Multimedia Tools and Applications (2024) 83:10367–10392
https://doi.org/10.1007/s11042-023-15924-7

FishTwoMask R-CNN: Two-stage Mask R-CNN approach for detection of fishplates in high-altitude railroad track drone images

Aradhya Saini[1,2] · Dharmendra Singh[1,2] · Mauricio Alvarez[3]

Received: 20 August 2022 / Revised: 6 May 2023 / Accepted: 29 May 2023 /
Published online: 21 June 2023

Abstract
Maintenance of railroad track safety is of utmost importance as derailment accidents cause significant loss to life and property. Inspection of railroad tracks and their components is necessary in order to ensure security and well-being of goods as well as humans. Fishplate is an essential component in the railroad track environment hence, periodic maintenance of fishplates is an imperative goal. In this paper, we propose a method for detection and segmentation of fishplate instances in high-altitude drone images (DI) for a closer-view and consequent inspection of fishplate instances. For this purpose, a novel two-stage Mask R-CNN-based framework termed as FishTwoMask R-CNN is proposed. A new fine-tuning strategy has been developed for the purpose of improving the detections in the second stage (Stage 2) which includes a training trick of modifying the loss weights for Stage 2 training. In the first stage (Stage 1), we detect fishplate instances, which are then cropped and fed as input to Stage 2, along with Stage 1 dataset. The Stage 2 network is then trained through a modified weighted loss and produces final detections for segmentation and further inspection. The"layers" hyper-parameter is assigned as "heads" for Stage 1 and updated to "4+" for Stage 2. Also, the critical analysis of Mask R-CNN hyper-parameters has been carried out during both the stages which has lead to an improved detection precision rate of 97% in Stage 2 as opposed to 47% in Stage 1. We evaluate our proposed approach on five different test image scenarios in order to view fishplate instance detection results. There has been statistical evaluation on out-of-distribution test images also in order to compute the metrics values. The comparative results have been evaluated using metrics of precision, recall, and F1-score on Mask R-CNN Stage 1 and Stage 2 along with Faster R-CNN and YOLOv5 methods. It is inferred that the proposed approach achieves appreciable metrics values and thus can be gathered suitable for fishplate instance segmentation in drone images.

Keywords Instance segmentation · Drone images · Fishplate instances · Railroad track · Faster R-CNN · Mask R-CNN · YOLOv5

✉ Dharmendra Singh
dharmfec@gmail.com

[1] Department of Electronics and Communication Engineering, Indian Institute of Technology, Roorkee, Roorkee, Uttarakhand, India

[2] RailTel - IIT Roorkee Center of Excellence in Telecommunication, Roorkee, Uttarakhand, India

[3] Department of Computer Science, The University of Sheffield, Sheffield S10 2TN, UK

1 Introduction

Rail transportation networks are one of the primary modes of commuting all over the world. Periodic monitoring of railroad tracks and their components is necessary in order to ensure safety of passengers and goods in railroad environments. Fishplates are critical components in railroad track infrastructure. The inspection of fishplates is an important measure in maintaining railroad track safety. A fishplate, also referred as splice or rail joint bar, is a metal bar, which is bolted to the ends of two rail lines in order to join them together, as shown in Fig. 1d, 2d, 2e and 3d [3]. The malfunction of fishplates happens due to either loosening of nuts or bolts, cracking of the fishplates, or incorrect maintenance or tampering [13]. This may cause misalignment of track sections potentially leading to catastrophic failures such as derailment of trains [13].

There has been an advent of various sensing or image acquisition systems (IAS) along with machine learning and computer vision-based methods for fishplate monitoring. In [20] an IOT-based real-time fishplate monitoring module comprising of electrical pulse generator (EPG) and GSM-based systems is proposed. This proposed automated system monitors the condition of the railroad fishplate bolts in order to send warning signals and avoid accidents. The fishplate peak is distinguished in [7] while a sensor, developed from Turck sensor, is fixed under a wheeled car for collecting magnetic signals. Fishplate localization is performed in images captured through rail imaging system while travelling along rail [8]. Fiber Optic Sensing (FOS) has been used in [4] for structural health monitoring of fishplates.

However, these aforementioned IAS have variable limitations, which include low strain sensitivity jacketed fibers, broken bare fibers, and high installation expenses in case of FOS[6] along with inaccessibility for remote geographical locations, limited detection range and high cost in case of rail-mounted vehicles or inspection trains [11, 26]. Also, records obtained through human inspectors are subjective and irregular. In such scenarios drone-based image acquisition systems are very beneficial. Drones are the latest trend for railroad environment monitoring as they offer various advantages such as ease of control, cost-effectiveness and flexibility while aiming inaccessible areas [26]. They are lightweight Unmanned Aerial Vehicles (UAVs) which also score over human inspectors, rail-mounted vehicles or inspection trains as they provide efficient track image acquisition without railroad traffic blockage.

Although drone-based image acquisition systems offer large number of benefits, however drone-based fishplate monitoring is observed to face the following challenges:

1. Rail lines have varying orientations as observed in drone images. Fishplates are used to join two rail lines together and orientation of rail lines affects orientation of fishplates. Thus such variances in the position and direction lead to complex fishplate detection scenarios in drone images.
2. Flying drones at various flight heights leads to capturing different sizes of fishplates in drone images due to varied views. In addition, fishplate may be misconstrued in high-altitude drone images thus making fishplate detection in drone images a difficult problem.
3. Different illumination scenarios such as partially sunny/cloudy, sunny lead to illumination inconsistencies. In addition, partial occlusion of railroad track along with shaking of the drone due to environmental factors such as wind may cause low or uneven brightness as well as contrast as observed in different railroad environments captured in DI.

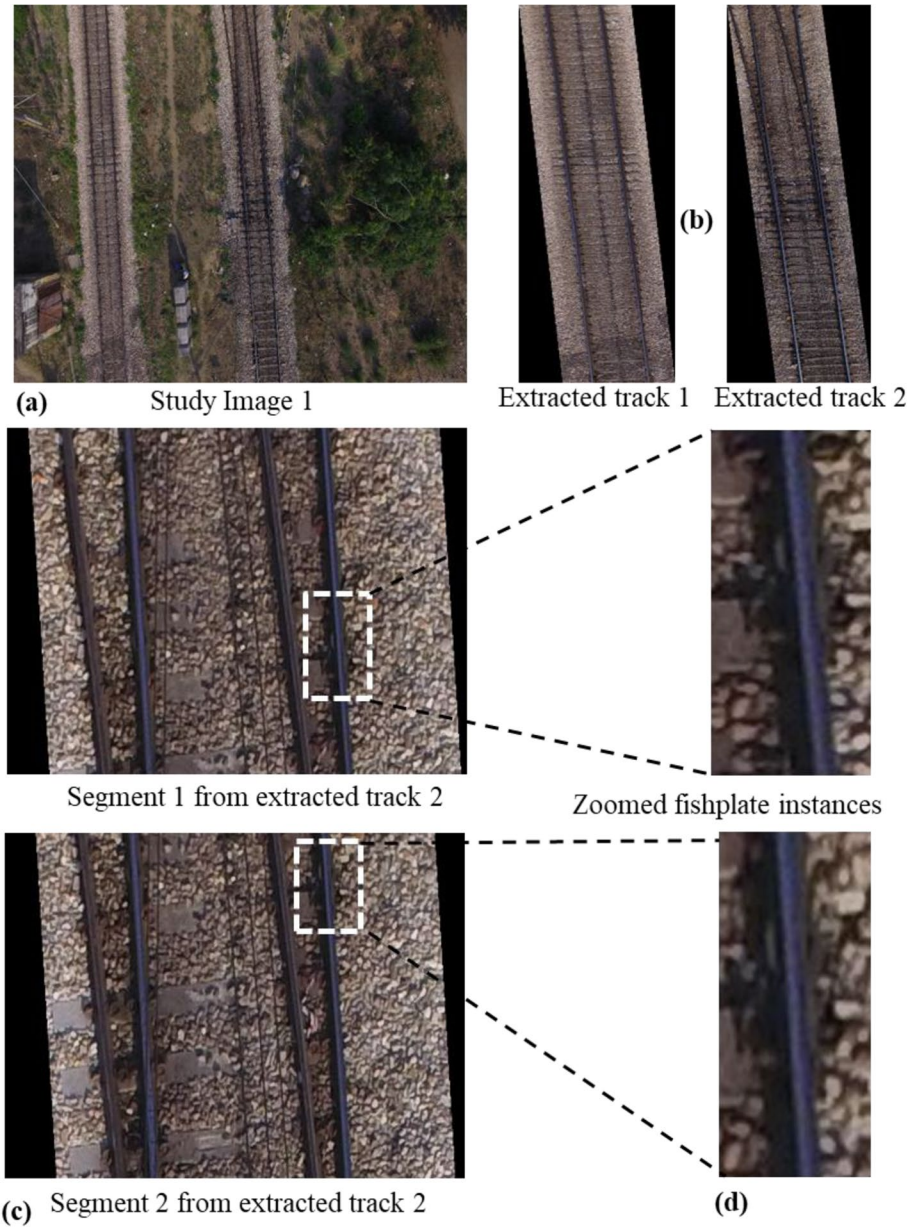

Fig. 1 Scenario 1 (**a**) Study Image 1(**b**) Extracted tracks 1 and 2 (**c**) Segments from extracted track 2 (**d**) Zoomed fishplate instances

This necessitates an adaptive method for fishplate detection in drone images.

As these challenges are observed for drone-based image acquisition, fishplate monitoring in drone images remains a difficult task. In order to overcome these shortcomings, a novel approach for fishplate detection in railroad images is proposed and termed as FishTwoMask R-CNN. The motivation lies in detection and

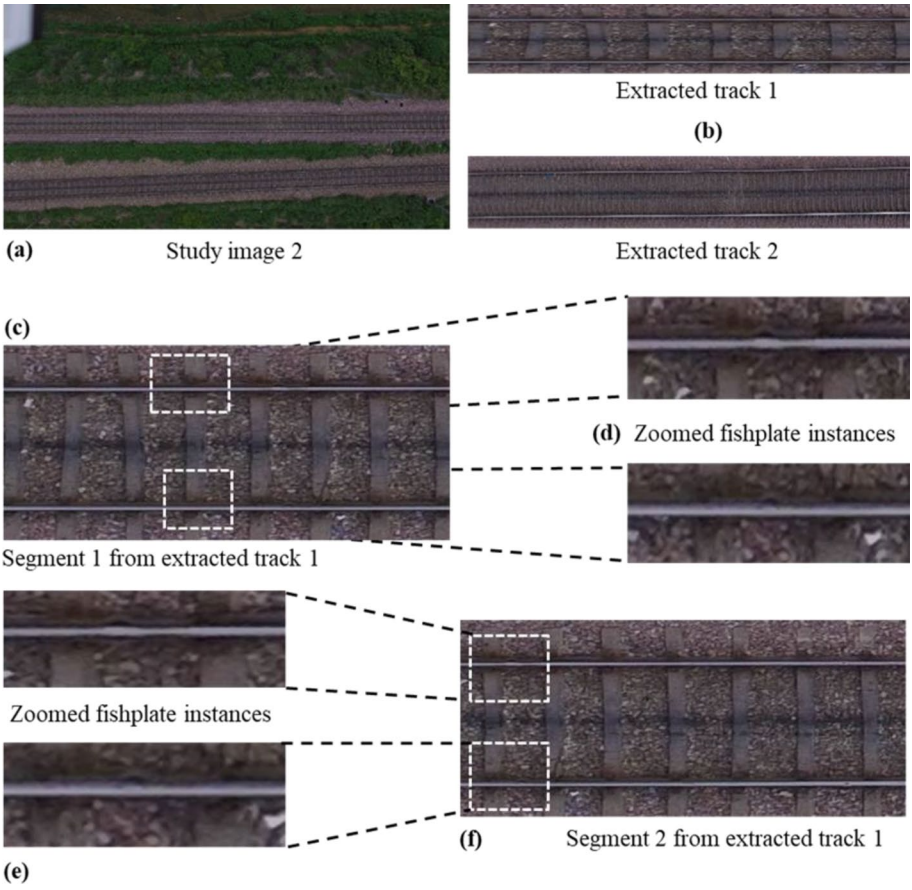

Fig. 2 Scenario 2 (**a**) Study Image 2 (**b**) Extracted Tracks 1 and 2 (**c**), (f) Segments from extracted track 1 (**d**), (**e**) Zoomed fishplate instances

segmentation of tiny fishplate instances for obtaining their closer view for inspection in high-altitude drone images. This in turn is helpful for railroad track health monitoring for the provision of safety during transportation in order to avoid mishaps, which are indicative of loss of life and property.

The organization of the article are as follows: In Sect. 2, algorithms related to detection, and health monitoring of railroad tracks and their components is discussed. The image acquisition and data generation are discussed in Sect. 3. While image acquisition comprises of study area and datasets used, data generation highlights steps for Stage 1 and Stage 2 fishplate instances dataset creation. In Sect. 4 theoretical background is presented. The proposed method, comprising of Stage 1 and Stage 2 training, is discussed in Sect. 5. The evaluation of proposed method based on experimental results is discussed in Sect. 6 along with complexity analysis. Finally, conclusion and future work in the article are presented in Sect. 7.

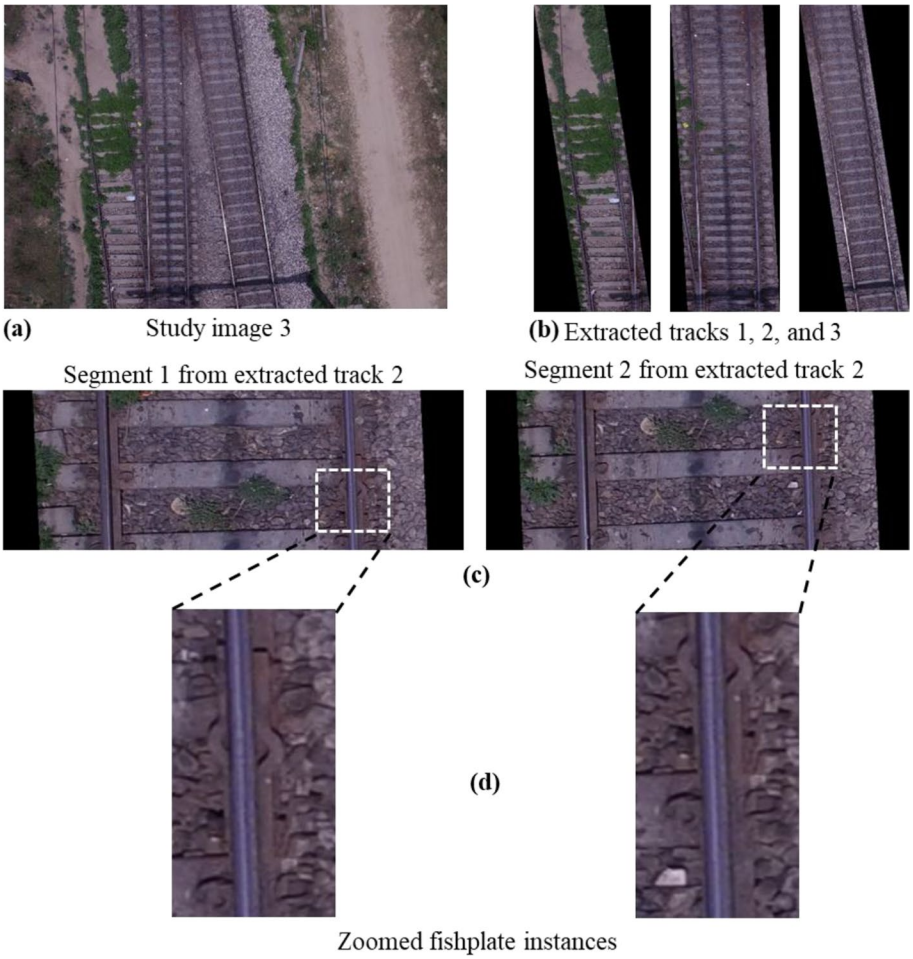

Fig. 3 Scenario 3 (**a**) Study Image 3 (**b**) Extracted Tracks 1, 2 and 3 (**c**) Segments from extracted track 2 (**d**) Zoomed fishplate instances

2 Literature review

Various optimization and machine-learning techniques have been implemented for object classification and detection tasks in different real-life applications [9, 12]. In addition, object localization is performed for various field applications in railroad environment [24, 31]. Object detection and segmentation algorithms have also been developed for detection, segmentation and inspection of railroad tracks and their components in railroad track environments [11]. The task of classifying and localizing multiple objects in an image is termed as Object detection. Semantic Segmentation can be defined as the classification of every pixel in an image labeled as the object class it belongs to. However, in this different objects belonging to the same class are undisguisable. Consequently, instance segmentation comes into picture. Instance segmentation is similar to semantic segmentation however, all objects of the same class are not merged into one big lump instead, each of the class objects are identified as a unique entity for

instance, each of the individual fishplates in a DI are distinguishable. Mask R-CNN is an efficient model for instance segmentation.

A comprehensive review on the merits and demerits of the existing works have been tabulated in Table 1. Faster R-CNN algorithms have been implemented for aerial supervision of railroad tracks in drone images [14, 21]. In [21] Faster R-CNN along with YOLO and other algorithms is developed for obstacle detection in railroad track aerial images. Faster R-CNN is also developed for small object detection and the imbalanced dataset, based on images captured using unmanned aerial vehicles (UAV) [17]. In [24] railroad track health monitoring is performed through track assets detection using YOLOv3. It is observed that Faster R-CNN tends to be more complex than Yolo hence it is slower as compared to the single shot detection method. The reason being single shot methods do not require per region processing. Mask R-CNN [12] is advantageous over YOLOv5 and Faster R-CNN. Pixel-to-pixel alignment is the key to Mask R-CNN which is otherwise missing in Faster R-CNN [29]. Additionally, Mask R-CNN unifies both object detection and semantic segmentation to perform instance segmentation.

In [33] YOLOv5 and Mask R-CNN framework have been implemented for rails and fasteners localization and rail surface defect detection and segmentation respectively. The dataset has been captured using special rail inspection vehicle. In [5] Mask R-CNN algorithm has been devised for segmentation and extraction of rail and fastener areas while the dataset images have been collected using an inspection cart. The detection of railroad components such as rail, clip and spike is performed using Mask R-CNN on track images captured using iPhone 8 smartphone in [11]. In [10] iPhone 8 smartphone is used to collect images for the goal of automatic rail surface defects detection based upon Mask R-CNN. In aforementioned studies Mask R-CNN architecture has been implemented along with data samples being collected using inspection vehicles and phones. As observed, Mask R-CNN-based architecture is well-suited and very efficient for the task of inspecting railroad tracks and their components.

In [9] Mask R-CNN is used for the detection of sleepers and spaces between sleepers in drone images. Alongside, Mask-R-CNN architecture enables segmentation in drone images during identification of healthy and missing rail fasteners [32]. Mask R-CNN is advantageous as like object detection it can handle multiple fishplate instances alongside differentiating the identities. For detection of fishplate instances in DI various image processing and statistical methods have been previously developed. In [23] feature-based template matching has been implemented for fishplate detection in DI. However, this work has been computationally extensive as large number of features have been calculated beforehand in order to select one suitable feature. The work in [25] computes fishplate detection in drone images using Normalized correlation coefficient and Non-maximum suppression. In this approach, a large number of false positives are observed. The fishplate detections are obtained in drone images captured at fixed heights, both in [23, 25]. Therefore, it can be concluded that Mask R-CNN method is explored in a limited manner for fishplate detection in railroad environment drone imagery making it seem a suitable architecture to achieve our goal.

Therefore, our aim is fishplate instance detection and segmentation for inspection purpose in drone images. These DI are captured at different flight heights, under uneven illumination and with varying rail line orientations in different railroad environments. To achieve this aim we propose the novel two-stage Mask R-CNN framework termed as FishTwoMask R-CNN and the main contributions of the proposed approach are summarized as follows:

Table 1 Comprehensive review of existing works

Reference No	Method Used	Track Component Focused	Merits	Demerits
[16]	**Faster R-CNN Inception v2 and Atrous models**	Vegetation and obstacle detection	The Inception v2 model provides good performance while Atrous provides even better performance, especially for small-scale objects	As development of a dedicated dataset for different railroad terrains is recommended, it is inferred that classification knowledge in one railroad terrain cannot be transferred to another railroad terrain
[20]	**Faster R-CNN, Yolo**	Obstacle detection	The detection models could be used for real-time obstacle detection	More datasets with diversity could be collected. A better high-end system could be used with modifications in the deep neural network models
[32]	**Faster R-CNN**	Catenary support device inspection, small object detection and imbalanced dataset	Improved Faster R-CNN achieves better performance than classic methods	Avoidance of category imbalance required
[7]	**Yolov3**	Track Assets detection	Multiple track assets are detected	Network architecture changes required for better performance
[30]	**Yolov5, Mask R-CNN**	Rail surface and fasteners defect detection	Mask R-CNN is found efficient	More data augmentation methods to expand the defect samples. Also, further improvement of the robustness of our method
[21]	**Mask R-CNN, SVDD (Support Vector Data Description)**	Foreign Objects defect detection	Detection of foreign objects in ballastless trackbed images which the algorithm has not learned; and still obtained desirable results	Need for simplification of the detection process and increase in the detection speed in the future
[3]	**Mask R-CNN**	Rail surface defects inspection	Mask R-CNN presents promising results upon comparison with Otsu taking into consideration different lighting conditions and severities	Prediction performance of the developed model maybe improved while more training data is used
[4]	**Mask R-CNN**	Detection of sleepers and spaces between sleepers	Sleepers are located, compared using two different methods based on the Otsu method and Mask R-CNN	Expansion towards detection of defects in various components alongside developed autonomous UAV

Table 1 (continued)

Reference No	Method Used	Track Component Focused	Merits	Demerits
[29]	**Mask R-CNN**	Railroad fastener fault detection	Achieved commendable results for fastener fault detection	The drone images are closely captured
[13]	**Handcrafted Features**	Fishplate detection	Promising results with lesser number of false detections	Computationally intensive
[1]	**Normalized Correlation coefficient, Template matching**	Fishplate detection	Less computationally intensive	Large Number of false alarms (false positives)

1. The work proposes a novel two stage Mask R-CNN framework termed as FishTwoMask R-CNN for fishplate instances detection and segmentation in high-altitude drone images. This implies working with a tiny railroad component such as a fishplate within a large drone image. This algorithm is adaptive due to its ability to detect fishplate instances in drone images captured under different railroad environments. Our method is a hierarchical approach with only two Mask R-CNNs.
2. A new fine-tuning strategy has been proposed for the improved detection of the fishplate instance in drone images. This includes a training trick of modifying the loss weights for the second stage of training (Stage 2) in order to reach the top-performing level in the network.
3. The devised training method also comprises of changing the 'layers' hyper-parameter in the architecture while training in Stage 1 and Stage 2. Additionally, the cropped fishplate instances from the first stage are incorporated alongside the Stage 1 dataset for the second stage of training.

In order to achieve the goal of fishplate instance segmentation for railroad track safety monitoring in drone images, the images are acquired and the dataset for the two stages is generated as discussed in Sect. 3.

3 Image acquisition and data generation

3.1 Study area and datasets used

The images are acquired for generation of both Stage 1 and Stage 2 datasets. In this work, drone-based image acquisition has been performed using DJI phantom quadcopter. The drone specifications comprise of high definition 4 K resolution RGB colour camera along with GPS unit in order to capture geotagged standardized RGB (sRGB) images. Each image is of size 3000×4000 pixels. The total number of drone images in the acquired datasets is equivalent to 215. These drone images are captured over railroad tracks spanning across different track locations near Roorkee, Haridwar, India. These drone images are acquired frame by frame and comprise of fishplate instances captured at different locations, different flight heights, varied dates/ time and in uneven illumination as observed in complex railroad environments.

Some of the fishplate instances in drone images are as shown in Fig. 1d, 2d, 2e and 3d. The description of datasets,used as source for Stage 1 and Stage 2 datasets creation, is as mentioned in Table 2 and includes SID (sample ID), DOA(date of acquisition), central

Table 2 Datasets Description

SN	SID	F_h(m)	DOA	Central Lat/Long	GSD (cm)	Illumination Scenario	No. of images in dataset
(1)	S_{1mp}	22 m	10–05-2018	29°51′0.6129″N/ 77°52′50.9917″E	0.89	Sunny, Little occlusion	56
(2)	S_{2mp}	25 m	07–07-2017	29°51′14.9380″N/ 77°52′5.8300″E	1.06	Dark	63
(3)	S_{3mp}	11 m	25–04-2017	29°46′3.3522″N/ 78°0′35.0221E	0.47	Little bright	96

latitude and longitude, flight height (F_h), GSD(ground sample distance) [26], illumination scenarios and total number of images present in the dataset. GSD is indicative of how big each pixel is on the ground [26]. The changes in flight height (F_h) lead to changes in corresponding pixel size which consequently changes number of pixels between two rail lines of a rail line pair. Therefore, the calculated pixels are essential during rail line pair selection. The notationsS_{1mp}, S_{2mp} and S_{3mp} denote IDs of *p th* fishplate instance segment acquired from the *m th* extracted track of their respective *i th* drone image D_i acquired in datasets for train and test purpose for Stage 1 and Stage 2. The method used for railroad track extraction from railroad track images is DroneRTEF as proposed and discussed in [26]. Fishplate instances from three different datasets are described as follows:

Scenario 1: Sample instance(s) $p\left(S_{1mp}\right)$, segmented from extracted tracks E_m of Study Image 1 from dataset D_1, are as shown in Fig. 1a. Study Image 1 is captured on a sunny day at 4:46 p.m. at 22 m flight height. A little amount of occlusion is observed in the image. The extracted tracks (Tracks 1 and 2) are as shown in Fig. 1b. The segments from extracted track 2 can be viewed in Fig. 1c while the fishplate instances can be viewed in Fig. 1d.
Scenario 2: Sample instance(s) $p\left(S_{2mp}\right)$, segmented from extracted tracks E_m of Study Image 2 from dataset D_2, are as shown in Fig. 2a. These fishplate instances are segmented from Study Image 2 which is captured at 2:51 p.m. and at a flight height of 25 m in low brightness (dark) environment. The extracted tracks 1 and 2 can be viewed in Fig. 2b. The segments from extracted track 1 can be viewed in Fig. 2c, 2f while Fig. 2d, 2e depict zoomed fishplate instances.
Scenario 3: Sample instance(s) $p\left(S_{3mp}\right)$, segmented from extracted tracks E_m of Study Image 3 from dataset D_3, are as depicted in Fig. 3a. The Study Image 3 is captured at 11 m at 11:53 a.m. on a little brighter day. The extracted tracks 1, 2 and 3 are observed in Fig. 3b while Fig. 3c depicts segments from extracted track 2. Figure 3d shows zoomed fishplate instances.

3.2 Data generation

The datasets acquired in Sect. 3.1 form the data acquisition module of the FishTwoMask R-CNN architecture. The input images in the datasets are of importance as correct representation of fishplate ground truth images is helpful in developing Stage 1 and Stage 2 datasets as discussed in Sect. 3.2.

3.2.1 Stage 1 Dataset description

The drone images D_i undergo various preprocessing steps in order to form Stage 1 fishplate instances dataset. The Stage 1 dataset creation flowchart is represented in Fig. 4 and the same is explained algorithmically in Fig. 6. Drone-based image acquisition is carried out for obtaining t drone images $\left(D_t\right)$ of the railroad environments. The g railroad tracks are then extracted from the acquired images [26]. The segmentation of each of these extracted railroad track images is performed into a total of n overlapping smaller sized railroad track segmentsS_{imp}. The railroad track segments S_{imp} with fishplate instances 'fishplate' are then selected to form dataset $Dataorig_1$ which comprises of segments with fishplate samples, as

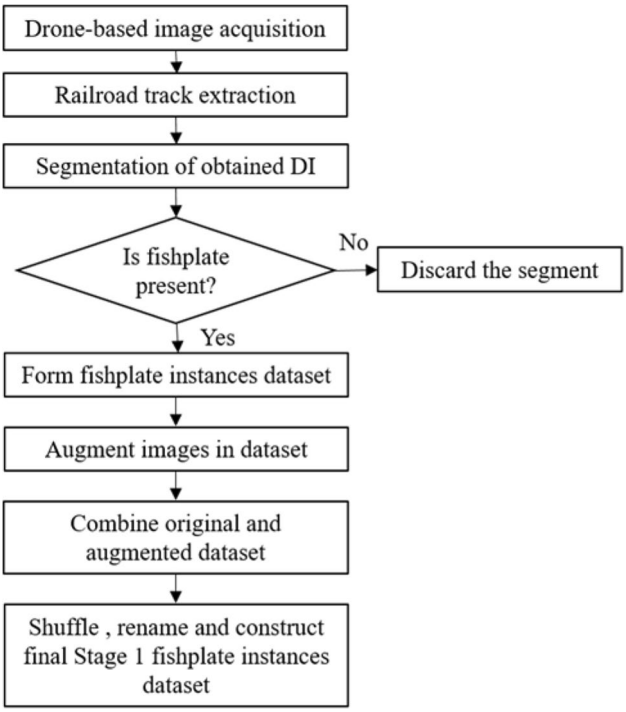

Fig. 4 Description of Stage 1 dataset creation

discussed in Step 3- Step 7 (Fig. 6). The matching criteria of fishplate in the S_{imp} is checking for the presence of fishplate though visual inspection of these ground truth segments. The has '*fishplate'* in Step 6 of Algorithm 1 (Fig. 6) indicates whether fishplate instance(s) are present in S_{imp} and can be viewed through naked eyes in the segment. Consequently, the S_{imp} are examined and if fishplate instance(s) are found in the S_{imp} segments then those corresponding S_{imp} are selected and added to $Dataorig_1$ as discussed in Step 7 (Fig. 6).These $Dataorig_1$ segments are then augmented for obtaining new images, as mentioned in Step 8 (Fig. 6). These new segment images in the augmented dataset Aug_1 exhibit a variety of conditions for fishplate instances such as different brightness, scales, locations, and orientations. As depicted in Step 9 (Fig. 6) the functions of shuffle and rename are performed on the combined original $Dataorig_1$ and augmented Aug_1 datasets in order to obtain Stage 1 fishplate instances dataset$Data_1$.

3.2.2 Stage 2 Dataset description

The requirement for Stage 2 dataset arises as, after Stage 1 training, incorrect detections are observed (discussed in Sect. 5.1). Consequently, for Stage 2, we have developed the dataset from Stage 1 fishplate instances dataset $Data_1$ and termed it as $Data_2$. The pictorial depiction is provided in Fig. 5 along with the algorithm, which is as discussed in Fig. 7. In Fig. 7 $Data_1$ is considered as the input. In Step 2 (Fig. 7), the resultant bounding boxes obtained as outputs from $Data_1$ in Stage 1 training, are used to crop the instances from the corresponding segments. The top left coordinates (x, y) of the bounding box along with the

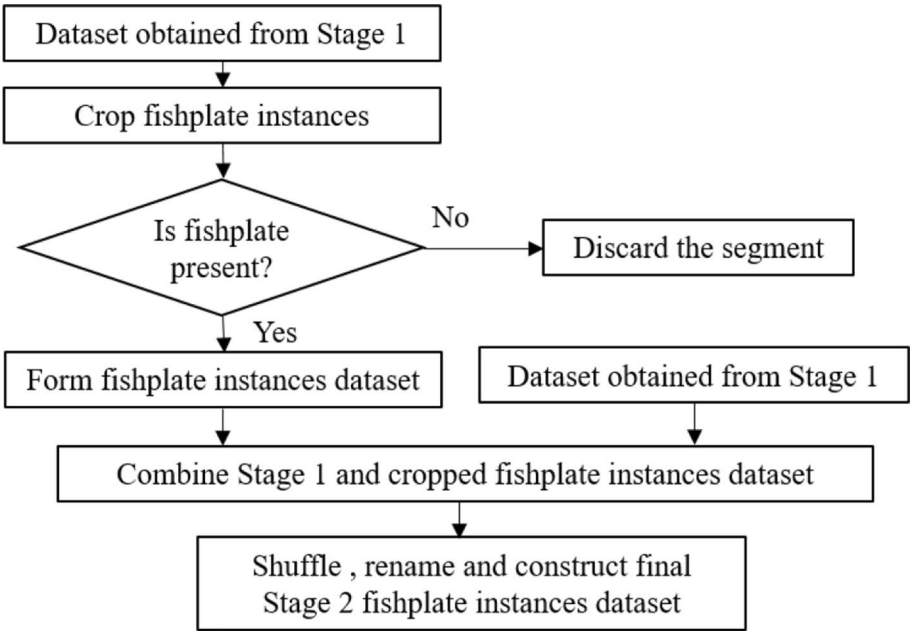

Fig. 5 Steps for Stage 2 dataset creation

width w, height h are required for cropping using $[x, y, x + w, y + h]$. This forms the $fishinst_1$ dataset. The cropping step is of significance as in order to get more context for the fishplate instances for better analyses, our approach includes first localizing the fishplate instances in Stage 1 segments and cropping around them, then using these cropped portions of the segments for Stage 2.

These cropped images in $fishinst_1$ dataset are evaluated for fishplate instances *'fishplate'*. If this image comprises of the fishplate instance, it is added to the fishplate instances dataset $fishinstdata_1$ or else it is discarded. The has *'fishplate'* in Step 3 of Algorithm 2 (Fig. 7) indicates whether a visible fishplate instance structure is present in the cropped image $fishinst_1$. The fishplate presence is checked again in Step 3 as $fishinst_1$ is sourced from $Data_1$.

As $Data_1$ is divided into test and train images therefore it is important to check presence of fishplate instances in $fishinst_1$ images especially in those sourced from test images. The $fishinstdata_1$ is then combined with the Stage 1 fishplate instances dataset $Data_1$. In Step 5 (Fig. 7) combining $Data_1$ **&** $fishinstdata_1$ indicates merging both the datasets together in order to apply the functions of shuffle and rename onto this combined fishplate instances dataset. This forms Stage 2 fishplate instances dataset $Data_2$, which is then fed to Stage 2 for training.

4 Theoretical background

The goal of this work is fishplate instance detection and segmentation for fishplate instance monitoring purpose and this is facilitated through development of a two-stage Mask-RCNN method while a brief review on the network is discussed here in Sect. 4.

Algorithm 1: Data Generation for Stage 1

Input: Drone image D_i

Output: Fishplate instances dataset $Data_1$

1. $Dataorig_1 \leftarrow \varnothing$
2. Obtain segments S_{imp} from extracted track E_m of D_i
3. *for* $i := 1 : t$
4. | *for* $m := 1 : g$
5. | | *for* $p := 1 : n$
6. | | | *if* S_{imp} has *'fishplate'*
7. | | | | *Add* S_{imp} to $Dataorig_1$
8. Create augmented dataset Aug_1
9. Construct $Data_1$ from $Dataorig_1$ & Aug_1

Fig. 6 Algorithm for Stage 1 Data Generation

Mask R-CNN has been developed by the Facebook AI Research group in 2017 [12]. Mask R-CNN extends Faster R-CNN by adding a branch for prediction of the high-quality segmentation mask for each instance in conjunction with the existing branch for bounding box regression and classification. The feature extraction is performed using the backbone network which is ResNet. The backbone is followed by a Region proposal network (RPN) along with two head branches. One of them is for the bounding box regressor while the other is subjected to classification. The added branch takes into account the region of interest (ROI), extracted using RPN, in a fully convolutional network in order to predict an instance mask for the ROI. Mask R-CNN has a complex loss function calculated as the weighted sum of different losses as represented in Eq. 1. The weights of the network are adjusted accordingly during training while the total loss is being calculated as:

$$\begin{aligned} loss = {} & w_1 \times rpn\_class\_loss + w_2 \times rpn\_bbox\_loss \\ & + w_3 \times mrcnn\_class\_loss + w_4 \times mrcnn\_bbox\_loss \\ & + w_5 \times mrcnn\_mask\_loss \end{aligned} \tag{1}$$

In Eq. 1, the RPN class loss *rpn_class_loss* and bounding box loss *rpn_bbox_loss* are in relation with the output of the RPN. The RPN class loss, a binary classification loss, is assigned to improper classification of anchor boxes by RPN and is positive incase intersection over union (IoU) between proposed region and the ground truth(GT) bounding box is > 0.5. This is to be increased incase multiple fishplate detections do not happen in final

Algorithm 2: Data Generation for Stage 2

Input: $Data_1$

Output: Fishplate instances dataset $Data_2$

1. $fishinstdata_1 \leftarrow \varnothing$
2. Crop instances from $Data_1$ to form $fishinst_1$
3. *if* $fishinst_1$ has *'fishplate'*
4. | *Add* $fishinst_1$ to $fishinstdata_1$
5. Combine $Data_1$ & $fishinstdata_1$
6. Construct $Data_2$ from Step 5 data through shuffle and rename

Fig. 7 Algorithm for Stage 2 Data Generation

output. The RPN bounding box loss is equated as a regression loss between the four corner points of the GT bounding box and proposed region bounding box. This weight is tuned in case the bounding box needs to be corrected. The class loss *mrcnn_class_loss* is assigned to improper classification of fishplate present in region proposal. The weight corresponding to bounding box loss *mrcnn_bbox_loss* is increased if correct classification of fishplate class is attempted however, precise localization is not achieved. For *mrcnn_mask_loss* the corresponding mask loss weight is increased if pixel level identification of fishplates is of importance. For Stage 2 the mask loss and bounding box loss weights are emphasized upon which is helpful in boosting the performance during training.

Recall that our main objective is fishplate inspection in high-altitude drone images. This accounts to segmentation of fishplate instance in the railroad track drone images. The size of the fishplate is very small compared to the drone image, hence making fishplate instance segmentation hard. Using Mask R-CNN on the drone image segment will give pixel wise decision regarding classification into fishplate instance or not. For this purpose, we have designed the proposed method in Sect. 5.

5 Proposed method and implementation

The advantage of the Mask R-CNN in FishTwoMask R-CNN architecture is that it provides with both the bounding box and semantic segmentation. Thus, this caters to a multi-stage approach for the purpose of semantic segmentation using the same architecture. Mask R-CNN [12] implementation, performed by Matterport [1], has been the core of our experiments. The code is implemented using Tensorflow framework (version 1.12.0) along with Keras (version 2.2.4) from Google and it has been run on an HP Z8 G4 Workstation with NVIDIA Quadro P5000 graphics card. The original implementation of Mask R-CNN uses a fixed learning rate (lr) equivalent to 0.02 which is decreased to 0.002(towards the end) along with a weight decay of 0.0001. In the Tensorflow implementation by Matterport lr of 0.001 is set with a comment stating weight explosion might be due to optimizer differences. The proposed method is trained using hyper-parameters described in Table 3 for both stages. Alongside, "layers" hyper-parameter is assigned as 'heads' for Stage 1 and '4+' for Stage 2 respectively. The "layers" hyper-parameter allows selecting which layers to train. Consequently, 'heads' denotes training the RPN, classifier and mask heads of the network(all layers but the backbone) for Stage 1 while '4+' is equivalent to training Resnet stage 4 and up for Stage 2. The Mask R-CNN hyper-parameter evaluation has been performed on a high performance computing (HPC) system. Varied values of Mask R-CNN hyper-parameters such as loss weights

Table 3 Training hyper-parameters of Stage 1 and Stage 2 Mask R-CNN in Proposed architecture

Parameters/ models	Stage 1 Mask R-CNN	Stage 2 Mask R-CNN
Learning rate	0.001	0.001
Momentum	0.9	0.9
Weight Decay	0.0001	0.0001
Steps	1000	1000
Loss weights	[1,1,1,1,1]	[1,1,1,1.5,2]
Backbone/base	Resnet101	Resnet101
Train images	835	1794
Test images	206	205

(*rpn_class_loss*, *rpn_bbox_loss*, *mrcnn_class_loss*, *mrcnn_bbox_loss and mrcnn_mask_loss*), weight decay, learning rate, momentum, backbone, steps per epoch need to be evaluated in parallel for obtaining optimum results on the training network. These values need to be chosen from a list for instance the *mrcnn_mask_loss* is evaluated on a list of values [1, 1.5, 2]. Similar evaluations are performed by providing different set of hyper-parameters, in parallel.

The Stage 1 comprises of analyzing the fishplate instance segments in their original view/ scale, and the Stage 2 focuses on the fishplate instances upon cropping, along with Stage 1 dataset, for further analysis. The flowchart is depicted in Fig. 8 and the algorithm is discussed in Fig. 9. In the proposed algorithm $Data_1$ is obtained as input for Stage 1 training and Mask R-CNN model is computed, as discussed in Sect. 5.1. The ***fishplateloc*** is obtained for evaluation of $mask_i$. The incorrect detections are observed on test images in Stage 1 training (discussed in Sect. 5.1). Therefore $Data_2$ is generated (discussed in Sect. 3.2.2) and computed upon by fine-tuned and training trick modified Mask R-CNN in the Stage 2 training (discussed in Sect. 5.2). This step outputs ***fishplateloc,*** $mask_j$ and determines if there is a missed, false or no detection. The two stages of the architecture are as discussed in Sect. 5.1 and 5.2 respectively.

5.1 Stage 1 Training

The flowchart for the proposed method FishTwoMask R-CNN is shown in Fig. 8. The three datasets used for datasets generation, as discussed in Table 2, are obtained under different railroad track environmental scenarios hence different distributions are considered. In Stage 1 the Mask R-CNN network, pre-fit on the COCO dataset, is used as a starting point and then the weights are tuned on Stage 1 fishplates instances $Data_1$ dataset using transfer learning. The Mask R-CNN loss weights w_1,w_2,w_3,w_4 and w_5 are initialized with default

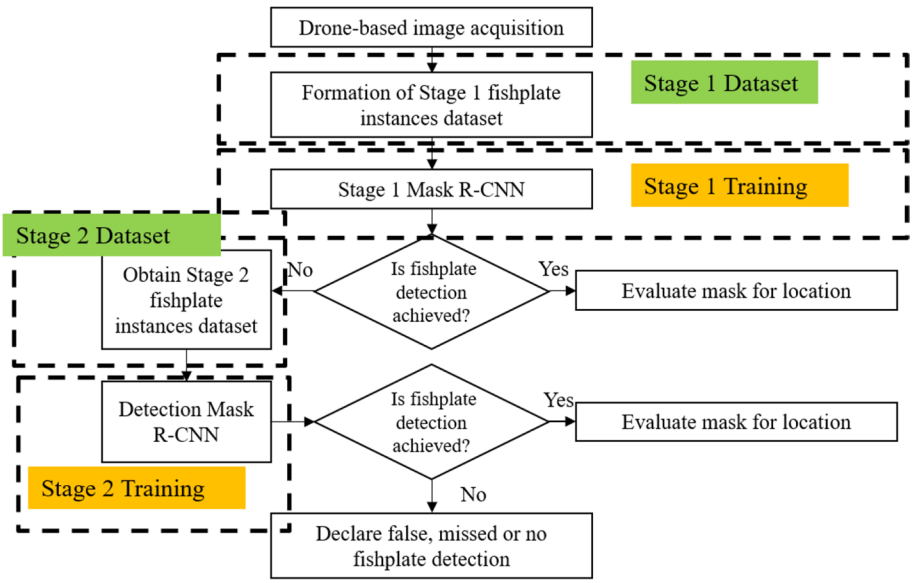

Fig. 8 Flowchart for proposed approach FishTwoMask R-CNN

values of 1,1,1,1 and 1 respectively. The 'heads' "layers", comprising of the RPN, classifier and mask heads of the network, is trained during this stage alongside other hyperparameters as discussed in Table 3 for Stage 1. In Step 2 –Step 3 (Fig. 9) Stage 1 produces masks $mask_i$ for fishplate instances in segments if ***fishplateloc*** is detected as a result. This indicates fishplate instances locations, using bounding boxes, might have been detected in segments. The evaluation of $mask_i$ is then performed statistically and visually. The results of the evaluation are indicative of performance of Stage 1 training. To obtain these results the experiments have been performed and discussed in Sect. 6.1 for visual analysis, while metrics evaluation for Stage 1 is discussed in Table 5 (Sect. 6.2). Incorrect detections are observed while trained model is tested on new test images, which are outside the distribution of the dataset images used in the training. It can be observed that the detections produce false alarms. Therefore, the resulting bounding boxes from this stage are used to crop the portions around the fishplate instances. These cropped images are then fed to the second stage for Stage 2 training.

5.2 Stage 2 Training

Assessment of the segmentation is performed upon evaluation of the performance of the Stage 1 model on the test set images. This is conducted through analysis of experiments and evaluation metrics values obtained in the Stage 1 as discussed in Sect. 6.1 and Table 5. This helps us determine if the extra steps of cropping the images alongside training another model (for Stage 2) are worth the effort. From the experiments discussed in Sect. 6.1 and evaluation in Table 5, it is observed that fishplate instances may not have been correctly detected by the network in Stage 1. In Stage 2, we are more focused about the correct localizations, no missed detections and the pixel-level identification. After the localization of the fishplate instances in Stage 1, the bounding boxes are used for cropping the fishplate instances portions for the purpose of analysis in Stage 2 training. The padding with extra pixels is performed on each side of the bounding box before using it, in order to crop the image to get more information about the fishplate, for better analysis. The bounding box of the fishplate instances helps us access the accuracy of the localization in Stage 1 and Stage 2.

The Stage 1 network investigates the entire drone image segment in order to locate and segment around fishplate instances, as discussed in Sect. 5.1. For Stage 2, we primarily take into account Stage 1 fishplate instances dataset along with $Data_1$ and $fishinstdata_1$

Algorithm 3: Proposed Approach

Input: $Data_1$

Output: Fishplate instances location ***fishplateloc*** on test images

1. Compute Mask R-CNN
2. ***if fishplateloc*** achieved
3. | ***Evaluate*** $mask_i$ for location
4. ***Else*** obtain $Data_2$
5. Compute Mask R-CNN
6. ***if fishplateloc*** achieved
7. | ***Evaluate*** $mask_j$ for location
8. Else report ***missed, false or no detection***

Fig. 9 Proposed Architecture Algorithm

Table 4 Training hyper-parameters for YOLOv5 and Faster R-CNN

Parameters/ models	Learning rate	Momentum	Weight Decay	Train images	Test images
Yolov5	0.01	0.9	0.0005	1372	228
Parameters/ models	Learning rate	Images per batch	Steps	Train images	Test images
Faster R-CNN	0.001	4	1000	1360	213

to form $Data_2$(discussed in Sect. 3.2.2). Therefore, for Stage 2, the loss weights are tuned on Stage 2 dataset $Data_2$. The Mask R-CNN in Stage 2 has also been pre-fit on COCO dataset for the starting point while the modified loss weights are then trained on Stage 2 dataset $Data_2$. To test the hypothesis, the weights associated with the bounding box and mask which are w_4, and w_5 respectively, as in Eq. 1, are increased to 1.5 and 2.0 from 1.0 and 1.0 respectively while the remaining w_1,w_2,w_3 weights are set to 1.0(default value). The bounding box weight w_4 needs to be increased as the detection process requires precise and correct localization of fishplate samples in image segments with training '4+' stage. The bounding box weight is increased as the detection process is dependent on finding the proper bounding box around the fishplate instance. Therefore, there is a slight increase in order to locate the bounding box and mask $mask_j$ first before attempting the segmentation. It is hypothesized that fishplate samples are visually structurally similar across datasets and pixel level identification is important, so emphasizing on the mask loss weight w_5 while training would help in boosting model performance. The rest of the hyper-parameters for Stage 2 are as discussed in Table 3. In Step 6-Step 7 (Fig. 9) Stage 2 training model result $mask_j$ is evaluated visually through experiments that have been performed and discussed in Sect. 6.1 and statistically in Table 5 (Sect. 6.2). The visual results for Stage 2 in Fig. 11(Sect. 6.1) show promising detections and can hence be proved through statistical evaluation of Stage 2 trained model as shown in Table 5 (discussed in Sect. 6.2).

The visual description for fishplate instance segmentation in proposed approach is as shown in Fig. 10. The fishplate instance segment dataset $Data_1$ is inputted in Stage 1, Stage 1 training is conducted using initial weights as trained on this dataset (discussed in Sect. 5.1) and the located masks$mask_i$,for ***fishplateloc,*** are obtained on the segment as shown in blue and red color. The experiments are then performed for evaluation of

Table 5 Evaluation of metrics on 80 test images (40 test images each for fishplate and no fishplate instances)

Metrics	1st Stage		2nd Stage		
	50W8E1.0L	59W8E1.0L	60W32E1.5L	60W34E1.0L	100W37E1.5L
Accuracy	0.504	0.530	0.444	0.466	0.868
Precision	0.452	0.473	0.402	0.429	0.975
Recall	0.840	0.860	0.820	0.796	0.780
F1 score	0.587	0.610	0.539	0.557	0.867
Read stage abbreviation as below					
50W8E1.0L : 50.h5 ***W*** *eight file from 8th* ***E*** *valuation with bounding box* ***L*** *oss equal to 1.0*					

obtained masks in Stage 1 model. Upon evaluation it is observed that Stage 2 training is required hence the bounding boxes obtained from these Stage 1 segments, are used for cropping fishplate instances segments, which along with $Data_1$ are used to create $Data_2$ for Stage 2 (as discussed in Sect. 3.2.2). It is observed in Stage 2 that upon Stage 2 training (as discussed in Sect. 5.2), the $mask_j$ for ***fishplateloc*** is depicted in red color. Thus, we obtain the fishplate instance locations through masks in FishTwoMask R-CNN which can be used for final segmentation and further inspection of fishplate instance component.

6 Experimental results and discussion

As discussed in Sect. 5, the detection and segmentation task of fishplate instances in FishTwoMask R-CNN comprises of two stages: Stage 1 is localizing the fishplate instances in the image segments and cropping these instances while Stage 2 takes as input the Stage 1 dataset and cropped images in order to determine the correct and precise fishplate instance locations. This two stage process in FishTwoMask R-CNN for fishplate instance detection process is as shown in Fig. 10. The evaluation of the proposed method is as shown in Sect. 6.1 along with discussion on metrics in Sect. 6.2. In Sect. 6.1 test images from new datasets, apart from existing test images, have also been considered for test purposes. The description for these datasets is provided in the respective experimental scenarios in Sect. 6.1.

6.1 Evaluation of proposed method

The test images have been randomly selected from the image sets and the results are as depicted in Fig. 11**.** To a great extent the results reflect the field performance for Stage 1 and Stage 2 in the proposed architecture. The test images represent different scenarios with drone image segments extracted from DI captured at varied illumination scenarios, flight heights, and orientations. The training for Stage 1 and Stage 2 Mask R-CNN methods in the proposed architecture FishTwoMask R-CNN is as discussed in Sect. 5.1 and Sect. 5.2 respectively. To assess the performance of the proposed method, we have used three different evaluation metrics: Precision, Recall, F-1 score [30]. For the purpose of evaluation, test images from five different railroad environmental scenarios have been discussed in Sect. 6.1.1, 6.1.2, 6.1.3, 6.1.4, 6.1.5 respectively. The description of the new test images has been provided in the respective Sections (Sect. 6.1.2, Sect. 6.1.3 and Sect. 6.1.5).

6.1.1 Experimental Scenario 1

The test image Fig. 11a has been captured as per Scenario 1, discussed in Sect. 3.1, with the central latitude/longitude coordinates of 29°51′0.2884″N/ 77°52′52.8059″E respectively. The Stage 1 test image output indicates presence of fishplate at the correct location masked with blue color, along with two false detections for fishplate locations masked in green and red as shown in Fig. 11b. The two false detections are rail line areas having no fishplate instances. However, at Stage 2 in FishTwoMask R-CNN the test image output in Fig. 11c indicates elimination of false detections while outputting only correct fishplate location marked in red color.

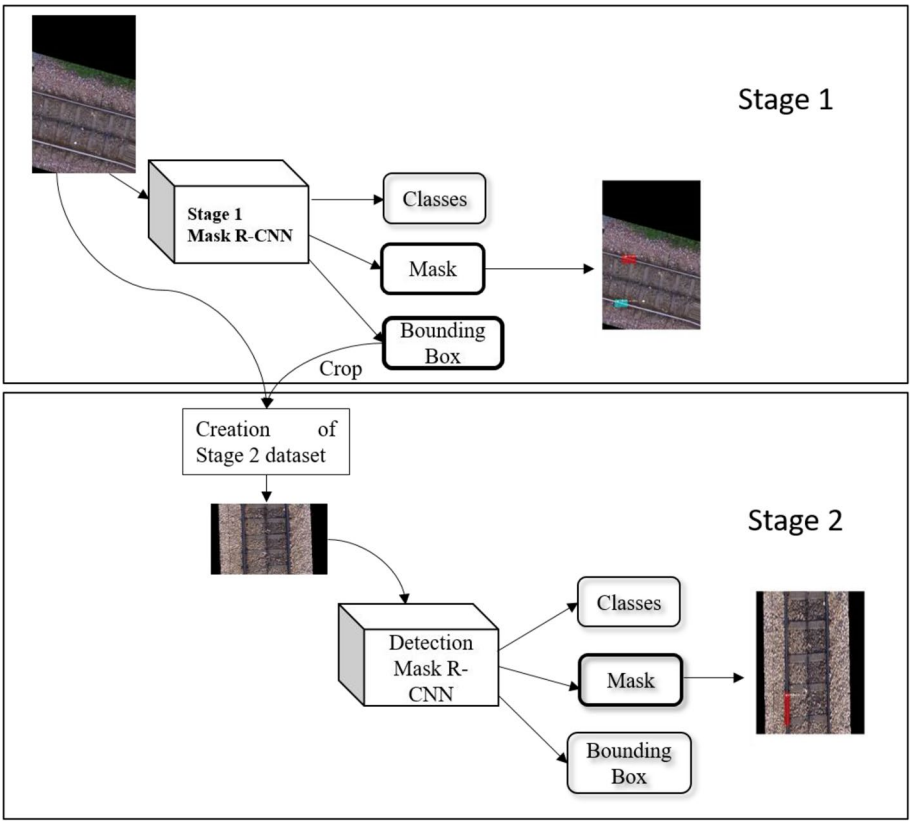

Fig. 10 Fishplate instance detection process in FishTwoMask R-CNN. The Stage 1 and Stage 2 in the architecture are depicted as marked Stage 1 and Stage 2

6.1.2 Experimental Scenario 2

This test image in Fig. 11d has been captured dated 21–06-2017 at a height of 25.2 m with the central latitude/ longitude coordinates of 29°46′3.8871″N/ 78°0′35.2280″E respectively. The illumination scenario can be inferred as sunny. It is observed that the image contains no fishplate. Although, Stage 1 showcases false detection on rail line area masked as fishplate location in red as shown in Fig. 11e however, Stage 2 test image output is correct as no fishplate is detected as in Fig. 11f which is as per ground truth observation.

6.1.3 Experimental Scenario 3

The test image shown in Fig. 11g has been captured on 30–08-2017 at a height of 25 m with central latitude/longitude coordinates of 29°49′39.5516″N/ 77°55′30.0578″E respectively. The image has been captured on a bright sunny day. However, the track has been overshadowed by the train running on the rail lines in the railroad environment. The image has no fishplate and both Stage 1 as well as Stage 2 test images output correctly regarding absence of fishplate as observed in Fig. 11h, 11i respectively.

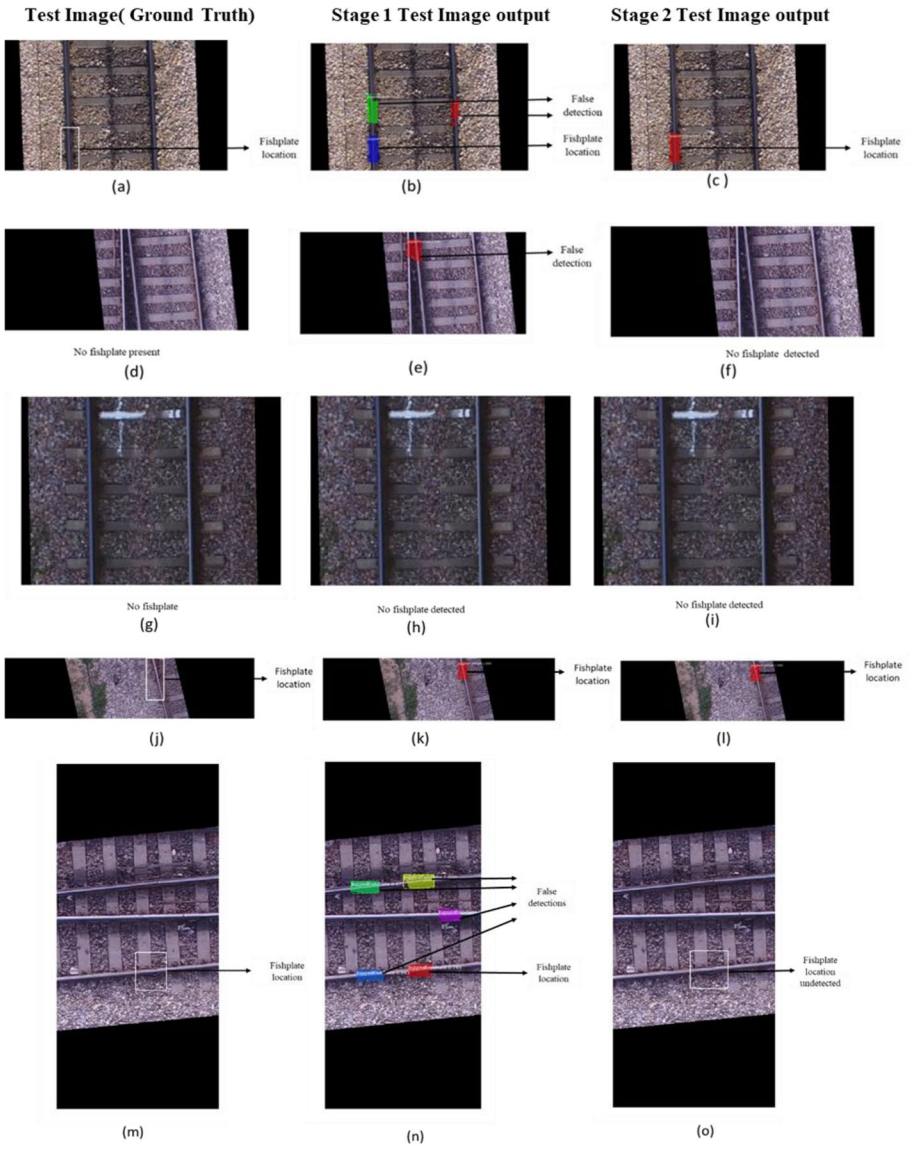

Fig. 11 Fishplate instances detections depicted in test images captured under different environmental scenarios as discussed in Sect. 6

6.1.4 Experimental Scenario 4

The test image in Fig. 11j has been captured as per Scenario 3 with the central latitude/longitude coordinates of 29°46′3.3522″N/78°0′35.0221″E respectively. It is observed that both Stage 1 and Stage 2 test image output showcase correct fishplate location in red as per ground truth observation as shown in Fig. 11k, and 11l respectively.

6.1.5 Experimental Scenario 5

The test image in Fig. 11m is captured on 21–06-2017 at a height of 25.2 m with the central latitude/ longitude coordinates of 29°46′3.8871″N/ 78°0′35.2280″E respectively. The image comprises of one fishplate. However it is observed that even though Stage 1 test image output showcases correct fishplate location masked as red still four incorrect fishplate locations masked as blue, green, yellow, pink can also be seen in Fig. 11n. The incorrect locations are rail line areas having no fishplate instances. The architecture at Stage 2 is unable to detect the fishplate in the test image as observed in Fig. 11o.

6.2 Metrics evaluation and discussion

The visual evaluation of the proposed method is performed as discussed in Sect. 6.1 and shown in Fig. 11 (c), (f),(i),(l),(o) which are Stage 2 (final) detection results. The fishplate test images undertaken for performing experimental results have been captured in different railroad environments at varied heights, locations and different dates/time. The purpose of statistical evaluation is in terms of computation of the metrics. The description of the metrics is as given in Eq. 6, Eq. 7, Eq. 8, and Eq. 9.

In this work we assume, True Positive (TP) as number of correct fishplate instance matches, True Negative (TN) as no fishplate instances correctly rejected, False Positive (FP) as proposed fishplate instances that are incorrect and False Negative (FN) as matches that are not correctly detected. The metrics are computed as mean values on 80 test images (40 test images each for fishplate and no fishplate instances (absence of fishplate instance in the test image)), 100 test images (50 test images each of fishplate and no fishplate instances), and 116 test images (58 test images each for fishplate and no fishplate instances). The values of TP, TN, FP, and FN are computed through Eq. 2, Eq. 3, Eq. 4 and Eq. 5. TP_{fish} denotes TP values for fishplate instances, TN_{nofish} denotes TN calculated on no fishplate instances while FP_{fish}, FP_{nofish} denotes FP values calculated on fishplate and no fishplate instances respectively. FN_{fish} denotes FN values computed for fishplate instances respectively. The analysis results of various evaluations for mean values for 80 test images are represented in Table 5. The final Stage 1 and Stage 2 results have been highlighted in the red boxes. The Table 8 and Table 9 in Appendix I highlight metrics values for 100 and 116 test images respectively.

Various evaluations have been performed with different sets of w_4,w_5 loss weights alongside different training "layers" in order to obtain best detection results for both stages. Few of these combinations have been listed in Table 5. For Stage 1 the notations of 50W8E1.0L indicate 50.h5 weight file from 8th evaluation with w_4, w_5 equal to 1.0., 1.0 respectively, and 59W8E1.0L indicate 59.h5 weight file from 8th evaluation with w_4, w_5 weight equal to 1.0.,1.0 respectively. For Stage 2 60W32E1.5L indicates 60.h5 weight file from 32nd evaluation with w_4, w_5 equal to 1.5., 1.0 respectively, 60W34E1.0L indicates 60.h5 weight file from 34th evaluation with w_4, w_5 equal to 1.0., 1.0 respectively, and 100W37E1.5L indicates 100.h5 weight file from 37th evaluation with w_4, w_5 of 1.5, 2.0 respectively. Upon critical analysis of the Stage 1 and Stage 2 metrics values, 59W8E1.0L and 100W37E1.5L have been chosen for Stage 1 and 2 respectively. As observed the metrics values of Stage 2 are higher than those in Stage 1 except that recall for Stage 1 is higher than recall for Stage 2.

$$TP = TP_{fish} \tag{2}$$

$$TN = TN_{nofish} \tag{3}$$

$$FP = FP_{fish} + FP_{nofish} \tag{4}$$

$$FN = FN_{fish} \tag{5}$$

6.2.1 Metrics evaluation

The metrics: Precision, Recall and F1 score are computed on the following previous algorithms for fishplate detection in DI: Normalized Correlation Coefficient-based Template Matching [25], Features-based template Matching [23] alongside Stage 1 and Stage 2 Mask R-CNN. In [25] the normalized correlation coefficient is evaluated on the image- pair while non-maximum suppression method is used for merging nearby fishplate detections in DI. A large number of false positives are observed in this method which is indicated, upon calculation, through the precision and F1-score values shown in Table 6. This is the case as these metrics are a function of FP, as depicted in Eq. 7 and Eq. 9. The work in [23] computes various feature descriptors and their respective Separability Index (SI)values, which is computationally intensive as it involves handcrafted feature extraction. As observed in Table 6 the precision, and F1-score values are higher than the work in [25] but lower than respective Stage 1 and Stage 2 metrics values. However, recall values for [23] are observed to be lower than for [25] as well as from Stage 1 and Stage 2 in proposed approach.

The comparative analysis is performed, through evaluation metrics, with the existing popular object detection algorithms: YOLOv5, Faster R-CNN, alongside Stage 1 & Stage 2 Mask R-CNN in proposed approach FishTwoMask R-CNN, as shown in Table 7. In this work YOLOv5 has been trained using Tesla T4 with batch size, input image size, and number of epochs as 16, 416, and 100 respectively. The training hyper-parameters for Faster R-CNN network along with rest of the hyper-parameters for YOLOv5 are presented in Table 4. It is observed that during test image evaluations each of the metrics values for Stage 1 are much lower than corresponding Stage 2 values except recall values. In Table 7 it is observed that Faster R-CNN has obtained metrics values of precision, recall and F1-score higher than respective YOLOv5 metrics values in this work. It can also be well observed that Mask R-CNN in Stage 1 of the proposed approach achieves recall and F1-score higher than Faster R-CNN indicating that most of the fishplate instances have been located. The Stage 2 achieves higher metrics values than those for Faster R-CNN. Also, it can well be inferred that Mask R-CNN, at any stage, is an extension of the functionality of Faster R-CNN and very-efficient. The results for the proposed method FishTwoMask R-CNN have been depicted in blue in Table 7. The Stage 2 of FishTwoMask R-CNN (proposed approach) has obtained higher metrics values as compared to Stage 1 Mask R-CNN except for recall hence, the adaptation in the loss weights w_4 andw_5, in Eq. 1, from 1 and 1 to 1.5 and 2 respectively

Table 6 Comparative analysis with previous methods through Metrics performance

Metrics Evaluation			
Models/Metrics	Precision	Recall	F1 Score
Normalized Correlation Coefficient and Template Matching [1]	0.011552	0.670886	0.022713
Features-based Template Matching [23]	0.133	0.641	0.220
Stage 1 Mask R-CNN	0.473	0.860	0.610
FishTwoMask R-CNN (Proposed Approach)	0.975	0.780	0.867

Table 7 Comparative analysis with deep learning models through Metrics performance

	Metrics Evaluation		
Models/Metrics	Precision	Recall	F1 score
Yolov5	0.400	0.666	0.499
Faster R-CNN	0.500	0.750	0.600
Stage 1 Mask R-CNN	0.473	0.860	0.610
FishTwoMask R-CNN (Proposed Approach)	0.975	0.780	0.867

is observed to have reflected the desired changes in the detection results. Also, the changes in the "layers" from ' heads' to '4+' along with reinforcement of cropped fishplate instances in the Stage 2 dataset can be held accountable for the improvement and precise detection results.

$$Accuracy = TP + TN/(TP + TN + FP + FN) \tag{6}$$

$$Precision = TP/(TP + FP) \tag{7}$$

$$Recall = TP/(TP + FN) \tag{8}$$

$$F1score = 2 * Precision * Recall/(Precision + Recall) \tag{9}$$

6.3 Complexity analysis

Fishplate instances are critical railroad track components essential to maintain railroad track safety and avoid mishaps, which are indicative of loss of life and property, as discussed in Sect. 1. Therefore, it is important to locate fishplate instances accurately even with comparable amount of computational overhead leading to increase in model complexity. As fishplates are tiny objects in the railroad track drone images, Stage 1 Mask R-CNN does not yield good results. Also, artefacts such as illumination, similar color rail lines deteriorate these metrics values for Stage 1. Alongside these challenges, problems posed for DI as discussed in Sect. 1, are also taken into consideration. Consequently, need for Stage 2 Mask R-CNN arises and the metrics are also indicative of good results marked in blue in Table 6 and Table 7. As the fishplate component is very small compared to drone image size as well as a critical component for safety therefore, this makes correct detections, with an overhead on time, as the foremost goal. The time complexity for the method during evaluation is calculated through computation on each of the test images which is ≈ 1310.312636 s (Stage 1 with 11 s/ image) and achieved ≈ 1286.798505 s (Stage 2 with 10 s /image).

7 Conclusion

In this paper, we have proposed a novel adaptive two-stage Mask R-CNN framework termed as FishTwoMask R-CNN for fishplate instances detection and segmentation in drone images for further inspection of railroad track health. During the process of achieving our goal the following observations have been made:

I This framework is divided into two stages. The Stage 1 involves creation of fishplate instances dataset from segmented railroad track DI and training using these original images. However, it is observed this gives a precision and an F1-score of only 47% and 61% respectively. In order to remove false detections and boost detection accuracy the cropped fishplate instances along with the original Stage 1 dataset are used for training purpose in Stage 2. It is observed that the framework then achieved a detection precision rate of 97% with an F1-score of 86%.

II The Stage 2 network in the proposed architecture is henceforth trained using different weights for loss function components. This helps in improving performance of the framework on different test datasets acquired from different distributions while indicating potential field applications in the future.

III The change in the "layers" hyper-parameter value from Stage 1 to Stage 2 also indicates potential improvements in the precision, recall and F1-score values in the proposed architecture.

IV The work has been tested on different railroad track environmental scenarios and our algorithm has performed well on all of them thus, indicating the robustness of the proposed method.

V The fishplate instances detection in DI has been evaluated using other previously developed methods such as Features-based Template matching [23] as well as existing object recognition algorithms such as Faster R-CNN and YOLOv5. The proposed algorithm is observed to achieve better detections as well as evaluation metrics values than the respective values in most of the aforementioned algorithms.

Appendix I

Table 8 Evaluation of metrics on 100 test images (50 test images each for fishplate and no fishplate instances)

Metrics	1st Stage		2nd Stage			
	50W8E1.0L	59W8E1.0L	60W32E1.5L	60W34E1.0L	45W35E1.0L	100W37E1.5L
Accuracy	0.453	0.490	0.423	0.434	0.820	0.829
Precision	0.407	0.431	0.375	0.400	0.955	0.956
Recall	0.767	0.810	0.750	0.733	0.700	0.717
F1 score	0.532	0.563	0.500	0.518	0.808	0.819

Table 9 Evaluation of metrics on 116 test images (58 test images each for fishplate and no fishplate instances)

Metrics	1st Stage		2nd Stage			
	50W8E1.0L	59W8E1.0L	60W32E1.5L	60W34E1.0L	45W35E1.0L	100W37E1.5L
Accuracy	0.459	0.473	0.420	0.435	0.843	0.843
Precision	0.409	0.414	0.371	0.394	0.962	0.944
Recall	0.794	0.833	0.779	0.765	0.735	0.750
F1 score	0.540	0.553	0.502	0.520	0.833	0.836

Acknowledgements The authors would like to thank RailTel, India for supporting this work. The authors would also like to extend their thanks to British Council and IIT Roorkee for granting Newton Bhabha Fund under which a part of this research work has been carried out at The University of Sheffield, Sheffield, United Kingdom.

Data availability The datasets generated during and/or analyzed during the current study are not publicly available due to privacy reason but are available from the corresponding author on reasonable request.

Declarations

Conflict of interest The authors would like to state that there is no conflict of interest amongst any of the authors.

References

1. Abdulla W (2017) Mask r-cnn for object detection and instance segmentation on keras and tensorflow
2. Bharath B, Kanmani M (2017) Swarm intelligence based image fusion for thermal and visible images, in 2017 International Conference on Computation of Power, Energy Information and Commuincation (ICCPEIC), 043–048
3. Bhat S, Karegowda D, Noushad I (2021) Smart railway track monitoring system
4. Buggy SJ et al (2016) Railway track component condition monitoring using optical fibre Bragg grating sensors. Meas Sci Technol 27(5):055201
5. Chen Z et al (2022) Foreign object detection for railway ballastless trackbeds: A Semisupervised Learning Method. Measurement, 110757
6. Du C, Dutta S, Kurup P, Yu T, Wang X (2020) A review of railway infrastructure monitoring using fiber optic sensors. Sensors and Actuators A: Physical 303:111728. https://doi.org/10.1016/j.sna.2019.111728
7. Gao M, Wu H, Shen Y, Wang X, Zeng Y (2019) A peak detection algorithm adopting magnetic sensor signal for rail spike location in tamping machine. Adv Mech Eng 11(11):1687814019891570
8. Gavai G, Eldardiry H, Wu W, Xu B, Komatsu Y, Makino S (2019) Hybrid image-based defect detection for railroad maintenance. Electronic Imaging 2019(9):360–361
9. Güçlü E, Aydın İ, Akın E (2022) Measurement of railway sleepers spacing using mask R-CNN, in 2022 International Conference on Decision Aid Sciences and Applications (DASA), 1416–1420
10. Guo F, Qian Y, Rizos D, Suo Z, Chen X (2021) Automatic rail surface defects inspection based on Mask R-CNN. Transp Res Rec 2675(11):655–668
11. Guo F, Qian Y, Wu Y, Leng Z, Yu H (2021) Automatic railroad track components inspection using real-time instance segmentation. Computer-Aided Civil and Infrastructure Engineering 36(3):362–377
12. He K, Gkioxari G, Dollár P, Girshick R (2017) Mask r-cnn, in Proceedings of the IEEE international conference on computer vision, 2961–2969
13. Hodge VJ, O'Keefe S, Weeks M, Moulds A (2015) Wireless sensor networks for condition monitoring in the railway industry: A Survey. IEEE Transactions on Intelligent Transportation Systems 16(3):1088–1106. https://doi.org/10.1109/TITS.2014.2366512
14. Kafetzis D, Fourfouris I, Argyropoulos S, Koutsopoulos I (2020) UAV-assisted aerial survey of railways using deep learning, in 2020 International Conference on Unmanned Aircraft Systems (ICUAS), 1491–1500
15. Kanmani M, Narasimhan V (2019) An optimal weighted averaging fusion strategy for remotely sensed images. Multidimension Syst Signal Process 30(4):1911–1935
16. Kanmani M, Narasimhan V (2020) Optimal fusion aided face recognition from visible and thermal face images. Multimedia Tools and Applications 79(25):17859–17883
17. Liu J, Wang Z, Wu Y, Qin Y, Cao X, Huang Y (2020) An Improved Faster R-CNN for UAV-Based Catenary Support Device Inspection. Int J Software Eng Knowl Eng 30(07):941–959
18. Madheswari K, Venkateswaran N (2017) Swarm intelligence based optimisation in thermal image fusion using dual tree discrete wavelet transform. Quantitative Infrared Thermography Journal 14(1):24–43
19. Madheswari K, Venkateswaran N, Sowmiya V (2016) Visible and thermal image fusion using curvelet transform and brain storm optimization, in 2016 IEEE Region 10 Conference (TENCON), 2826–2829

20. Nayan MMR, Al Sufi S, Abedin AK, Ahamed R, Hossain MF (2020) An IoT based real-time railway fishplate monitoring system for early warning, in 2020 11th International Conference on Electrical and Computer Engineering (ICECE), 310–313
21. Rampriya RS, Suganya R, Nathan S, Perumal PS (2022) A comparative assessment of deep neural network models for detecting obstacles in the real time aerial railway track images. Applied Artificial Intelligence, 1–33
22. Ravichandran A, Raja A, Kanmani M (2017) Entropy optimized image fusion: Using particle swarm technology and discrete wavelet transform, in 2017 international conference on computation of power, energy information and commuincation (ICCPEIC), 068–074
23. Saini A, Agarwal A, Singh D (2020) Feature-based template matching for joggled fishplate detection in railroad track with drone images, in IGARSS 2020–2020 IEEE International Geoscience and Remote Sensing Symposium, 2237–2240
24. Saini A, Kishore KG, Sriram KSS, Singh D, Singh KP (2022) Machine learning approach for detection of track assets for railroad health monitoring with drone images, in IGARSS 2022–2022 IEEE International Geoscience and Remote Sensing Symposium, 4891–4894
25. Saini A, Singh D (2018) Development of computer vision based robust approach for joggled fish plate detection in drone images, in 2018 9th International Symposium on Signal, Image, Video and Communications (ISIVC), 33–38
26. Saini A, Singh D (2021) DroneRTEF: development of a novel adaptive framework for railroad track extraction in drone images. Pattern Anal Appl 24(4):1549–1568
27. Singh P, Garg RD (2014) Classification of high resolution satellite images using spatial constraints-based fuzzy clustering. J Appl Remote Sens 8(1):083526
28. Singh PP, Garg RD (2015) Fixed point ICA based approach for maximizing the non-Gaussianity in remote sensing image classification. Journal of the Indian Society of Remote Sensing 43(4):851–858
29. Sumit SS, Watada J, Roy A, Rambli DRA (2020) In object detection deep learning methods, YOLO shows supremum to Mask R-CNN. J Phys: Conf Ser 1529(4):042086
30. Szeliski R (2010) Computer vision: algorithms and applications. Springer Science & Business Media
31. Wu Y, Meng F, Qin Y, Qian Y, Xu F, Jia L (2023) UAV imagery based potential safety hazard evaluation for high-speed railroad using Real-time instance segmentation. Adv Eng Inform 55:101819
32. Yilmazer M, Karakose M (2022) "Mask R-CNN architecture based railway fastener fault detection approach". International Conference on Decision Aid Sciences and Applications (DASA) 2022:1363–1366
33. Zheng D et al (2021) A defect detection method for rail surface and fasteners based on deep convolutional neural network. Computational Intelligence and Neuroscience