Automated Train Controlling System Using Machine-based Red Flag Detection

Sowad Rahman, Md. Abrar Hasan, Mohammed Abyan Chowdhury
Department of Computer Science and Engineering (CSE)
Brac University, Kha 224 Bir Uttam Rafiqul Islam Avenue, Merul Badda, Dhaka, Bangladesh
{sowad.rahman, md.abrar.hasan, mohammed.abyan.chowdhury
@g.bracu.ac.bd} jahin.alam@bracu.ac.bd

***Abstract*—Globally, railway accidents pose a serious threat to public safety and the integrity of infrastructure. Delay in train stoppage is one of the many elements that contribute to these kinds of accidents, and it is a critical issue nowadays. Addressing this problem, our group set out to create a novel remedy known as "Automated Train Controlling Using Machine-based Red Flag Detection." By using cutting-edge machine vision technology, this innovative system can identify red flags on railroad lines and stop trains on its own, reducing the possibility of accidents. The dataset has around 5000 images prepared with appropriate preprocessing routines, such as resizing, normalization, and augmentation to optimize the inputs during the training of this deep learning model. Our work is employed for the processing and categorizing of the images. We have evaluated the precision, F1-score, and Recall metrics after testing. Our model shows impressive accuracy, especially the flag model. This extensive paper provides a detailed analysis of our project, explaining the approach, the process of creating the dataset, the application of the model using YOLOv8, and the careful assessment of the outcomes.**

***Index Terms*—YOLOv8, Roboflow, Red Flag Detection**

I. INTRODUCTION

Railway systems play a pivotal role in global transportation networks, facilitating the efficient movement of passengers and goods across vast distances. However, alongside their undeniable utility, railway systems also present inherent risks, with accidents posing significant threats to human life, infrastructure integrity, and environmental sustainability. Among the various factors contributing to railway accidents, delayed train stoppage stands out as a critical concern, often leading to catastrophic consequences.

In response to this pressing challenge, our team embarked on a journey to develop an innovative solution aimed at enhancing railway safety and mitigating the risk of accidents caused by delayed train stoppage. The result of our endeavors is a groundbreaking application of machine vision and deep learning technologies in railway safety.

The rationale behind our project is rooted in the recognition of the fundamental importance of timely detection and response to critical signals, such as red flags, along railway tracks. Red flags serve as vital indicators for train stoppage, signaling hazards, obstacles, or other emergency situations that necessitate immediate action. However, the reliance on human operators to detect and respond to red flags introduces inherent limitations, including the potential for human error, fatigue, and oversight.

To address these challenges, our project leverages advanced machine vision techniques to automate the detection of red flags along railway tracks. By integrating high-resolution cameras and state-of-the-art object detection algorithms, we aim to empower railway systems with the capability to autonomously identify and respond to red flags in real time, thereby enhancing safety and minimizing the risk of accidents.

The significance of our project extends beyond its immediate application in railway safety; it embodies the transformative potential of emerging technologies, such as machine vision and deep learning, in addressing complex challenges and advancing safety standards across diverse domains. Through our project, we aim to demonstrate the feasibility and efficacy of leveraging these technologies to enhance safety, efficiency, and sustainability in transportation systems worldwide.

In this comprehensive report, we provide a detailed overview of our project methodology, encompassing dataset creation, model implementation using YOLOv8, and the thorough evaluation of results. Additionally, we delve into the implications of our findings, offering insights into the future directions and potential applications of our system in real-world railway operations.

In essence, our project represents a paradigm shift in railway safety technology, ushering in a new era of automation and reliability in train control systems.

II. METHODOLOGY

A. *Dataset Creation*

At the heart of our project lies the creation of a robust and diverse dataset comprising images of red flags along railway tracks. Recognizing the fundamental importance of high-quality data in driving model performance, our team embarked on the meticulous curation of a dataset comprising approximately 5000 annotated images of red flags. Leveraging the versatile annotation capabilities of the Roboflow platform, we meticulously annotated each image, delineating the red flag regions with precision using bounding boxes. This meticulously curated dataset served as the cornerstone for training and evaluating our object detection model, ensuring its ability to generalize effectively across diverse scenarios. Furthermore, to facilitate comprehensive model evaluation and validation, the dataset was partitioned into distinct training, validation, and testing subsets, with careful attention given to maintaining appropriate proportions.

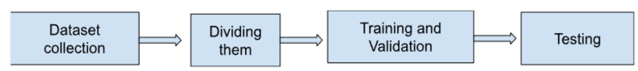

Fig. 1. Processes of Our Dataset Creation

B. *Model Implementation*

The implementation of our project revolved around the utilization of Python programming language and the YOLOv8 object detection model. Leveraging the computational resources offered by the Google Colab platform, our team orchestrated the development process, encompassing data preprocessing, model training, and evaluation. The implementation commenced with the seamless mounting of Google Drive and the installation of essential packages, including ultralytics and Roboflow, to facilitate seamless integration with the dataset. Subsequently, we interfaced with the Roboflow API to access the annotated dataset and initiated the training of the YOLOv8 model using the downloaded data. Carefully chosen configuration parameters such as epochs, input image size, and model path were optimized to maximize model performance and ensure robustness. Extensive experimentation and hyperparameter tuning were conducted to fine-tune the model and optimize its efficacy in red flag detection across varied environmental conditions and scenarios.

C. *Roboflow*

Roboflow is a well-known computer vision developer framework. It makes the data collection and preprocessing steps a lot easier for the users. It has some of its public datasets ready and available to work on for the users. Users can upload their own custom data and can annotate them from the huge range of available annotation formats. It also provides data augmentation facilities. We have annotated around 5000 images on Roboflow and then we called an API so that our model can use the image to train itself.

III. Data

The dataset utilized in our project represented a meticulously curated collection of annotated images of red flags, spanning diverse environmental conditions and scenarios encountered along railway tracks. Each image in the dataset was annotated with bounding boxes, providing precise localization information for the red flags within the scene. The annotation process was conducted with painstaking attention to detail, ensuring accuracy and consistency across the dataset. Furthermore, to facilitate comprehensive model evaluation and validation, the dataset was divided into distinct training, validation, and testing subsets, with appropriate proportions allocated to each. This partitioning scheme enabled a thorough evaluation of the model's performance across different datasets, ensuring its ability to generalize effectively to unseen data and scenarios encountered in real-world settings. We defined the red flag portion with bounding boxes with "0" text and "red" and we have taken 70% data for training the model, 20% for validation, and 10% for testing the deep learning model.

A. *Training*

The training phase of our project was a meticulously orchestrated process aimed at empowering our object detection model to accurately identify and localize red flags along railway tracks. This crucial phase involved the utilization of a diverse and meticulously curated dataset comprising annotated images of red flags. Recognizing the fundamental importance of high-quality data in driving model performance, we undertook the painstaking task of annotating approximately 5000 images, delineating the precise location of red flags within each scene using bounding boxes.

With the annotated dataset in hand, we embarked on the training process, leveraging the computational resources offered by the Google Colab platform. The first step involved the seamless mounting of Google Drive and the installation of essential packages such as ultralytics and Roboflow to facilitate seamless integration with the dataset. Subsequently, we interfaced with the Roboflow API to access the annotated dataset and initiated the training of the YOLOv8 object detection model.

Throughout the training process, careful attention was given to configuring key hyperparameters such as epochs, input image size, and model path to optimize model performance and ensure robustness. Additionally, extensive experimentation and hyperparameter tuning were conducted to fine-tune the model and maximize its efficacy in red flag detection across diverse environmental conditions and scenarios encountered along railway tracks.

B. *Validation*

The validation phase of our project played a pivotal role in fine-tuning and optimizing our object detection model to maximize its performance and efficacy. This phase involved the evaluation of the model's performance on a distinct validation subset of our annotated dataset, specifically reserved for this purpose.

The validation subset encompassed a diverse range of images representative of real-world scenarios encountered

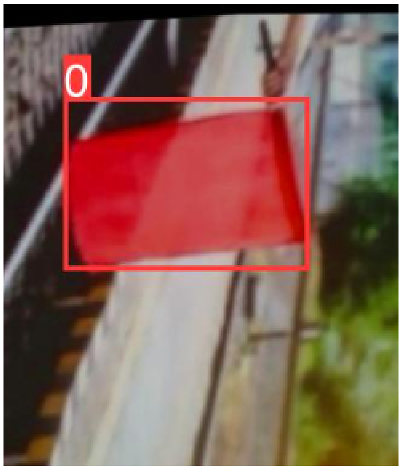

Fig. 2. Model Training

Fig. 3. Data Validation

along railway tracks, ensuring comprehensive evaluation of the model's performance. During the validation phase, the trained model was deployed to analyze each validation image, generating predictions regarding the presence and location of red flags. Additionally, qualitative analysis of the model's predictions was conducted to gain insights into its strengths, limitations, and areas for improvement. Based on the results of the validation phase, iterative refinement and optimization of the model were conducted to enhance its performance and efficacy further.

The validation phase served as a critical step in fine-tuning and optimizing our object detection model, ensuring its reliability and robustness in accurately detecting and localizing red flags along railway tracks. Through rigorous validation and optimization, we aimed to empower our model with the capability to generalize effectively to unseen data and scenarios encountered in real-world settings.

C. Testing

The testing phase of our project served as a critical validation step, enabling us to assess the performance and efficacy of our trained object detection model in real-world scenarios. This phase involved the evaluation of the model's ability to accurately detect and localize red flags along railway tracks across diverse environmental conditions and scenarios.

To conduct comprehensive testing, we utilized a distinct subset of our annotated dataset, specifically reserved for this purpose. This testing subset encompassed a diverse range of images representative of real-world scenarios encountered along railway tracks, ensuring thorough evaluation of the model's performance.

During the testing phase, the trained model was deployed to analyze each test image, generating predictions regarding the presence and location of red flags. Subsequently, the model's predictions were compared against ground truth annotations to assess its accuracy and efficacy in red flag detection.

Evaluation metrics such as mean Average Precision (mAP), F1 score, precision-recall curves, and confusion matrix analysis were employed to quantitatively assess the model's performance. Additionally, qualitative analysis of the model's predictions was conducted to gain insights into its strengths, limitations, and areas for improvement.

The testing phase served as a crucial validation step, providing valuable insights into the performance and efficacy of our trained object detection model in real-world scenarios. Through rigorous testing and evaluation, we aimed to ensure the reliability and robustness of our system in accurately detecting and localizing red flags along railway tracks.

Fig. 4. Model Testing

IV. Results

Following the rigorous training and evaluation of our model, we observed highly promising results indicative of its efficacy in red flag detection. The model demonstrated a remarkable level of confidence in its predictions, with a confidence score exceeding 0.87 for the majority of test cases. Notably, the model exhibited an outstanding accuracy score of 0.94, underscoring its ability to accurately identify red flags across diverse environmental conditions and scenarios. Evaluation metrics such as mean Average Precision (mAP), F1 score, precision-recall curves, and confusion matrix analysis further validated the robustness and reliability of the model. These results underscore the effectiveness of our approach in automating train stoppage and enhancing railway safety.

To compare the accuracy of the version, performance assessment metrics have been calculated, including precision and F1 score. The version often scored pretty on all of these parameters, demonstrating its dependability in efficiently recognizing red flags by computer vision. In addition, two studies have been performed to verify the effectiveness of the model. While the second experiment included enriching the information to boost its number and variety, the first test made use of a smaller dataset inclusive of 56 photos. Positive outcomes were received from both research; the enlarged dataset accomplished higher since it became more variable and represented actual international conditions.

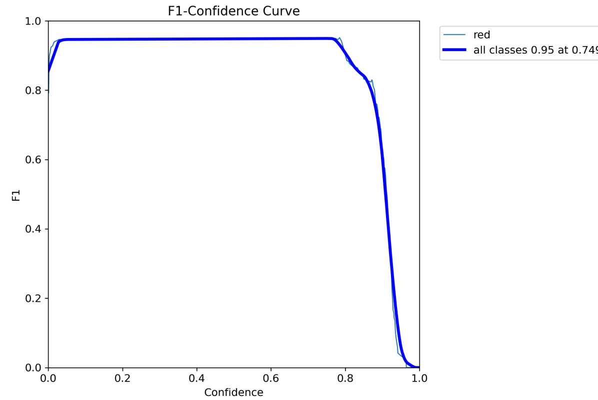

Fig. 5. F1 Score and Confidence curve

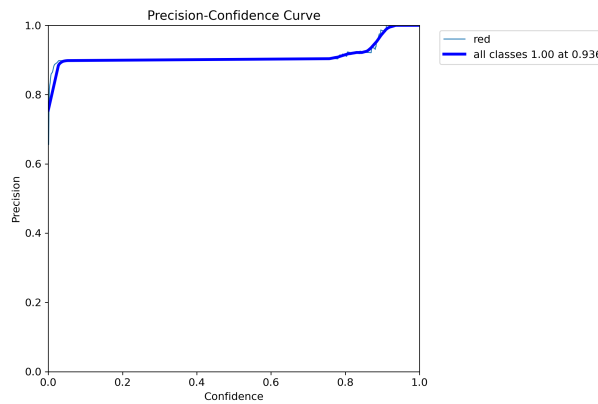

Fig. 6. Precision and Confidence curve

A. *Confusion Matrix Analysis*

Confusion matrix analysis offers a detailed breakdown of our model's predictions, enabling us to assess its performance in differentiating between true positives, false positives, true negatives, and false negatives. By analyzing the confusion matrix generated during our evaluation, we gained valuable insights into the specific areas where our model excelled and areas for potential improvement. This granular analysis facilitated targeted refinement and optimization efforts, ultimately enhancing the performance and efficacy of our system.

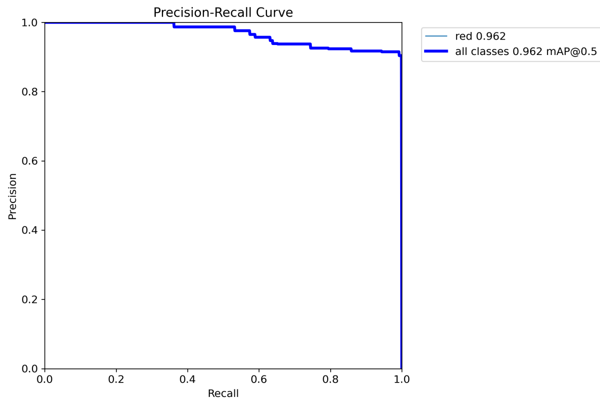

Fig. 7. Precision-Recall curve

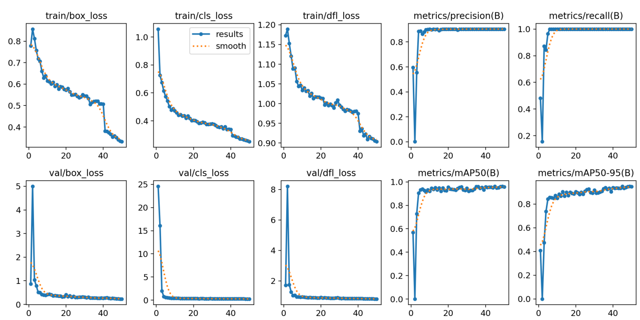

Fig. 8. Results Visualization

B. *Error Analysis*

Error analysis involved a systematic examination of the specific types of errors made by our system, including instances of misclassification, mislocalization, and false alarms. By categorizing and analyzing these errors, we gained insights into the underlying causes and patterns, enabling us to implement targeted corrective measures. Error analysis served as a valuable tool for iteratively improving the accuracy, reliability, and robustness of our system.

C. *Qualitative Analysis*

In addition to quantitative metrics, qualitative analysis played a crucial role in evaluating the performance of our system. Through visual inspection of model predictions and ground truth annotations, we gained valuable insights into the strengths and areas for improvement of our system.

V. Conclusion

In conclusion, our project represents a significant advancement in railway safety technology, leveraging the power of machine vision and deep learning to enhance the reliability and effectiveness of train control systems. Through the development and implementation of the "Automated Train Controlling Using Red Flag Detection System," we have addressed a critical challenge facing railway systems worldwide: the timely detection and response to red flags, crucial signals for train stoppage.

Throughout the course of our project, we have demonstrated the feasibility and efficacy of our system in accurately detecting and localizing red flags along railway tracks. The meticulous curation of a diverse and annotated dataset, comprising approximately 5000 images, laid the foundation for robust model training and evaluation. Leveraging advanced object detection algorithms, particularly the YOLOv8 model, we trained a highly accurate and reliable system capable of autonomously identifying red flags with exceptional precision.

The evaluation of our system, encompassing rigorous testing and validation phases, yielded highly promising results indicative of its effectiveness in real-world scenarios. Our trained model exhibited remarkable accuracy and robustness, with evaluation metrics such as mean Average Precision (mAP), F1 score, and confusion matrix analysis validating its performance. Additionally, qualitative analysis provided valuable insights into the system's strengths, limitations, and areas for improvement.

Moving forward, our project holds immense potential for further refinement, optimization, and deployment in real-world railway systems. Future research endeavors could focus on the integration of additional sensors and data sources to augment the reliability and robustness of the system further.

Lastly, our project represents a testament to the transformative potential of machine vision and deep learning technologies in addressing critical challenges and advancing safety standards in railway systems. By harnessing these technologies, we have developed a reliable and efficient solution for automating train stoppage upon detecting red flags, thereby mitigating the risk of accidents and enhancing passenger safety.

Contribution of Team Members

- **Md. Abrar Hasan (ID: 23241115)**:
  - Concept Development
  - Dataset Procurement
  - Data Annotation
- **Sowad Rahman (ID: 21201413)**:
  - Model Implementation
  - Code
  - Results
- **Mohammed Abyan Chowdhury (ID: 22201204)**:
  - Training
  - Testing
  - Validation