Automatic Detection of Rail Components via A Deep Convolutional Transformer Network

Tiange Wang, Zijun Zhang, Fangfang Yang, and Kwok-Leung Tsui

Abstract—Automatic detection of rail track and its fasteners via using continuously collected railway images is important to maintenance as it can significantly improve maintenance efficiency and better ensure system safety. Dominant computer vision-based detection models typically rely on convolutional neural networks that utilize local image features and cumbersome prior settings to generate candidate boxes. In this paper, we propose a deep convolutional transformer network based method to detect multi-class rail components including the rail, clip, and bolt. We effectively synergize advantages of the convolutional structure on extracting latent features from raw images as well as advantages of transformers on selectively determining valuable latent features to achieve an efficient and accurate performance on rail component detections. Our proposed method simplifies the detection pipeline by eliminating the need of prior settings, such as anchor box, aspect ratio, default coordinates, and post-processing, such as the threshold for non-maximum suppression; as well as allows users to trade off the quality and complexity of the detector with limited training data. Results of a comprehensive computational study show that our proposed method outperforms a set of existing state-of-art approaches with large margins.

Index Terms—attention mechanism, condition monitoring, deep learning, neural networks, rail track

I. INTRODUCTION

The inspection of railway track components is a crucial task in the railway maintenance. Any missing or lapsed railway components may lead to train derailments and other accidents. The recent advancement of computer vision methods and technologies has driven studies of applying convolutional neural networks (CNN) to efficiently inspect rail track components via a computer-aided automation of image processing. From the application perspective, previous studies can be categorized based on their detection targets and most of them focus on the defect detection [1-4] as well as the fastener detection [5-8]. These detection tasks generally consider targets of the uniform size and similar shape. In addition, components of different categories are detected independently, which means only local features of the railway track image are utilized in these tasks. From the methodological side, the direct application of classical CNN models into rail track component detections faces following challenges, cumbersome prior settings and post-processing. The prior information including region proposals [9], anchor box [4], and center points [10] needs to be appropriately defined as it might affect the overall performance. A large set of candidate boxes based on priors for all the pixels in feature map are generated even most of them are meaningless, which leads to a computational burden in the training process. After the generation of candidate boxes, post-processing, such as non-maximum suppression (NMS) [11], is usually applied to remove redundant boxes, which leads to more hyper-parameters and complicates the model tuning process.

To extract global context without preprocessing of input images and to realize multi-class detection with regards of much variety on object size in a more concise way, we propose a detection framework consists of a deep CNN structure and a decoder-only transformer to simplify the computational process in predictions and enhance the detection accuracy of rail track components. Although the transformers [12] have achieved a considerable success in the field of natural language processing, such as question answering and machine translation, its advantages on processing railway images is untapped. In additional to its scarcity in railway applications, applications of the transformer into detection tasks always follow a standard architecture of the same numbers of encoder and decoder layers [13, 14]. We challenge this convention by using only a shallow decoder instead of a full transformer to finish the class and bounding box (bbox) prediction. In this work, the significance of adopting the decoder on rail component detection task is well explored. The self-attention mechanism in decoder helps to build correlations between elements in feature maps that can be treated as the sequence of transformer input. Characteristics of transformer determine that the prediction size of decoder can be manually decided, which saves the cost on introducing prior conditions and simplifies the whole computation.

The detection targets are displayed in Fig. 1. Usually, a rail is located vertically and cross the whole image. Clips and bolts are symmetrically distributed on both sides of the rail. However, due to the changes in shooting conditions, the resulting images are usually asymmetric, and there will be several instances located at the edge of the image. In this case, we are not only committed to improving the overall detection efficiency, but also focus on strengthening the detection of edge objects. As visualized in Fig. 1, the detector we propose reaches higher confidence scores for all components compared to the traditional detector only using CNN, which indicates that the proposed detector can still achieve good detection performance even if the threshold is raised.

Main contributions of this work are summarized as follows:

1. A deep convolutional transformer based detector is proposed to enhance the rail component detection while maintain a low computational complexity of 65 billion
floating point operations (BFLOPS). By visualizing attention weights from the transformer, we demonstrate that the decoder block can put attentions on valuable latent features.

2. We analyze the benefit of the data augmentation on detecting truncated and small objects. By enriching the diversity of training set with additional augmentation methods, the detection of clips, which is a more challenging task, can be significantly enhanced.

3. Our method leads to significant improvements of 10.1 average precision (AP) and 9.2 AP against Faster RCNN on detection of small clips and bolts, respectively. By incorporating the attention mechanism into the decoder, the detector yields the highest accuracy of 61.9 AP with inference time of 32.2ms, which outperforms the existing state-of-the-art approaches.

II. RELATED WORK

Image processing technologies have been widely adopted in the rail visual inspection system, they directly affect the detection accuracy and speed. The deep convolutional neural network (DCNN) based approaches are the mainstream in previous studies with regard of specific targets, such as surface defects, fasteners, and tracks [15]. By establishing a dataset of the track image and manually calibrating the target type, the classification and localization are integrated to predict at the same time.

Rail surface defects mainly include surface cracks and rolling contact fatigue wears. They are usually small and irregularly distributed. A common way of detecting the surface defects is cropping the interested region out before training. The early rail surface-defect detection methods usually consider the hand-crafted characteristics of the image. Surface defects are identified by filtering, texture analysis, and classification of the input images [16]. With the rapid development of CNN methods, numerous approaches and systems have been designed to improve detection efficiency. Shang et al. [3] applied CNN to build a two-stage pipeline for localizing and classifying rail defects based on cropped rail images. Yanan et al. [4] took advantages of the YOLOv3 algorithm to realize the rail surface defect detection. Results were shown as accurate and fast. By preprocessing the railway track images, the size of the defects was fixed, and the number of various defects was not as dense as before, which helped to improve the detection performance. Faghih-Roohi and Hajizadeh [17] compared the detection effects of DCNNs with three different sizes and number of parameters. It concluded that although with longer training time, a deeper DCNN model should performed better than the shallower DCNN models in the defect detection.

Track fasteners are important components that prevent rail overturning and longitudinal movement. Therefore, the visual inspection of track fasteners is primarily for detecting the absence of or damage to fasteners. Li et al. [18] introduced a railway track detection system based on real-time computer vision. The hardware system of the research consisted of multiple cameras, global positioning systems and distance measuring instruments. Although there were three targets (anchor, tie, and tie plate) in the detection task, tie plate detection served the first step in the detection pipeline since it provided information to define the region of interests (ROI), in which other components could be located. Similarly, the support vector machine was combined with ROI in [19] to classify tie plate. Khan et al. [5] presented an automatic detection technique for detecting the absence of rail fasteners. The Harris-Stephen and Shi-Tomasi feature detectors were used to extract the feature points and feature vectors of the image. Similar with defect detection, previous detection work of rail fasteners was mainly about single class detection, which required uniform size and similar shape of targets. Even like multi-class detection task in [18], it failed to predict different types of objects at the same time. Moreover, additional inputs, such as the distance and ROI, were required to implement the component detection.

As for the detection of rail track, it is predominantly used for autonomous driving and foreign object detection. Ye et al. [20] proposed a fusion refine neural network (FR-Net) based on CNN to detect objects ahead in railway shunting mode. FR-Net applied a two-step strategy: a coarse detection module first refined locations and prior anchors as well as a finer detection module for more accurate object locations and classifications. Kaleli and Akgul [21] put forward a dynamic programming method to extract the train course and frontal railroad track. By using the video data, rails on the left and right sides were identified according to the vanishing point. Experiments showed that this method had good robustness at higher vehicle speeds. Weichselbaum et al. [22] proposed a 3D vision-based obstacle-detection system. The system used a laser scanner to operate single and stereo cameras in the visible and infrared spectra, as well as radar and ultrasonic sensors.

So far, the implementation of various railway detection tasks still relies on the feature extraction with CNNs. Some approaches even require additional inputs, such as the distance, laser, or sonic data. Priors and post-processing are also required to generate a sufficient number of candidate boxes and to remove redundant boxes, which makes the model tuning complex. To address the limitation of applying the pure CNN model into railway image data, the attention mechanism is introduced into our proposed multi-class detector, which, to the
III. METHOD DESCRIPTION

The overall network architecture of the proposed detector is depicted in Fig. 2. It is mainly composed of three blocks, a CNN backbone for extracting a compact feature representation, a transformer decoder for selectively discriminating the CNN output, as well as two parallel feed forward networks (FFN) for predicting the final class and bobox, respectively. Compared with many general detectors, the proposed model is developed based on DETR [13] to achieve a higher accuracy and lower model complexity as well as to work with limited railway image data.

The whole framework is illustrated as follows. Let a function $G(\cdot)$ denote the CNN architecture to obtain feature maps with a sufficient depth. The input RGB image $x \in \mathbb{R}^{3 \times H_0 \times W_0}$ generates lower-resolution feature maps $f \in \mathbb{R}^{C \times H \times W}$ via $f = G(x)$. In our experiments, $G(\cdot)$ mainly presents the ResNet-50, which is one of the classic frameworks in image recognition tasks, so that $H = \frac{H_0}{32}, W = \frac{W_0}{32}, c = 2048$ according to [23]. Before putting into the transformer decoder $D(\cdot)$, a 1x1 convolution is applied to reduce the channel dimension of the high-level activation map $f$ from $c$ to a smaller dimension $d = 512$. Spatial dimensions are then collapsed into one dimension, resulting in a two-dimension feature map $f_0 \in \mathbb{R}^{HW \times d}$. The sequence of attention input $f_0$ is defined as follows:

$$f_{\text{out}}(i, j, d_k) = f_{\text{in}}(i, j, c) \ast F_k(1, 1, c),$$

$$f_0[H \ast W, d] \equiv f_{\text{out}}[H, W, d],$$

where $f_0$ denotes the $k$-th filter weight with a depth $c$, and $d_k$ denotes the $k$-th dimension in the output feature map.

In our approach, the $f_0$ with fixed positional embedding $\text{pos}_{\text{relative}}$ directly passes through the decoder part, in which each element in the sequence learns an alignment to gather from others. After the positional encoding, three stacked decoders also take learned positional embedding $\text{pos}_{abs}$ as one of inputs to generate a fixed size of predictions. To avoid autoregressive, the number of input tokens of decoder has been determined by the length of $\text{pos}_{abs}$ for realizing parallel decoding. Two types of positional encoding are defined as follows:

$$\text{pos}_{\text{relative}}^{(l)} = \begin{cases} \sin\left(\frac{i}{10^{i/128}}\right), & j = 2t \\ \cos\left(\frac{i}{10^{i/128}}\right), & j = 2t + 1 \end{cases}$$

$$\text{pos}_{abs} \in \mathbb{R}^{50 \times 512} \sim N(0, 0.1^2),$$

where $i \in \{0, 1, \ldots, HW - 1\}, j \in \{0, 1, \ldots, d - 1\}$. Because the transformer is permutation-invariant, information about the positions must be added into the input embedding. For $\text{pos}_{\text{relative}}$, sinusoidal positional encoding is applied to inject relative positional information into latent features. The $\text{pos}_{abs}$ is randomly initialized from a normal distribution $N(0, 0.1^2)$. $D(\cdot)$ is pushed to learn the position features on objects with the positional embedding continuously added to each layer.

As shown in Fig. 2, each decoder layer follows the standard architecture of the transformer. It contains a multi-head self-attention layer $A_{\text{self}}(\cdot)$ for passing queries, a multi-head cross-attention layer $A_{\text{cross}}(\cdot)$ for building the information transfer with backbone, as well as a feed-forward layer for projecting the refined matrix to a larger space and extracting the required information more easily. The matrix calculation of $A_{\text{self}}(\cdot)$ generates three matrices from the same input, which are queries $(Q)$, keys $(K)$, and values $(V)$. A single-head self-attention layer is computed as follows:

$$Z^t_{\text{single}} = A^t_{\text{self}}(X_{\text{input}}) = \text{softmax}\left(\frac{Q^t V^t}{\sqrt{d}}\right),$$

where $Q^t = X_{\text{input}} W^t_Q$, $K^t = X_{\text{input}} W^t_K$, and $V^t = X_{\text{input}} W^t_V$.

Notice that $W^t_Q \in \mathbb{R}^{d \times d}$, $W^t_K \in \mathbb{R}^{d \times d}$, and $W^t_V \in \mathbb{R}^{d \times d}$. In the self-attention layer, $Q$, $K$, and $W$ perform the attention function in parallel and yield $d_v$-dimensional output values. Multiple $Z^t_{\text{single}}$ are concatenated and output a final $Z_{\text{multi}}$ with another weight $W_0 \in \mathbb{R}^{hd_v \times d_v}$. We follow [12] and set $h = 8, d_k = d_v = \frac{d}{h} = 64$.

$A_{\text{cross}}(\cdot)$ works in a similar way. The only difference is that $Q$ is passed from the last self-attention layer while $K$ and $V$ are passed from $f_0$, which plays a roll in cross-transfer of information. In our method, the first self-attention layer in the first decoder block takes only $\text{pos}_{abs}$ as input for generating $Q$ to pass through next cross-attention layer. In each $A_{\text{cross}}(\cdot)$ layer, $K$ and $V$ are passed using different weights. With attention mechanism, $D(\cdot)$ globally reasons about all components together using pair-wise relations between them, while being able to use the whole image as context.

The loss function of our method follows the common object detectors, which includes a linear combination of $-\log[p(c)]$ for class predictions as well as $L_{\text{box}}(b, \tilde{b})$ for the similarity of predicted and ground truth boxes. Specifically, $L_{\text{box}}(b, \tilde{b})$ is composed of the $L_1$ loss of the center coordinates and $giou(b, \tilde{b})$ of the normalized sizes. Since the decoder in the model deals with inputs in parallel, a pair-wise matching cost is defined to find a bipartite matching between a prediction within index $\sigma(i)$ and ground truth $y_i$ with the lowest cost:

$$L = \sum_{i}^{\text{batch size}} L_{\text{box}}(b_i, \tilde{b}_i) = \sum_{i}^{\text{batch size}} giou(b_i, \tilde{b}_i) + \lambda \sum_{i}^{\text{batch size}} \frac{1}{2} ||b_i - \tilde{b}_i||_1,$$

where $\lambda$ is a hyperparameter.

Fig. 2. The deep convolutional transformer network consists of a deep CNN backbone, a 3-layer decoder, and two parallel FFN to predict class and bobox, respectively.
\[ L_{\text{match}}(y, \hat{y}) = \sum_{i=1}^{50} \left[ -\log \hat{p}_{\theta(i)}(c_i) + 1_{\{c_i \neq \emptyset\}}L_{\text{box}}(h_i, \hat{b}_{\theta(i)}) \right], \]
\[
\hat{b} = \arg\min L_{\text{match}}(y, \hat{y}),
\]
\[
L(y, \hat{y}) = \sum_{i=1}^{50} \left[ -\log \hat{p}_{\theta(i)}(c_i) + 1_{\{c_i \neq \emptyset\}}L_{\text{box}}(h_i, \hat{b}_{\theta(i)}) \right],
\]

where \( \hat{p}_{\theta(i)}(c_i) \) is the class probability of class \( c_i \) and \( \hat{b}_{\theta(i)} \) is the predicted bbox. This procedure of finding matching plays the same role as the heuristic assignment rules used to match proposal or anchors to ground truth objects in modern detectors. By finding one-to-one matching for the direct set prediction without duplicates, the post-processing step is no longer needed at the inference time. With \( \hat{b} \), the loss function is calculated in (8).

The overall training is displayed in Algorithm 1. Images as well as labels including class \( C \) and bbox \( B \) are utilized to train the model. The final prediction is computed by a 3-layer perceptron with sigmoid function \( \text{mlp}(\cdot) \), \( \text{sigmoid} \) and a linear projection \( \text{linea}(\cdot) \) to predict bbox and class, respectively. \( L(y, \hat{y}) \) is calculated for every batch of inputs. If the loss value gets infinite, the training procedure stops automatically. Otherwise, the gradient \( \nabla \theta_t \) is calculated via \( \text{back propagation} \) and model weights \( \theta_t \) are updated with a learning rate \( \eta \).

Usually, in a transformer, there is an encoder before the decoder part. However, in our detection architecture, parallel decoding is fully capable to predict objects. If an encoder was added after the backbone, the encoder and decoder are learning from the very same input, which is considered as redundant learning in our proposal because of the same self-attention mechanism. The decoder-only design also helps to remove model weights and further reduce the computational complexity.

### IV. COMPUTATIONAL EXPERIMENTS

In this section, we first introduce the dataset of railway track images used in our study. Then, the detailed description on the training setup with model structures are provided. Results and analysis are described at last.

#### A. Dataset

The experiments are conducted with rail surface image dataset collected by the Hong Kong Metro Corporation (MTR). The image acquisition system consists of two cameras and six LED lights that installed under the train. Cameras randomly take images that are uploaded to the server manually when the train moves with a low speed. Due to changes of the environment, the brightness and size of images vary a lot. Fig. 1(a) displays target components in railway track images from the training set. Targets of the multi-class detection task include the rail track, clip, and bolt. Rail track is the largest component, its height is basically the height of the image. Clips and bolts are used to fasten the rail tracks and they are symmetrically distributed on two sides of the track. Because of vibrations on devices, clips can be partially captured with a relatively small ground truth box and an edge location. Bolts are distributed on the same side with clips while they can be completely captured by the image acquisition system because of the smaller size. This collected dataset contains 691 images for training, 345 images for validation and 350 test images. 6451 instances from 3 categories are annotated with ground truth bounding boxes including the class number, center coordinates, width, and height. Basically, each image includes at least one rail track with a varying number of bolts and clips.

Compared with some multiple object detection problems, targets in the considered detection do not overlap, which makes our detection task more specific. Nevertheless, there are still two challenges in the rail component detection. First, our dataset is limited compared to public domain dataset, such as COCO and PASCAL VOC. Thus, available samples for training are limited. Secondly, a considerable part of clips in the dataset is truncated. It is challenging to detect truncated components due to their significantly smaller sizes and more edge positions. Except for basic photometric distortion and geometric distortion [24], such as flipping, rescaling, exposure change and saturation change, extra data augmentation methods are utilized to overcome the limitation of the railway dataset itself and increase the diversity of the training set. One of them is stitcher [25]. Four different images are stitched in spatial dimension as one image for training. Thus, four different contexts are mixed and learned at the same time. Fig. 3(e) is a sample of stitcher method. The width and height of each image are rescaled to 1/2 of the averaged width and height of four images. By stitching them together, the composite image is about the same size as the normal image. Clips that were originally at the edge are located in the middle of the image with significantly smaller normalized size, which is potential to strengthen the detection of edge objects and small objects at the same time. Copy-pasting [26] is another augmentation approach focusing on detection of small clips. The number of truncated clips and bolts is increased by pasting them to different positions of the same image so that the contribution of
In addition to using AP to represent the accuracy of locating and classifying targeted objects, APs under iou = 0.50 (AP@50) and iou = 0.75 (AP@75) are also compared to clarify the detection capacity. The precision and recall required for \( p(f) \) are defined as follows:

\[
\text{precision} = \frac{\text{true positive}}{\text{true positive} + \text{false positive}} \\
\text{recall} = \frac{\text{true positive}}{\text{true positive} + \text{false negative}}
\]

Since the number of predictions is limited, the precision and recall values are discrete. \( AP^{iou} \) is easily to get by measuring the exact area under precision-recall curve. The metrics above are all averaged across different object categories, in which we primarily focus on small objects including clips and bolts. If not specified, AP from the last training epoch is reported as the validation AP. Except for accuracy, inference speed is another essential criterion to evaluate the performance of a detector. Since the input of our proposed detector are images, we use the average inference time on test set to compare the processing speed of the detector on each image. BFLOPs that are accumulated over multiple calculations and number of parameters are also recorded to estimate the algorithm complexity.

C. Results and Analysis

The convolutional transformer model is performed with pretrained Resnet-50 as the default backbone. Attention features are processed with 3 standard decoder layers. Although our available railway image dataset is limited, the training volume has been doubled with the help of data augmentation technologies to avoid overfitting. Fig. 4 shows the detection results of the trained model on the validation set. It compares the impact of data augmentation methods on the overall component detection and three categories, respectively. By applying new augmentation methods, AP value over multiple thresholds has increased by nearly 8%. When compare the precision-recall curve over all predictions, data augmentation has a significant impact on the detection of clips with no damages on the other two categories.

Since the transformer was first proposed, many prior works have assumed that the transformer architecture in detection tasks has the same number of encoder and decoder layers [13, 14]. We challenge this convention and explore the importance of encoder by changing the depth of transformer to derive the optimal depth for railway component detection task. Fig. 5 indicates that, in our approach, three decoder blocks are enough to yield highest accuracy with the smallest parameters and complexity compared to other layer allocations. Although the allocation of 1 encoder and 2 decoders has comparable performance on validation set, it is still inferior compared with decoder-only (the AP value is nearly 1% less). This result
confirms our conjecture that the encoder and decoder are learning the exact same thing in our method because of the same self-attention mechanism. Additionally, with deeper encoder, the detection accuracy drops sharply with comparable depth of decoder, which reveals that the advantages of encoder in railway component detection task are overestimated when the extracted features are fine enough.

Deeper decoder, such as 6 and 9 layers, have been tested in our proposal. Both have the same AP as 3 decoder layers in the absence of encoder. In Fig. 6, the self-attention weights from the last decoder layer are visualized. From top to bottom, the attention with 1, 2, 3, 6, 9 decoder layers are compared. The last row provides the test image and labels. For each target, the decoders put attention on the relative part for predicting specific bbox and class. As the attention level deepens, the model focuses more and more on specific objects instead of focusing on the entire image. The visualization gives the intuition that using 3 decoders could attain the same attention effect as using more stacked decoders in the proposed detection architecture. By directly concatenating the backbone and decoder, a new speed-quality baseline is provided for the future railway research toward a fast and accurate detection. Remaining experiments also prove that this simplified configuration attains the comparable detection accuracy just as using the full transformer model.

Besides the impact of layer allocations, the influences of prediction size on validation set are evaluated as well in Table 1. Since the learned positional encoding is one of inputs of the decoder, the prediction size should be the same as the sequence length of $pos_{abs}$. This is decided by the characteristics of the transformer. Although the maximum number of objects in an image of our dataset is 9, setting a large number of predictions with more slack (such as 50) can effectively increase the validation AP by almost 4% and improves the AP@50 by 5.2% compared with 10 predictions. For detectors with larger output sizes, they basically maintain the same accuracy but introduce more computations. Therefore, we set 50 as the output size in the training process. Finally, a full comparison between our proposal and other representative detectors on test
set is given in Table 2. Backbones utilized in our experiments including Vgg, ResNet, and Darknet are all pretrained on ImageNet [24], which has 14 million annotated images and contains 1000 categories to train a large-scale model. After concatenating with a shallow decoder, the proposed models are fine-tuned to solve the problem of insufficient training data for our railway component detection task. To achieve comparable AP values, Faster RCNN models and YOLO series are trained for a longer schedule of 10k iterations with limited training data. By applying multiple thresholds, our proposal achieved 61.9 AP performance on test set, which was 9.0 AP higher than the traditional detector using only CNNs. In detecting small or truncated clips and bolts, the self-attention mechanism helped to capture local context to better localize and size them, which resulted in at least 9.2 AP higher accuracy than the baseline. Our proposal made a lot of efforts on reducing the computational complexity and memory capacity, its inference speed was comparable to Faster RCNN models but still significantly slower than YOLOv2. The experiments revealed the limitation of using the encoder in transformer structure and proved that a shallow decoder was enough to realize good detection effect, which also indicated a reduction of the model complexity. With the detection framework simplified, we found a balance between the global and local context as well as achieved better performances on both of the accuracy and speed.

Besides the successful detection for truncated clips, the potential of convolutional transformer network on benefiting other rail track condition monitoring applications via analyzing railway images needs to be further explored. In future work, we will focus on automatic industrial inspections on small surface defects, which can benefit from the attention mechanism. Another challenge for the proposed method is further decreasing the model complexity and speeding up the detection, such that the real-time inspection is able to be implemented with videos as an input.

V. CONCLUSION

In this work, a deep convolutional transformer based method for detecting railway track components was introduced. It operated effectively to attain more advanced detection performance with limited training data. By applying multiple thresholds, our proposal achieved 61.9 AP performance on test set, which was 9.0 AP higher than the traditional detector using only CNNs. In detecting small or truncated clips and bolts, the self-attention mechanism helped to capture local context to better localize and size them, which resulted in at least 9.2 AP higher accuracy than the baseline. Our proposal made a lot of efforts on reducing the computational complexity and memory capacity, its inference speed was comparable to Faster RCNN models but still significantly slower than YOLOv2. The experiments revealed the limitation of using the encoder in transformer structure and proved that a shallow decoder was enough to realize good detection effect, which also indicated a reduction of the model complexity. With the detection framework simplified, we found a balance between the global and local context as well as achieved better performances on both of the accuracy and speed.

Besides the successful detection for truncated clips, the potential of convolutional transformer network on benefiting other rail track condition monitoring applications via analyzing railway images needs to be further explored. In future work, we will focus on automatic industrial inspections on small surface defects, which can benefit from the attention mechanism. Another challenge for the proposed method is further decreasing the model complexity and speeding up the detection, such that the real-time inspection is able to be implemented with videos as an input.

V. CONCLUSION

In this work, a deep convolutional transformer based method for detecting railway track components was introduced. It operated effectively to attain more advanced detection performance with limited training data. By applying multiple thresholds, our proposal achieved 61.9 AP performance on test set, which was 9.0 AP higher than the traditional detector using only CNNs. In detecting small or truncated clips and bolts, the self-attention mechanism helped to capture local context to better localize and size them, which resulted in at least 9.2 AP higher accuracy than the baseline. Our proposal made a lot of efforts on reducing the computational complexity and memory capacity, its inference speed was comparable to Faster RCNN models but still significantly slower than YOLOv2. The experiments revealed the limitation of using the encoder in transformer structure and proved that a shallow decoder was enough to realize good detection effect, which also indicated a reduction of the model complexity. With the detection framework simplified, we found a balance between the global and local context as well as achieved better performances on both of the accuracy and speed.

Besides the successful detection for truncated clips, the potential of convolutional transformer network on benefiting other rail track condition monitoring applications via analyzing railway images needs to be further explored. In future work, we will focus on automatic industrial inspections on small surface defects, which can benefit from the attention mechanism. Another challenge for the proposed method is further decreasing the model complexity and speeding up the detection, such that the real-time inspection is able to be implemented with videos as an input.

V. CONCLUSION

In this work, a deep convolutional transformer based method for detecting railway track components was introduced. It operated effectively to attain more advanced detection performance with limited training data. By applying multiple thresholds, our proposal achieved 61.9 AP performance on test set, which was 9.0 AP higher than the traditional detector using only CNNs. In detecting small or truncated clips and bolts, the self-attention mechanism helped to capture local context to better localize and size them, which resulted in at least 9.2 AP higher accuracy than the baseline. Our proposal made a lot of efforts on reducing the computational complexity and memory capacity, its inference speed was comparable to Faster RCNN models but still significantly slower than YOLOv2. The experiments revealed the limitation of using the encoder in transformer structure and proved that a shallow decoder was enough to realize good detection effect, which also indicated a reduction of the model complexity. With the detection framework simplified, we found a balance between the global and local context as well as achieved better performances on both of the accuracy and speed.

Besides the successful detection for truncated clips, the potential of convolutional transformer network on benefiting other rail track condition monitoring applications via analyzing railway images needs to be further explored. In future work, we will focus on automatic industrial inspections on small surface defects, which can benefit from the attention mechanism. Another challenge for the proposed method is further decreasing the model complexity and speeding up the detection, such that the real-time inspection is able to be implemented with videos as an input.

V. CONCLUSION

In this work, a deep convolutional transformer based method for detecting railway track components was introduced. It operated effectively to attain more advanced detection performance with limited training data. By applying multiple thresholds, our proposal achieved 61.9 AP performance on test set, which was 9.0 AP higher than the traditional detector using only CNNs. In detecting small or truncated clips and bolts, the self-attention mechanism helped to capture local context to better localize and size them, which resulted in at least 9.2 AP higher accuracy than the baseline. Our proposal made a lot of efforts on reducing the computational complexity and memory capacity, its inference speed was comparable to Faster RCNN models but still significantly slower than YOLOv2. The experiments revealed the limitation of using the encoder in transformer structure and proved that a shallow decoder was enough to realize good detection effect, which also indicated a reduction of the model complexity. With the detection framework simplified, we found a balance between the global and local context as well as achieved better performances on both of the accuracy and speed.

Besides the successful detection for truncated clips, the potential of convolutional transformer network on benefiting other rail track condition monitoring applications via analyzing railway images needs to be further explored. In future work, we will focus on automatic industrial inspections on small surface defects, which can benefit from the attention mechanism. Another challenge for the proposed method is further decreasing the model complexity and speeding up the detection, such that the real-time inspection is able to be implemented with videos as an input.

V. CONCLUSION

In this work, a deep convolutional transformer based method for detecting railway track components was introduced. It operated effectively to attain more advanced detection performance with limited training data. By applying multiple thresholds, our proposal achieved 61.9 AP performance on test set, which was 9.0 AP higher than the traditional detector using only CNNs. In detecting small or truncated clips and bolts, the self-attention mechanism helped to capture local context to better localize and size them, which resulted in at least 9.2 AP higher accuracy than the baseline. Our proposal made a lot of efforts on reducing the computational complexity and memory capacity, its inference speed was comparable to Faster RCNN models but still significantly slower than YOLOv2. The experiments revealed the limitation of using the encoder in transformer structure and proved that a shallow decoder was enough to realize good detection effect, which also indicated a reduction of the model complexity. With the detection framework simplified, we found a balance between the global and local context as well as achieved better performances on both of the accuracy and speed.

Besides the successful detection for truncated clips, the potential of convolutional transformer network on benefiting other rail track condition monitoring applications via analyzing railway images needs to be further explored. In future work, we will focus on automatic industrial inspections on small surface defects, which can benefit from the attention mechanism. Another challenge for the proposed method is further decreasing the model complexity and speeding up the detection, such that the real-time inspection is able to be implemented with videos as an input.

V. CONCLUSION

In this work, a deep convolutional transformer based method for detecting railway track components was introduced. It operated effectively to attain more advanced detection performance with limited training data. By applying multiple thresholds, our proposal achieved 61.9 AP performance on test set, which was 9.0 AP higher than the traditional detector using only CNNs. In detecting small or truncated clips and bolts, the self-attention mechanism helped to capture local context to better localize and size them, which resulted in at least 9.2 AP higher accuracy than the baseline. Our proposal made a lot of efforts on reducing the computational complexity and memory capacity, its inference speed was comparable to Faster RCNN models but still significantly slower than YOLOv2. The experiments revealed the limitation of using the encoder in transformer structure and proved that a shallow decoder was enough to realize good detection effect, which also indicated a reduction of the model complexity. With the detection framework simplified, we found a balance between the global and local context as well as achieved better performances on both of the accuracy and speed.

Besides the successful detection for truncated clips, the potential of convolutional transformer network on benefiting other rail track condition monitoring applications via analyzing railway images needs to be further explored. In future work, we will focus on automatic industrial inspections on small surface defects, which can benefit from the attention mechanism. Another challenge for the proposed method is further decreasing the model complexity and speeding up the detection, such that the real-time inspection is able to be implemented with videos as an input.
[2] Q. Li and S. Ren, "A real-time visual inspection system for discrete surface defects of rail heads," IEEE Transactions on Instrumentation and Measurement, vol. 61, no. 8, pp. 2189-2199, 2012.

[3] L. Shang, Q. Yang, J. Wang, S. Li, and W. Lei, "Detection of rail surface defects based on CNN image recognition and classification," in 2018 20th International Conference on Advanced Communication Technology (ICACT), 2018: IEEE, pp. 45-51.

[4] S. Yanan, Z. Hui, L. Li, and Z. Hang, "Rail surface defect detection method based on yolov3 deep learning networks," in 2018 Chinese Automation Congress (CAC), 2018: IEEE, pp. 1563-1568.

[5] R. A. Khan, S. Islam, and R. Biswas, "Automatic detection of defective rail anchors," in 17th international IEEE conference on intelligent transportation systems (ITSC), 2014: IEEE, pp. 1583-1588.

[6] Ç. Aştekin, Y. Rezaeitabar, S. Dogru, and I. Ulusoy, "Railway fastener inspection by real-time machine vision," IEEE Transactions on Systems, Man, and Cybernetics: Systems, vol. 45, no. 7, pp. 1101-1107, 2015.

[7] X. Gibert, V. M. Patel, and R. Chellappa, "Robust fastener detection for autonomous visual railway track inspection," in 2015 IEEE Winter Conference on Applications of Computer Vision, 2015: IEEE, pp. 694-701.

[8] Y. Y. Xia, F. Xie, and Z. Jiang, "Broken railway fastener detection based on adaboost algorithm," in 2010 International Conference on Optoelectronics and Image Processing, 2010, vol. 1: IEEE, pp. 313-316.

[9] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in Advances in neural information processing systems, 2015, pp. 91-99.

[10] X. Zhou, D. Wang, and P. Krähenbühl, "Objects as points," arXiv preprint arXiv:1904.07850, 2019.

[11] T. Wang, F. Yang, and K.-L. Tsui, "Real-Time Detection of Railway Track Component via One-Stage Deep Learning Networks," Sensors, vol. 20, no. 15, p. 4325, 2020.

[12] A. Vaswani et al., "Attention is all you need," in Advances in neural information processing systems, 2017, pp. 5998-6008.

[13] X. Gibert, V. M. Patel, and R. Chellappa, "Deep multitask learning for railway track inspection," IEEE Transactions on Intelligent Transportation Systems, vol. 18, no. 1, pp. 153-164, 2016.

[14] X. Gibert, V. M. Patel, and R. Chellappa, "Deep multitask learning for railway track inspection," IEEE Transactions on Intelligent Transportation Systems, vol. 15, no. 2, pp. 760-770, 2013.

[15] M. Kisantal, Z. Wojna, J. Murawski, J. Naruniec, and K. Cho, "Augmentation for small object detection," arXiv preprint arXiv:1902.07296, 2019.