NRCan

Kishan Patel, Michal Aibin, Rohan Sharma, Sanyami Shah,
Khoury College of Computer Sciences, Northeastern University, Vancouver, Canada
email: m.aibin@northeastern.edu

***Abstract*—Object Detection is an elemental part of Computer vision with a wide range of real-world applications. It involves the detection of various objects in digital images or video. One such object detection algorithm is Yolo (You only look once) which has gained much popularity in the computer vision community due to its high performance and low inference time. In this project, we propose a proof of concept of how the Yolo algorithm can be leveraged to improve the maintenance of railway tracks operated by Via Rail Canada. Via rail operates about 500 trains running on 12,500 km of track. These tracks pass through long stretches of sparsely populated lands which makes maintaining these tracks a tough job due to the sheer amount of resources required to identify the points of interest such as Vegetation, Missing or broken Tie, etc. This project uses the Yolo algorithm to identify these points of interest with the help of drone flights. This project also explores the performance of different versions of Yolo (Yolov4 and Yolov5) to achieve the highest accuracy possible.**

***Keywords*—**

I. Introduction

Railways are a crucial part of the modern world as they not only help provide affordable public transport but also facilitate the supply chain of goods that are essential for the current lifestyle. Railway tracks span thousands of kilometers and maintaining these tracks is not an easy job for multiple reasons. The first and quite obvious reason is that these tracks pass through some of the most remote areas which are beneficial in terms of connectivity, but on the other hand it also makes surveillance and monitoring of these tracks quite difficult as some of the areas might not be easily accessible or require additional resources. Second, it also requires railway companies to employ a large workforce to ensure smooth operations and the safety of the general public.

One of the most crucial parts in the maintenance of railway tracks is to identify which sections require repairs. Via Rail operates more than 500 trains per week across several Canadian provinces and 12,500 kilometers (7,800 mi) of the track. The current approach requires manual monitoring of tracks regularly, which is resource-intensive. one of the important points of interest involves identifying vegetation growing on the side of tracks which could prove to be fatal if not attended on a timely basis. Identifying missing and broken wooden ties is an important task to increase the lifetime and operability of the tracks. Rainy weather could contribute to the sudden accumulation of water near tracks that is extremely difficult to identify quickly. Advancement in certain technologies could revolutionize the identification of these points of interest.

Drone technology has essentially opened new opportunities for railway operators, offering safety and dependability, as well as reliable support. The increased safety of workers is now commendable with Drone Technology on the railway sector's side. With the arrival of drones, railway activities that previously required personnel to engage in dangerous scenarios for inspections are no longer a concern. The integration of premium drones, advanced sensors, high-resolution cameras, Artificial Intelligence, and Machine Learning has enabled this transportation industry to capture real-time data and do smart data analysis.

Evolution of computer vision capabilities has made it possible to identify particular objects from various media sources such as images and real-time video streaming. Mao et al. offered a way to perform the direct extraction of points of interest on the rail surface by improving the value of reducing the maximum variance between classes, although the accuracy of the recommended approach must be improved [1]. Tang et al. used the gray-level projection to achieve quick rail surface segmentation; however, the approach is unadaptable [2]. Yuan et al. employ the enhanced Ostu algorithm to segment the rail picture and calculate the rail surface area, although the threshold selection of this method is not uniform [3]. He et al. extracted rail surface flaws using the mean background subtraction approach, although the algorithm robustness was lacking [4].

The following are the main contributions of this paper.

- **Approach:** We study and apply the YOLOv5 object detection algorithm to detect broken ties, missing ties, vegetation and water pooling around railway tracks.
- **Experimentation:** We collect aerial image footage for training and testing purposes using DJI Inspire in Ottawa region.
- **Evaluation:** We evaluate the algorithm's accuracy by measuring how correctly it determines a particular point of interest. We take in consideration precision, recall and mean average precision to determine the overall accuracy. Overall we find that, precision is 66% and mean average precision@0.5 is 60.3%.

II. Related Work

Tiange Wang, Fangfang Yang, and Kwok-Leung Tsui in [5] use the one-stage detection model YOLO to detect Rail, Clip, and Bolt in railway tracks. Bolts are used to fasten rail ends together at joints and elastic clips are used with rail sleeper to fasten rails. In their approach, they use the k-means clustering method to discover the best aspect ratios of prior boxes with the distance metric changed to 1-IoU. They use 6 anchors for YOLOv2 and 9 anchors for YOLOv3. They experiment and compare three versions of YOLO, versions 1, 2, and 3. YOLO version 1 resulted in mAP (mean average precision) of 92%, precision of 89%, and recall of 93%. YOLO version 2 resulted in mAP (mean average precision) of 93%, precision of 83%, and recall of 97%. YOLO version 3 resulted in mAP (mean average precision) of 89%, precision of 94%, and recall of 84%.

978-1-7281-9656-5/20/$31.00 ©2020 IEEE

In [6] Wenju Li, Zihao Shen and Peigang Li employ YOLO, Faster R-CNN, and Improved YOLO to detect cracks in the track plate of railway tracks. They refer to Improved YOLO to RPN-FCN network, in which the fully connected layer in the original YOLO structure is replaced by a 1*1 convolutional layer and an anchor box is introduced. In their approach, YOLO resulted in the precision of 79.61% and recall of 76.13%, Faster R-CNN resulted in the precision of 83.24% and recall of 50.54% and Improved YOLO resulted in the precision of 83.54% and recall of 79.93%.

Face detection has been made insufficient progress in the VR games field, due to the lack of database of VR games. In available technology of face detection, specifically for the uncovered facial occlusion, mainly relies on the eyes features to detect faces. However, it does not work to detect a face when the face has facial occlusion. Therefore, it needs to train a model with a neural network to detect faces. The first paper [7] detected the face from the VR game where they collected the face images of the VR games, annotated the position of the face and used the YOLOv5 neural network as a single target face detection.

With the widespread use of X-ray screening devices, intelligent detection of contrabands in X-ray screening pictures has become increasingly important. Due to the random distribution of the items, which might cause the target objects and other objects to overlap, detecting contrabands in X-ray screening pictures is a difficult challenge in the field of security detection. Traditional image processing and recognition algorithms struggle to partition X-ray security pictures into separate candidate zones containing different items. In recent years, the YOLO (You Only Look Once, a Realtime Object Detection System) Model was introduced, which gives a basic framework for directly predicting bounding boxes and class probabilities from entire photos. A YOLO-based approach is utilised to detect contrabands in X-ray screening pictures in this research [8]. The results of the experiments reveal that the precision and recall rate of contraband identification against a simple backdrop are both greater than 98 percent and 94 percent, respectively. Although accuracy remains around 95 percent in a complicated context, the recall rate of particular contrabands has declined to below 70 percent.

It is impractical to assume that images are always captured in perfect light conditions. Low light images usually degrades with scene height and suffers from severe object information loss. Author in article [9] proposed physical based image enhancement enhances the image contrast by exploiting the relation among the atmosphere light and transmission map. Another subnet detects salient object from enhanced images. Author Used five SOD datasets such as DUT-OMRON, PASCAL-S, NTI-V1 are used to evaluate model for precision recall curves, F-measures and Mean absolute error. Promising results were observed in NTI-V1 and public dataset with high volume of false positive due to noise interference.

Large number of images with ground truth information in form of annotation is bottleneck for any object detection application. Author in article [10] studies existing crowdsource techniques such as Amazon Mechanical Turk and proposed turn based annotation system consisting of three simple task (single object detection, a quality verification task, coverage verification task) is proposed. Pascal VOC 2007 dataset was used for evaluation where 3 rounds of labeling are performed to deal with low annotation difficulty. Proposed technique shows improvement in accuracy by up to 8-10%. Proposed system can assist object detector to obtain higher accuracy in a cost effective manner.

III. Algorithm Description

A. YOLO

YOLO (You Only Look Once) reframes object detection as a single regression problem rather than a classification problem. A single convolutional network simultaneously predicts multiple bounding boxes and class probabilities for those boxes shown in figure 1 [11]. YOLO reasons globally about the

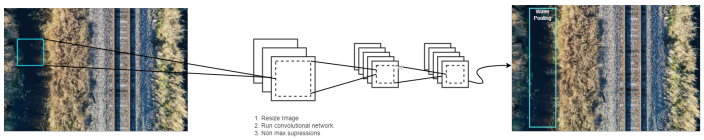

Fig. 1. YOLO Object Detection

image when making predictions. It sees the entire image during training and testing time, so it implicitly encodes contextual information about classes and their appearance[11].

1) Detection: The input image is divided into an $S * S$ grid. If the center of an object falls into a grid cell, that grid cell is responsible for detecting the Object. Each grid cell predicts B bounding boxes and confidence scores for those boxes. Confidence is defined as $P_r(Object) * IOU(truth, pred)$. If no object exists in that cell, the confidence score is zero. Otherwise, the confidence score equals the intersection over union (IOU) between the predicted box and the ground truth[11].

Each bounding box consists of 5 predictions: *x, y, w,h*, and *confidence*. The *(x,y)* coordinates represent the center of the box relative to the bounds of the grid cell. The *(w,h)* coordinates represent the width and height of the bounding box and are predicted relative to the whole image. Finally, the confidence prediction represents the intersection over union (IOU) between the predicted box and any ground truth box[11].

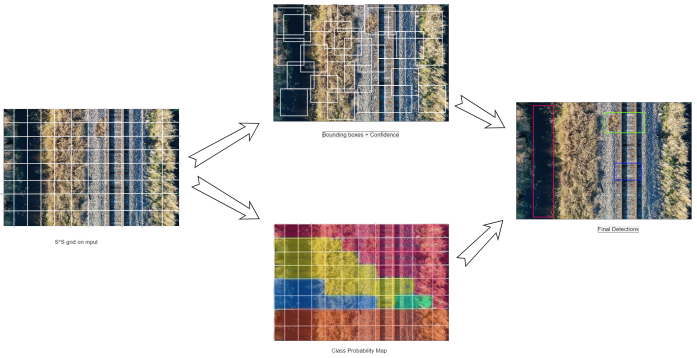

Fig. 2. Object Detection using YOLO

Each grid cell also predicts C conditional class probabilities, $P_r(Class_i \mid Object)$. These probabilities are conditioned on the grid cell containing an object. Only one set of class probabilities

are predicted per grid cell regardless of the number of boxes. At test, the conditional class probabilities and the individual box confidences are multiplied as[11],

$$P_r\left(Class_i|Object\right) * P_r\left(Object\right) * IOU_{pred}^{truth} = P_r\left(Class_i\right) * IOU_{pred}^{truth} \quad (1)$$

2) Architecture: The model is implemented as a convolutional neural network. The initial convolutional layers of the network extract feature from the image, while the fully connected layers predict the output probabilities and coordinates. The detection network architecture has 24 convolution layers followed by two fully connected convolution layers. Alternating 1 * 1 convolution layers reduce the features space from preceding layers. The network is shown in figure 3.

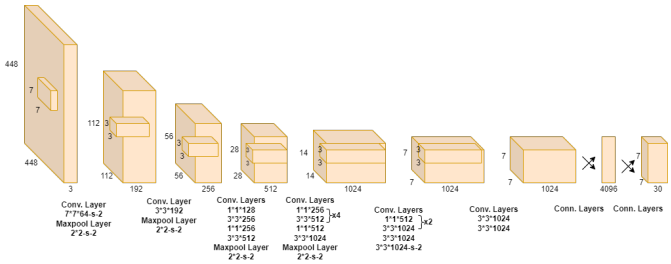

Fig. 3. Architecure of YOLO

3) Activation and Loss Function: The final layer predicts both class probabilities and bounding box coordinates. The bounding box coordinates are normalized to 0 and 1. A linear activation function for the final layer and all other layers use the following leaky rectified linear activation:

$$\phi(x) = \begin{cases} x, & if x > 0 \\ 0.1x, & otherwise \end{cases} \quad (2)$$

The sum-squared error in the output of the loss function is optimized using the equation below.

$$\begin{aligned} &\lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^{B} 1_{ij}^{obj} \left[\left(x_i - \widehat{x}_i\right)^2 + \left(y_i - \widehat{y}_i\right)^2 \right] + \\ &\lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^{B} 1_{ij}^{obj} \left[\left(\sqrt{w_i} - \sqrt{\widehat{w}_i}\right)^2 + \left(\sqrt{h_i} - \sqrt{\widehat{h}_i}\right)^2 \right] \\ &+ \sum_{i=0}^{S^2} \sum_{j=0}^{B} 1_{ij}^{obj} \left(C_i - \widehat{C}_i\right)^2 + \\ &\lambda_{noobj} \sum_{i=0}^{S^2} \sum_{j=0}^{B} 1_{ij}^{noobj} \left(C_i - \widehat{C}_i\right)^2 + \\ &\sum_{i=0}^{S^2} 1_i^{obj} \sum_{c \varepsilon classes} \left(p_i(c) - \hat{p}_i(c)\right)^2 \end{aligned} \quad (3)$$

B. YOLOv5

The architecture of YOLOv5 consists of three parts. Backbone: CSPDarknet, Neck: PANet, and Head: Yolo Layer.

The data are first input to CSPDarknet for feature extraction, and then fed to PANet for feature fusion. Finally, Yolo Layer outputs detection results (class, score, location, size) [12].

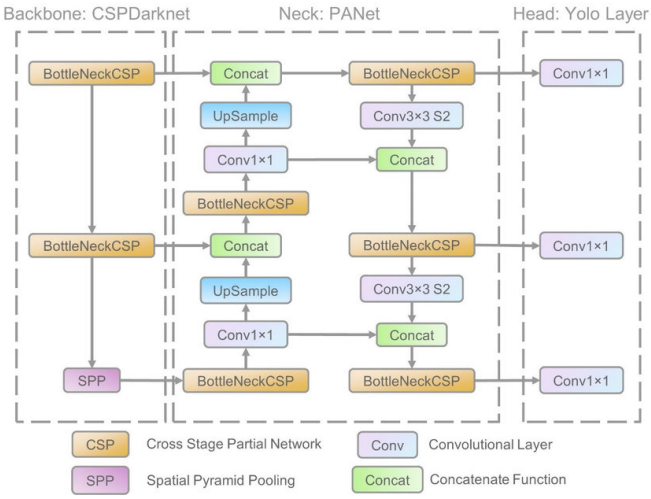

Fig. 4. Architecure of YOLOv5

Joseph Redmon (inventor of YOLO) introduced the anchor box structure in YOLOv2 and a procedure for selecting anchor boxes of size and shape that closely resemble the ground truth bounding boxes in the training set. By using the k-mean clustering algorithm with different k values, the authors picked the 5 best-fit anchor boxes for the COCO dataset (containing 80 classes) and use them as the default. That reduces training time and increases the accuracy of the network[13].

However, when applying these 5 anchor boxes to a unique dataset (containing a class not belonged to 80 classes in the COCO dataset), these anchor boxes cannot quickly adapt to the ground truth bounding boxes of this unique dataset. For example, a giraffe dataset prefers the anchor boxes with the shape thin and higher than a square box. To address this problem, computer vision engineers usually run the k-mean clustering algorithm on the unique dataset to get the best-fit anchor boxes for the data first. Then, these parameters will be configurated manually in the YOLO architecture[13].

Glenn Jocher (inventor of YOLOv5) proposed integrating the anchor box selection process into YOLOv5. As a result, the network has not to consider any of the datasets to be used as input, it will automatically "learning" the best anchor boxes for that dataset and use them during training[13].

IV. Methodology

Following Figure 5 shows the model pipeline with each of the steps involved from data gathering to running inferences on the images. Data was collected using UAV by performing flights over the Ottawa region during fall 2021. High resolution images were captured from various distances [25ft, 50ft, 75ft]. In the next step after performing data pre-processing, RoboFlow tool was used to perform annotation for the four points of interests.

Roboflow also enabled the augmentation and versioning of dataset, After creating dataset package in Roboflow, google colab notebook was used to run Yolov5 algorithm for training the model. Once the model was trained trained weights were extracted and used for performing inference on the images.

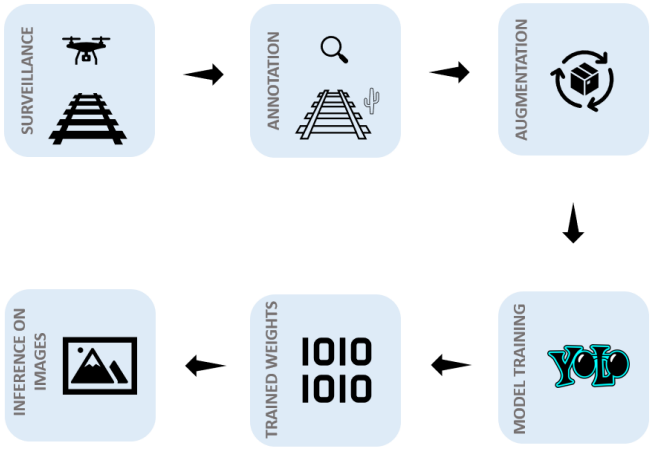

Fig. 5. Model Pipeline

A. Dataset Pre-processing

Dataset pre-processing is performed to ensure that the data is in the suitable format for the algorithm's training. After an initial assessment, the following pre-processing steps were performed on the dataset to enable efficient annotation, which is critical for machine learning model performance.

1) Image Orientation Correction: Annotation is performed with a square or rectangular block signifying the point of interest. To make the annotation process more manageable and efficient, the orientation of images was corrected such that tracks are in a vertical or horizontal direction.

2) Artificial Vegetation: As the rail tracks are well maintained, there were very few images where vegetation was visible. To achieve higher accuracy, we needed to train the model using a well-balanced set of images with points of interest; thus, artificial vegetation was added on and besides the tracks.

3) Artificial Missing Ties: Similarly, there were no images with missing ties; thus, ties were removed from some of the images to train the model for the missing tie scenarios.

B. Annotation

After pre-processing the data, the next step was to start annotating the images for training the algorithm. 4 points of interest that were selected for the training purpose were as follows:

- **Vegetation**: Each image was reviewed closely to check any vegetation near the track. If it was present at a close distance to the track, it was annotated so that algorithm gets trained accordingly.
- **Missing Tie**: The track portion was annotated where there was a missing tie.
- **Broken Tie**: If there was a crack on the tie or seemed broken, it was annotated as a broken tie.
- **Water Pooling**: Any water body captured in the images was annotated as water pooling to identify the sites affected by water pooling near tracks.

C. Augmentation

One of the most fascinating aspects of computer vision is the ability to enhance your effective sample size by combining current images with random alterations. Assume you have a single snapshot of a coffee mug. After that, duplicate the snapshot and rotate it 10 degrees clockwise. You haven't accomplished much, in your opinion. However, you've more than doubled the quantity of photographs you're preparing to provide your model! Your computer vision model now has a completely different viewpoint on how that coffee mug seems.

With an existing image dataset, data augmentation can increase model performance when creating computer vision models. Image augmentation expands a dataset's size and variability, enhancing model generalizability.

After the images annotation process, different augmentation was tried to compare their effect on the accuracy of the model.

- Exposure: Adjust the gamma exposure of an image to be brighter or darker.
- Saturation: Saturation augmentation is similar to hue except that it adjusts how vibrant the image is. A fully desaturated image is grayscale, partially desaturated has muted colors, and a positive saturation shifts colors more towards the primary colors.
- Mosaic: It works by taking four source images and combining them together into one.

V. Results

Overall we achieved 66% precision as shown in figure 6 and 60.3% mAP (mean average precision) for the detection of all the classes as in figure 7.

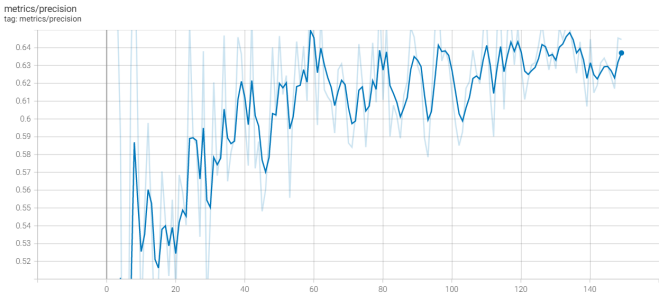

Fig. 6. Precision

Individually, we achieved 49% precision of detection for Broken Ties with 53.6% mAP.

For Missing Ties, we we achieved 82% precision of detection and 78% mAP.

Similarly for for Vegetation, we we achieved 75.6% precision of detection and 68.6% mAP.

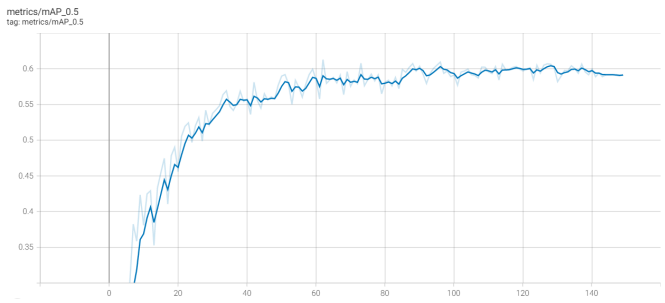

Fig. 7. mAP

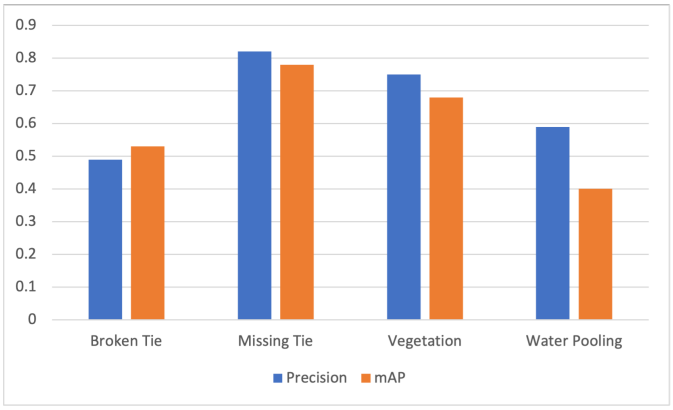

Fig. 8. graph

And for Water Pooling, we we achieved 59.7% precision of detection and 40.8% mAP.

A. *Example Results of Detection*

1) Detection of Vegetation: One of the tasks of our model was to detect vegetation near railway tracks.

The image in figure 9 and 10 shows vegetation detected on top of tracks with confidence score of 0.85 and 0.84.

Fig. 9. Vegetation near railway tracks

Fig. 10. Vegetation near railway tracks

2) Detection of Missing Tie: One of the tasks of our model was to detect missing ties on railway tracks.

The image in figure 11 and 12 shows vegetation detected on top of tracks with confidence score of 0.90 and 0.87.

Fig. 11. Missing Tie on Railway Track

3) Detection of Broken Tie: One of the tasks of our model was to detect broken ties on railway tracks.

The image in figure 13 and 14 shows vegetation detected on top of tracks with multiple confidence scores.

4) Detection of Water Pooling: One of the tasks of our model was to detect water pooling near railway tracks.

The image in figure 15 and 16 shows vegetation detected on top of tracks with confidence score of 0.71 and 0.79.

VI. Discussion

During the model training, many different experiments were performed to optimize the performance of each point of interest by analysing the initial results.

Fig. 12. Missing Tie on Railway Track

Fig. 13. Broken Tie on Railway Track

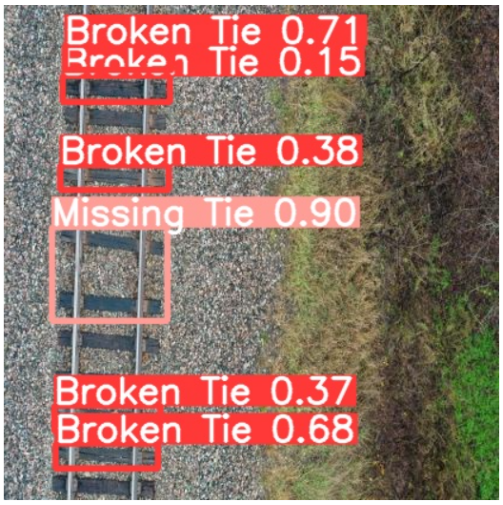

Fig. 14. Broken Tie on Railway Track

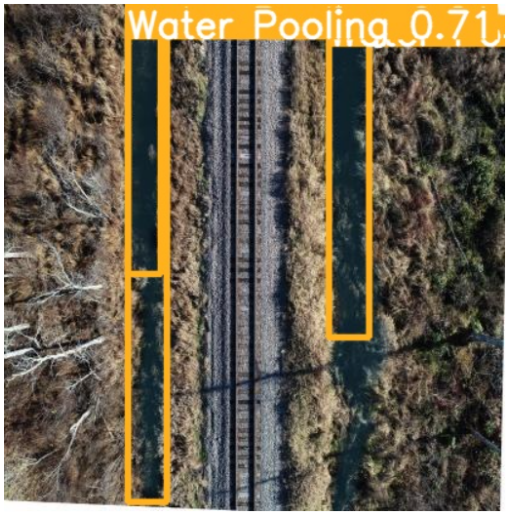

Fig. 15. Water Pooling near Railway Track

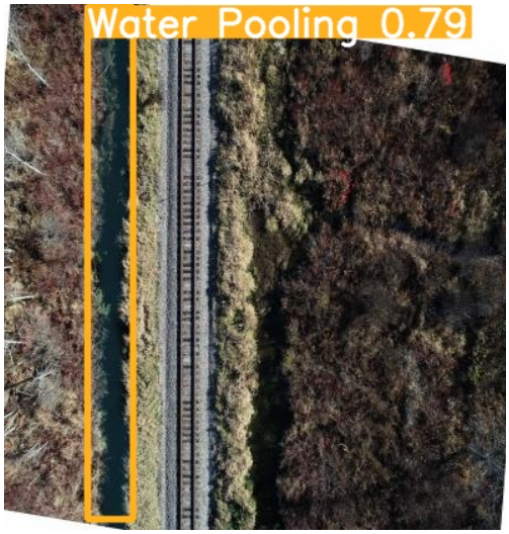

Fig. 16. Water Pooling near Railway Track

A. *Broken Ties*

During the initial iteration of the model training, the image data set consisted of images taken from greater height [50 - 70 meters] which contributed to lower performance as identifying the cracks in ties was difficult due to lower resolution. Thus, it was recommended that drone images need to be taken from a lower height [25 meters] to have optimum results. Pictures taken from greater heights were removed from the data set. As a result of these changes, 7% improvement in performance was witnessed. Overall, the performance of broken ties remained average due to number of different factors affecting algorithms' ability to detect broken tie. These factors involve high variation in cracks and low samples for each type of crack, material of tie [wooden and concrete], light conditions. All of these factors affect the appearance of the tie. Further research and experimentation with a large data set focused on the broken tie is required to optimize this category further. Future work could involve introducing a different level of severity identified to assign maintenance priority.

B. Missing Tie and Vegetation

As the drone images were taken from active rail tracks, there were very few instances of missing ties and vegetation. Thus, a photo editing tool was used to add artificial vegetation and missing ties to train the algorithm. Vegetation scenarios were created using the vegetation samples found on the original image to create more realistic scenarios. The impressive performance was witnessed initially, which was also maintained when tested on images with original vegetation and missing tie scenarios. Further research in the vegetation category could involve mapping the size of vegetation or tracking the level/amount of ballast to implement effective deterrence against vegetation growth. Similarly, tracking the distance between two instances of the missing tie could also help prioritize the maintenance.

C. Water pooling

Achieving high precision in the case of water pooling is extremely difficult, but having low precision does not necessarily mean that model performance is not good. As can be seen in the figure 17, annotation indicating ground truth does not fully overlap with the predicted boundary box resulting in the low precision due to low IOU [intersection over union] value. Low IOU value contributes to detection classified as false positive, which reduces the precision value. But in the actual scenario, the goal is to identify water, not the exact area of water pooling. Thus, having higher detection is more important than precision.

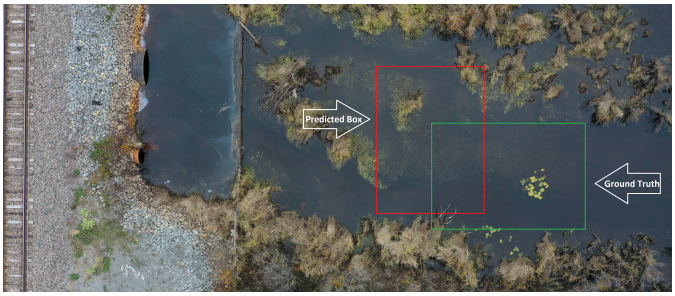

Fig. 17. Prediction Vs Ground Truth

Very poor precision of only 20% was witnessed initially but it was observed that detection of water was impressive. Annotation changes were made to improve precision by annotating a larger portion of the water body rather than multiple small annotations. This resulted in huge precision improvement by almost 40% due to less number of predictions classified as false positive.

VII. Conclusion

This paper tried to show the approach of identifying the various points of interest around the railway track using the aerial image footage and feeding them to the Yolo object detection algorithm. The algorithm's accuracy was evaluated by measuring how correctly it determines the points of interest.

It was observed that after feeding the algorithm with preprocessed image and adding the vegetation and missing ties, we could conclude that the accuracy of various points of interest was directly proportional to the distance of the images. Images with broken ties were hard to identify from a greater distance. Also, the algorithm was easily able to identify the vegetation and missing ties in the images from a closer distance. In the second iteration, it was noticed that as the number of images increased, the accuracy used to spike as the algorithm was able to learn about more images and was able to train itself more precisely. So more the images, better is the accuracy.

References

[1] Z.-c. Mao and F. Wu, "Visual detection algorithm for rail surface defects based on image sensor," *Transducer Microsystem Technol*, vol. 34, pp. 141–144, 2015.

[2] X. Tang and Y. Wang, "Visual inspection and classification algorithm of rail surface defect," *Computer Engineering*, vol. 39, no. 3, pp. 25–30, 2013.

[3] X. C. Yuan, L. S. Wu, and H. Chen, "Rail image segmentation based on otsu threshold method," *Optics & Precision Engineering*, vol. 24, no. 7, pp. 1772–1781, 2016.

[4] Z. He, Y. Wang, J. Liu, and F. Yin, "Background differencing-based high-speed rail surface defect image segmentation," *Chinese Journal of Scientific Instrument*, vol. 37, no. 3, pp. 640–649, 2016.

[5] 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, 2020. [Online]. Available: https://www.mdpi.com/1424-8220/20/15/4325

[6] W. Li, Z. Shen, and P. Li, "Crack detection of track plate based on yolo," in *2019 12th International Symposium on Computational Intelligence and Design (ISCID)*, vol. 2, 2019, pp. 15–18.

[7] T. Xie, Z. Chen, M. Cao, P. Hu, Y. Zeng, and Z. Pan, "Face detection in vr games," in *2020 the 3rd International Conference on Control and Computer Vision*, ser. ICCCV'20. New York, NY, USA: Association for Computing Machinery, 2020, p. 7–10. [Online]. Available: https://doi.org/10.1145/3425577.3425579

[8] J. Wu, H. Shi, and Q. Wang, "Contrabands detection in x-ray screening images using yolo model," in *Proceedings of the 4th International Conference on Computer Science and Application Engineering*, ser. CSAE 2020. New York, NY, USA: Association for Computing Machinery, 2020. [Online]. Available: https://doi.org/10.1145/3424978.3425106

[9] X. Xu, S. Wang, Z. Wang, X. Zhang, and R. Hu, "Exploring image enhancement for salient object detection in low light images," *CoRR*, vol. abs/2007.16124, 2020. [Online]. Available: https://arxiv.org/abs/2007.16124

[10] Y. Hu, Z. Ou, X. Xu, and M. Song, *A Crowdsourcing Repeated Annotations System for Visual Object Detection*. New York, NY, USA: Association for Computing Machinery, 2019. [Online]. Available: https://doi.org/10.1145/3387168.3387242

[11] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 779–788.

[12] R. Xu, H. Lin, K. Lu, L. Cao, and Y. Liu, "A forest fire detection system based on ensemble learning," *Forests*, vol. 12, no. 2, 2021. [Online]. Available: https://www.mdpi.com/1999-4907/12/2/217

[13] D. Thuan, "Do thuan evolution of yolo algorithm and yolov5: The state-of-the-art object detection algorithm evolution of yolo algorithm and yolov5: The state-of-the-art object detection algorithm," 2021.