multiple models with different structures and combining their predictions through voting or weighting. However, it's worth noting that this strategy may not be suitable for scenarios with real-time requirements.

S6: Optimizing Architecture. By adjusting the neural network structure and loss function, the model can be designed to better meet the specific requirements of the turnout recognition task.

The performance of vision-based turnout recognition algorithms may be improved by utilizing these tactics individually or in combination. Researchers have already investigated a few tactics, like transfer learning. In this work, we focus on the exploration of preprocessing, including data augmentation, the selection of training size, etc.

B. Model Application

In this study, the ultimate goal of turnout recognition is aimed at assisting train GPS positioning. Thus, we further elucidate how turnout recognition approach supports train localization, as detailed in Algorithm 1. Initially, the current segment is a polyline between the start point and the first turnout. During the positioning process, it projects the current

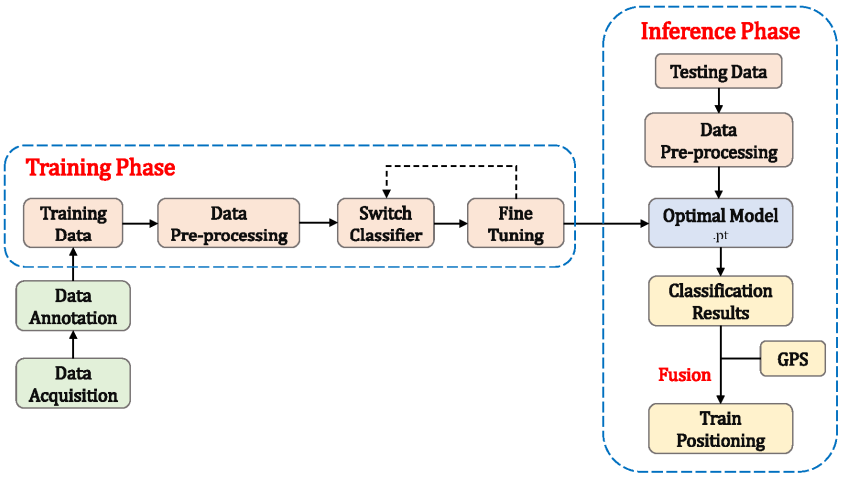

Fig. 5. Workflow of turnout recognition system.

GPS location to the current segment. When it comes to turnout, it relies on the pose prediction to update the current segment. Fig. 6 illustrates the railway track geographical database and shows the position revision. In this case, by combining the results of turnout recognition, we can enhance the positioning accuracy to the track where the train is located.

Algorithm 1: Turnout Classification-aided GPS Positioning of Train

Input: The video frame at time $t - I_t$, GPS location at time $t - \hat{Y}_t$, railway track geographical database D.
Output: The train's location at time $t - Y_t$.

$S \leftarrow \{p_1, p_2, \dots, p_{n0}\}$ the polyline between the first point and the first turnout
initiate the current segment
for $t = 1, 2, \dots$ **do**
  if $dist(\hat{Y}_t, nextTurnout) < C$ **then**
    $X_t \leftarrow turnout\_detection(I_t)$
    $S \leftarrow \{p_1, p_2, \dots, p_{n0}\}$ // the current segment slides to the next segment based on X_t and D.
  // project the GPS location to the current segment
  min $\_dist \leftarrow \infty, Y_t \leftarrow null$
  for $t = 1, 2, \dots, n_t - 1$ **do**
    $dist, pt = project\_point\_to\_line(\hat{Y}_t, l(p_i, p_{i+1}))$
  If $dist < minDist$ **then**
    $Y_t \leftarrow pt, minDist \leftarrow dist$
  yield Y_t

III. Experiments and Discussion

Experiments were carried out using a customized dataset in order to validate some of the tactics that were presented in the preceding subsection that can improve the performance of vision-based turnout recognition. In this section, we primarily focus on the impact of data preprocessing on the turnout classification algorithm, including four aspects: 1) comparing the performance of common deep learning-based classification algorithms on turnout classification tasks; 2) analyzing the effect of different data augmentation methods on it; 3) studying the effects of image resolution, model input size, and methods of image scaling; 4) exploring the model deployment and how turnout recognition can assist train positioning.

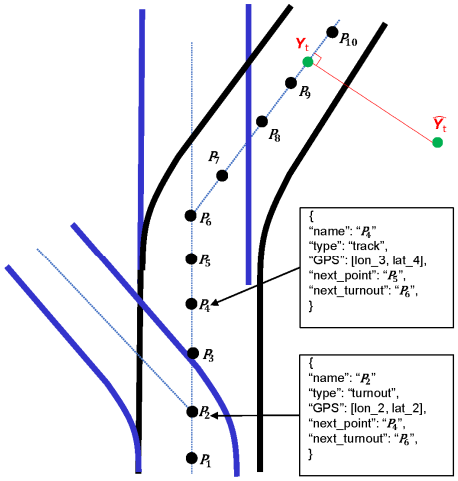

Fig. 6. Railway track geographical database, where p_i is the nodes in the track polyline, $\hat{Y}_t$ is the GPS position, Y_t is the revised position. The black track is the one that will be occupied by the train while the blue track will not be occupied. The polyline consists of two types of nodes: turnouts and tracks, each annotated with geographical coordinates indicating their neighbors. Additionally, the track node maintains a reference to its adjacent switch.

A. Experiment Setup

Dataset. We use a customized railway turnout dataset consisting of 640×360 and 1920×1024 resolution images. The dataset images were taken in day-to-night environments, including four categories: left, right, straight, and unknown. The entire annotation is only image level labels. We use an 8:2 ratio to divide the training set and the validation set.

Metrics. We mainly evaluate the performance of turnout classification from two aspects: accuracy and speed. We measure accuracy using Top-1 accuracy (Top-1 Acc) and runtime using frame per second (FPS). Top-1 Acc is calculated as the true predicted samples divided by the total samples. If the classifier returns the category with the highest probability scores equal to the ground truth, the prediction is correct, otherwise the prediction is wrong. Parameters and FLOPs are used to measure the model complexity.

Implementation Detail. All model training in this work was done on Ubuntu 20.04 with a NVIDIA GeForce GTX 3060 Ti and CUDA 11.7. Python 3.8 was chosen as the

TABLE I. THE DETAILS OF CUSTOMIZED DATASET

Image Resolution	All	Left	Right	Straight	Unknown
640×360	443	128	104	101	110
1920×1080	733	110	125	379	119
total	1176	238	229	480	229

TABLE II. THE CONFIGURATION OF TRAINING HYPER-PARAMETERS

Hyper-Parameters	Setting
Epoch	40
Batch size	16
Learning rate	0.0001
Optimizer	Adam
Adam betas	(0.9, 0.999)
StepLR	Step_size=10, gamma=0.1
Loss function	Cross Entropy

TABLE III. THE COMPARISON RESULTS OF COMMON CLASSIFICATION METHODS ON CUSTOMIZED DATASET

Model	FLOPs	Params	FPS	Top-1 Acc
VGG16	78.70G	134.3M	41	69.62%
MobileNetV2	0.45G	0.67M	91	69.20%
ResNet18	8.54G	11.18M	96	83.54%
ResNet18-SE	8.54G	11.27M	93	**85.65%**
DenseNet121	12.93G	6.87M	72	84.39%
EfficientNet_B0	1.82G	3.97M	88	81.01%
EfficientVit_B0	0.49G	2.13M	90	81.86%
VitTiny_Patch16	4.82G	5.49M	85	81.43%
RepVGG_A0	7.08G	7.81M	92	85.23%

programming language, and PyTorch 1.13.1 served as the deep learning framework. For detailed hyper-parameters information during the training phase, see Table II.

B. Baseline Selection

The overall workflow of the vision-based turnout recognition system is depicted in Fig. 5. Images are initially captured using an onboard camera, followed by data cleaning and annotation. Secondly, various data augmentation strategies are applied to expand the dataset, which is then fed into an end-to-end turnout classification model for training. Iterative training is performed by adjusting the hyper-parameters to obtain the optimal model. Afterwards, the trained model is deployed to the on-board edge device with limited resources, and certain acceleration tools are employed for model optimization. In this section, our emphasis is on selecting the baseline for vision-based turnout recognition tasks. When it comes to turnout recognition tasks, they can be implemented through detection or classification. However, in this paper, we opt for classification rather than detection. This decision is grounded in three reasons. First of all, our objective is to assist train positioning by identifying the status of the turnouts, without requiring its location in the image. Secondly, detection is more challenging to annotate compared to classification tasks. Turnout annotation typically requires knowledge and expertise in rail transit, and its accuracy is

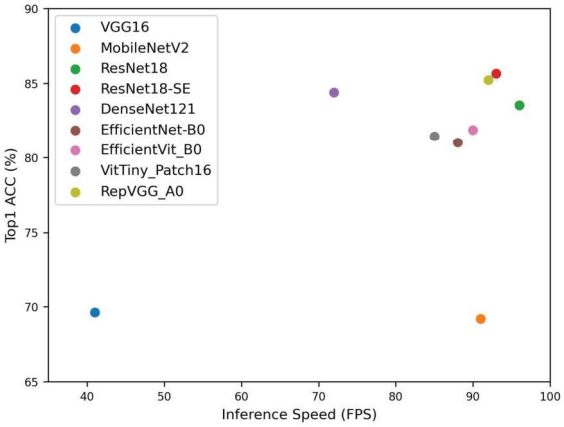

Fig. 7. Visualization of classification comparison results.

pivotal for model performance. Moreover, detection tends to more complex than classification as it involves not only object classification but also bounding box regression. To sum up the above points, we decided to use classification task to realize turnout recognition.

At present, there are many classification algorithms that have achieved relatively good performance. Here, we experiment with some common learning-based classification models, including VGG16 [17], ResNet18 [18], ResNet18-SE [19], MobileNetV2 [20], DenseNet121 [10], EfficientNetB1 [21], EfficientVit-B0 [22], VitTiny-Patch16 [23], and RepVGG-A0 [24]. We conduct experiments on a custom dataset and no data augmentation is adopted. All images are uniformly scaled to 640×360 through Resize before being fed into the model. We apply the same set of hyperparameters shown in Table II. Our preference is to employ simpler models to achieve a higher accuracy in turnout classification. The experimental results are shown in Table III. It can be seen that VGG16 has the slowest inference time, and the highest number of parameters yet achieves only 69.62% Top-1 Acc. MobileNetV2, having the fewest parameters, also exhibits relatively low accuracy at 69.20%. ResNet18-SE, a variant of ResNet18 incorporating SENet [19], achieves the highest Top-1 accuracy at 85.65%. RepVGG-A0 achieves an accuracy of 85.23%, second only to ResNet18-SE.EfficientNet-B0, EfficientVit-B0, and VitTinyPatch16 have fewer parameters, leading to a certain degree of accuracy decline. However, they still achieve a Top-1 Acc of approximately 81%. Based on the experimental results, we believe that real-time turnout classification with high accuracy can be attained with ResNet18-SE, which will be utilized in further experiments.

C. Effect of Data Augmentation

By adding variances and disturbances through data augmentation, one can expand the data distribution and improve the generalization ability and robustness of the model. Common data enhancements can be divided into geometric transformation, color space transformation, pixel-level transformation, random erasing, as well as mixing methods. However, not all data augmentation can improve the accuracy of the model. Model accuracy could be negatively impacted by improper data augmentation, which could introduce huge noise. What's more, the computational cost of training goes up with massive data augmentation. As a result, before applying data augmentation techniques, it is essential to thoroughly consider their suitability for the task at hand and

the dataset. The applicability of various data augmentation methods for railway turnout classification is discussed below.

1) Promising Data Augmentation

a) Color Jitter: Color Jitter can randomly change the brightness, contrast, saturation, and hue of an image. It can improve the robustness of the model as railway scenes may face different lighting and weather conditions.

b) Scaling: The shooting distance will cause the scale of the turnout to change, so scaling helps the model handle different scales.

c) Normalize: It maps data to a smaller range, reducing the influence of outliers on the model, thus enhancing its robustness and convergence speed.

2) Inappropriate Data Augmentation

a) Random Crop: The turnout usually appears in the center area of the image. Random cropping may crop out critical discriminative regions. In Fig. 8, we use a heat map to show the location distribution of turnouts on the image.

b) Horizontal & Vertical Flip: The left switch becomes a right switch after horizontal and vertical flipping, see Fig. 9, both destroying the original feature distribution. Therefore, neither horizontal nor vertical is appropriate.

c) Rotation: As the train travels along the railway track, the forward view is parallel to the track (excluding curved sections), making it unnecessary to introduce rotation.

d) Adding Noise: Excluding some specific scenarios such as rain, snow, and overgrown vegetation, the background of railway areas is generally clear. Introducing noise would, in fact, introduce unwanted interference.

e) Image Mixing: The dataset for railway switches is relatively small, and classification is a straightforward task. Mixing data may result in overly complex samples, which is detrimental to model performance.

f) Warp: The determination of the turnout direction is based on the distance between the switch rail and the stock rail. Image distortion may destroy this distance information.

By the way, it is noteworthy that data pre-processing for training and validation differs. During the training phase, different approaches of data augmentation are needed to generalize the data distribution. In contrast, during the inference phase, such augmentation is unnecessary. It is sufficient to ensure consistency only in the scaling and normalization strategy employed during training. We used ResNet18-SE in ablation experiments to validate the findings outlined above. Similarly, the models are configured with an input size of 640×360. The ablated variables in data augmentation included Resize, CenterCrop, RandomResizedCrop, Horizontal & vertical Flip (H-Flip, V-Flip), Color Jitter, and Blur. Observing the experimental results in Table IV, the following conclusions can be drawn. Firstly, random scaling performs the poorest as crucial discriminative regions may be randomly removed. Secondly, image normalization can enhance the classification accuracy of the model to some extent. Thirdly, as aforementioned, color jitter can significantly improve the accuracy, while horizontal and vertical flips as well as noise addition (e.g. Blur) may degrade model performance. Following the processing of resizing, normalization, and color jittering, we achieved the optimal model with an accuracy of 89.03%. To further validate the classification decisions, we employ the explainability-oriented Grad-CAM [24] algorithm to compare class activation maps (CAM) generated by the pre-optimized model (Resize-only) and the optimal model (through Resize, normalization, and Color Jitter) for input images, as illustrated in Fig. 11. Results indicate that the refined model focuses more on discriminative regions, contributing to improve the classification accuracy.

The images must be converted to the same size before being fed into the model for training since we employ single-scale training. Resizing images is an important step to take. The above analysis focused on data augmentation methods in

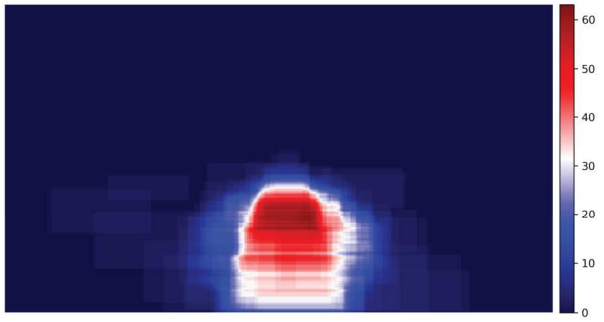

Fig. 8. Heat map of turnout locations on the image.

Fig. 9. Data augmentation of turnout discriminative region. The original left switch turns to right switch after horizontal and vertical flipping.

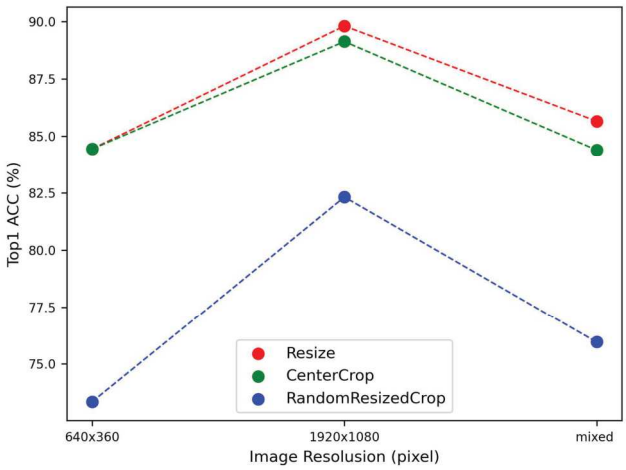

Fig. 10. Visualization of the impact of image resolutions on model performance. Image resolutions from left to right are 640×360, 1920×1080, and a mixture of the two.

TABLE IV. ABLATION RESULTS OF DATA AUGMENTATION ON RESNET18-SE

Resize	CenterCrop	RandomResizedCrop	Normalize	H-Flip	V-Flip	Color Jitter	Blur	Top-1 Acc
✓								85.65
	✓							84.39
		✓						75.95
		✓	✓					83.12
	✓		✓					86.50
✓			✓					86.92
✓			✓	✓				84.39
✓			✓		✓			85.23
✓			✓			✓		**89.03**
✓			✓			✓	✓	82.28
	✓		✓	✓				83.97
	✓		✓		✓			85.23
	✓		✓			✓		86.92
	✓		✓			✓	✓	85.65

TABLE V. COMPARISON OF MODEL PERFORMANCE FOR DIFFERENT IMAGE RESOLUTION AND SCALING STRATEGIES

Input Size	Image Resolution	Scaling Method	Top-1 Acc（%）				
			Left	*Straight*	*Right*	*Unknown*	*All*
640×360	640×360	Resize	80.77	95.24	76.19	86.36	84.44
		CenterCrop	92.31	95.24	80.95	68.18	84.44
		RandomResizedCrop	69.23	95.24	71.43	59.09	73.33
	1920×1080	Resize	68.18	100	80	87.5	89.80
		CenterCrop	77.27	100	80	75	89.12
		RandomResizedCrop	68.18	100	64	58.33	82.31
	640×360 1920×1080	Resize	83.33	97.94	82.61	65.22	85.65
		CenterCrop	70.83	98.97	80.43	71.74	84.39
		RandomResizedCrop	66.67	95.88	54.35	65.22	75.95

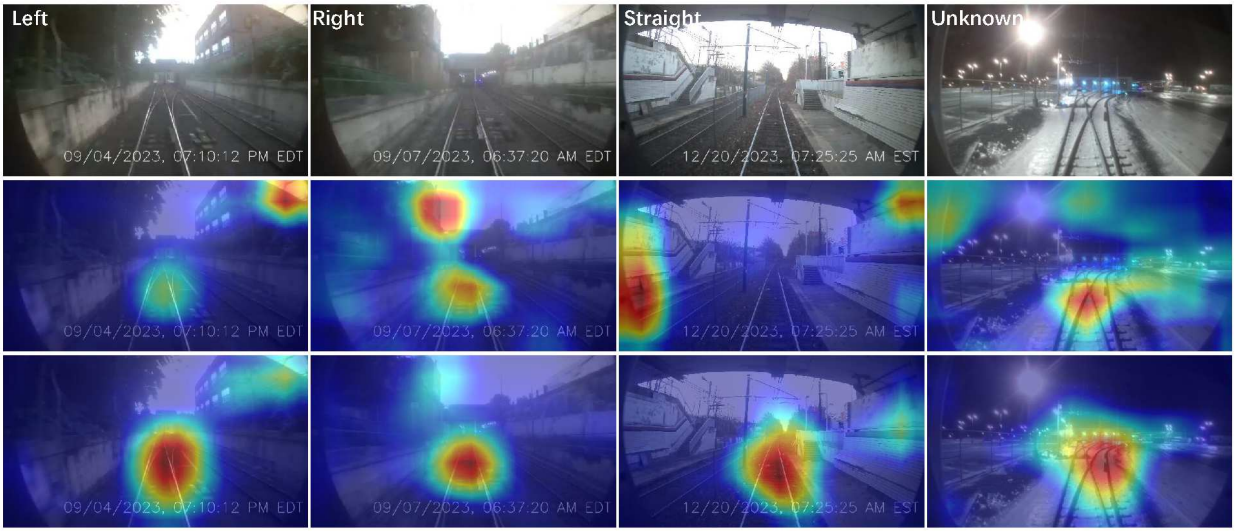

Fig. 11. Comparison of CAM across four classes. The maps highlight the discriminative regions in images used for image classification. Darker shades of red indicate greater contributions to the predicted output. From top to bottom rows respectively: original image, heatmap generated by the pre-optimized model, and optimal model.

turnout classification without delving into image resizing methods. In order to ensure the robustness of the algorithm, we use images of different resolutions, which may have an impact on the image resizing method. Taking into account the model input size and image resolution, we have to carefully evaluate the resizing approaches. A thorough analysis of it is presented in the following section.

D. Effect of Image Resolution

Our data in this study come in two resolutions: 640×360 and 1920×1080. Common ways for image resizing methods include Resize, CenterCrop, and RandomResizedCrop. To investigate the impact of image resolution, model input size, and image scaling methods on switch classification, we conducted experiments using ResNet18-SE. We evaluate the impact on turnout classification performance when the model input size is 640×360, without using any enhancement, at 640×360, 1920×1080, and a mixture of two. The experimental results are shown in Table V and Fig. 10. Based on the observations of these data, we can draw the following conclusions.

- The likelihood of turnouts occurring in the image's center is comparatively high, random cropping may result in the loss of critical information. As a result, the center-cropping approach is better than the random cropping approach.
- Under fixed input scales, high-resolution image classification consistently outperforms low-resolution and mixed-resolution images. This phenomenon arises from the richer texture and contextual information present in high-resolution images, making it easier to obtain discriminative features.
- In contrast to high-resolution images, low-resolution images exhibit relatively lower sensitivity to resizing and cropping operations. Low-resolution images inherently possess lower information density, thereby minimizing the introduction of distortion through resizing. The crop operation may not have much impact on the information content of the image as well, because the low-resolution image itself is relatively small, and most of the key information of the image may still be retained after cropping. However, if the crop size selected is too small, some important information in the image may be cropped out, thus affecting the performance of the model.
- For high-resolution images, the Resize strategy is slightly better than the CenterCrop, but this is not always certain because the input size of the model will also affect the scaling strategy. But no matter what scaling method is used, increasing the training size will generally enhance model performance within a certain range, but the improvement will gradually decrease and stop increasing or even decline after a certain scale threshold. This phenomenon stems from the limited model ability of feature learning. After exceeding a certain size, the model may not be able to effectively utilize additional information, and may even regard it as noise, affecting generalization ability.

E. Model Deployment

Deployment is the process to provide service for users utilizing the trained AI models. There are two options for model deployment depending on where the server that hosts the AI model is located. Cloud computing refers to transferring video frames and others over the internet to a remote provider, conducting the analytics and storing the results there. Edge computing refers to the decentralized analytics at or near the source of data generation, such as IoT devices, sensors, or edge servers.

In the realm of deep-learning-based models, GPU holds significant importance for both model training and inference processes. Nonetheless, within the railroad industry, the utilization of GPU-powered edge computers remains limited, despite their potential benefits. To enhance scalability, it is important to explore alternative deployment options that rely solely on CPUs. Leveraging frameworks like OpenVINO, AI inferencing can be optimized for CPU-based systems. These frameworks facilitate the acceleration of AI inferencing by optimizing neural network models and deploying them on a wide range of hardware architectures, including CPUs. By harnessing the capabilities of OpenVINO, the deployment of AI solutions for switch recognition on edge computers equipped with CPUs alone becomes a viable and scalable approach, offering potential benefits for the railroad industry.

We developed a hybrid edge and cloud computing system to optimize efficiency, which was tested on a testbed with an "edge computer" that was a cluster of 8 micro-computers with solely CPUs. On the one hand, turnout classification inferencing is placed in the edge computer since the low latency is needed for real-time analysis and the on-board network connectivity is unreliable or limited. In leveraging the edge computer cluster, we first enhance the turnout recognition model using OpenVINO for accelerated processing. and then model was deployed across all eight micro-computers with a load balancer for efficient distribution of tasks. On the other hand, it transfers certain metadata to the cloud for offline analysis and presentation on a dashboard.

IV. Conclusions

In this paper, we investigated vision-based turnout recognition algorithms, using a custom dataset to analyze some challenges and solutions associated with vision-based turnout recognition task. Extensive experiments were conducted to analyze the impact of data augmentation, image resolution, training size on the model performance. Our work provides some strong support in terms of empirical prior knowledge for intelligent turnout recognition in rail transit. However, we believe that customized vision-based turnout recognition tasks have not been extensively studied. For example, long-distance, multi-modal turnout recognition still has great potential to improve the accuracy and robustness of the framework. These are the areas we intend to focus on in future research efforts.

References

[1] D. Lu and E. Schnieder, "Performance evaluation of gnss for train localization," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 2, pp. 1054–1059, 2014.

[2] J. Liu, B.-g. Cai, and J. Wang, "A gnss/trackmap cooperative train positioning method for satellite-based train control," in *Proc. IEEE Int. Conf. Intell. Transp. Syst.*, 2014, pp. 2718–2724.

[3] R. Krc, J. Podrouzek, M. Kratochvılova, I. Vukusic, and O. Plasek, "Neural network-based train identification in railway switches and

crossings using accelerometer data," *J. Adv. Transp.*, vol. 2020, pp. 1–10, 2020.

[4] H. Qin, A. Zaman, and X. Liu, "Artificial intelligence-aided intelligent obstacle and trespasser detection based on locomotive-mounted forward-facing camera data," *Proc. Inst. Mech. Eng. Part F-J. Rail Rapid Transit.*, vol.237, no. 9, pp. 1230–1241, 2023.

[5] A. Zaman, Z. Huang, W. Li, H. Qin, D. Kang, and X. Liu, "Artificial intelligence-aided grade crossing safety violation detection methodology and a case study in new jersey," *Transp. Res. Rec.*, vol. 2677, no. 10, pp. 688–706, 2023.

[6] X. Wang, Y. Bai, and X. Liu, "Prediction of railroad track geometry change using a hybrid cnn-lstm spatial-temporal model," *Adv. Eng. Inform.*, vol. 58, p. 102235, 2023.

[7] J. Wohlfeil, "Vision based rail track and switch recognition for self-localization of trains in a rail network," in *Proc. IEEE Intell. Veh. Symp.* IEEE, 2011, pp. 1025–1030.

[8] M. Karakose, O. Yaman, and E. Akin, "Detection of rail switch passages through image processing on railway line and use of condition-monitoring approach," in *Int. Conf. Adv. Technol. & Sciences, ICAT*, vol. 16, 2016, pp. 100–105.

[9] O. Zendel, M. Murschitz, M. Zeilinger, D. Steininger, S. Abbasi, and C. Beleznai, "Railsem19: A dataset for semantic rail scene understanding," in *Proc. IEEE Conf. Comput. Vis. Patt. Recogn. Workshops*, 2019, pp. 0–0.

[10] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Patt. Recogn.*, 2017, pp. 4700–4708.

[11] K. Jahan, J. Niemeijer, N. Kornfeld, and M. Roth, "Deep neural networks for railway switch detection and classification using onboard camera images," in *Proc. IEEE Symp. Ser. Comput. Intell. (SSCI)*, 2021, pp. 01–07.

[12] A.-R. Alexandrescu, A. Manole, and L. Diosan, "Railway switch classification using deep neural networks." in *VISIGRAPP (4: VISAPP)*, 2023, pp. 769–776.

[13] L. Van der Maaten and G. Hinton, "Visualizing data using t-sne." *J. Mach. Learn. Res.*, vol. 9, no. 11, 2008.

[14] K. Oksuz, B. C. Cam, S. Kalkan, and E. Akbas, "Imbalance problems in object detection: A review," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 10, pp. 3388–3415, 2020.

[15] C. Shorten and T. M. Khoshgoftaar, "A survey on image data augmentation for deep learning," *J. Big Data*, vol. 6, no. 1, pp. 1–48, 2019.

[16] F. Zhuang, Z. Qi, K. Duan, D. Xi, Y. Zhu, H. Zhu, H. Xiong, and Q. He, "A comprehensive survey on transfer learning," in *Proc. IEEE*, vol. 109, no. 1, pp. 43–76, 2020.

[17] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[18] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Patt. Recogn.*, 2016, pp. 770–778.

[19] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proc. IEEE Conf. Comput. Vis. Patt. Recogn.*, 2018, pp. 7132–7141.

[20] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," in *Proc. IEEE Conf. Comput. Vis. Patt. Recogn.*, 2018, pp. 4510–4520.

[21] M. Tan and Q. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," in *Int. Conf. Mach. Learn.* PMLR, 2019, pp. 6105–6114.

[22] X. Liu, H. Peng, N. Zheng, Y. Yang, H. Hu, and Y. Yuan, "Efficientvit: Memory efficient vision transformer with cascaded group attention," in *Proc. IEEE Conf. Comput. Vis. Patt. Recogn.*, 2023, pp. 14 420– 14 430.

[23] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2020.

[24] X. Ding, X. Zhang, N. Ma, J. Han, G. Ding, and J. Sun, "Repvgg: Making vgg-style convnets great again," in *Proc. IEEE Conf. Comput. Vis. Patt. Recogn.*, 2021, pp. 13733–13742.

[25] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, "Learning deep features for discriminative localization," in *Proc. IEEE Conf. Comput. Vis. Patt. Recogn*., 2016, pp. 2921–2929.