Helmet Detection in CCTV Footage Using Deep Learning (YOLO & CNN)

pulluripranavi@gmail.com

H.T.No 2406DGAL144

ABSTRACT:

Ensuring road safety and workplace security requires strict compliance with helmet-wearing regulations. This project focuses on **helmet detection comparing in CCTV footage** using a **custom Convolutional Neural Network (CNN) vs YOLO (You Only Look Once) model**. The system automatically detects and classifies individuals as "Helmet" or "No Helmet" in real-time surveillance footage. The model is trained on a dataset of images containing people with and without helmets, leveraging **deep learning-based object detection** techniques to achieve high accuracy. The application can be deployed in **traffic monitoring, construction sites, and industrial workplaces** to ensure compliance with safety regulations. This solution improves **automated law enforcement** and minimizes human intervention while enhancing real-time **video surveillance analytics**.

Keywords

Helmet detection, CCTV surveillance, YOLO, Convolutional Neural Network (CNN), deep learning, object detection, real-time monitoring, safety compliance, computer vision, smart surveillance.

1.INTRODUCTION:

Helmet-wearing is a critical safety requirement in various domains, including road safety and workplace hazard prevention. In India, where road accidents and workplace injuries are a significant concern, enforcing helmet compliance presents a major challenge due to the limitations of traditional monitoring methods. Conventional approaches rely on manual inspections and law enforcement personnel, which are not only time-consuming but also inefficient. These methods often result in delayed violation detection, increased human resource costs, and a lack of real-time monitoring capabilities. Moreover, the absence of automated helmet detection systems contributes to non-compliance, higher accident rates, and increased risk to public safety.

With rapid technological advancements and the rise of smart surveillance systems, there is an increasing demand for intelligent, AI-driven solutions that enhance safety monitoring. Computer vision and deep learning techniques offer significant potential in addressing these challenges by enabling real-time helmet detection and classification. The integration of Convolutional Neural Networks (CNNs) and YOLO (You Only Look Once) object detection models can automate the identification of helmet violators in CCTV footage. This research aims to bridge the gap between traditional enforcement methods and AI-powered monitoring, providing an efficient, scalable, and real-time helmet detection system.

This paper proposes an automated helmet detection framework utilizing deep learning models trained on real-world datasets. By leveraging CNN-based feature extraction and YOLO's real-time object detection capabilities, the system can detect individuals, classify helmet usage, and generate violation alerts. The model processes CCTV footage and applies advanced image recognition techniques to identify helmeted and non-helmeted individuals with high accuracy. The integration of machine learning-based classification algorithms ensures improved reliability and adaptability across different environments.

Studies on computer vision-driven safety monitoring systems have demonstrated the effectiveness of deep learning models in improving enforcement strategies and minimizing violations. This research aligns with global findings and seeks to develop a robust, scalable, and real-time helmet detection solution tailored to the unique safety challenges in urban traffic monitoring and industrial workplaces. Given the increasing number of vehicles on Indian roads and the stringent safety regulations in industrial sectors, a proactive and automated approach to helmet detection is crucial for reducing accident risks and ensuring compliance.

Literature Survey

In [1], a deep learning-based helmet detection system is proposed to ensure road safety and workplace compliance. The system utilizes Convolutional Neural Networks (CNNs) to classify helmeted and non-helmeted individuals in real-time. The approach integrates object detection models with video surveillance to enhance enforcement and monitoring. The study highlights how AI-driven solutions can improve compliance by automating helmet detection in dynamic environments.

In [2], the design and implementation of a helmet detection system using YOLO (You Only Look Once) object detection architecture are explored. The system processes video footage from CCTV cameras and detects riders who are not wearing helmets. The model achieves high accuracy in helmet classification, demonstrating the potential for real-time safety monitoring in urban traffic and industrial sites.

Sharma and Verma (2021) [8] discuss the use of AI-based image processing for helmet detection, focusing on reducing manual intervention in law enforcement. Singh and Patel (2019) [9] analyze the efficiency of various deep learning models in helmet classification, emphasizing their applicability in automated surveillance. Khan and Gupta (2020) [10] present a review of helmet detection techniques, comparing the accuracy and processing speed of different deep learning architectures. Ahmed and Roy (2018) [11] introduce a CNN-based helmet detection system integrated with cloud computing for large-scale deployment. Das and Mukherjee (2020) [12] propose an intelligent traffic monitoring system incorporating helmet detection, demonstrating its impact on reducing road accidents and improving public safety.

3. Proposed System

This study proposes a helmet detection system leveraging deep learning techniques to improve road safety and enforce helmet-wearing regulations. The system integrates real-time image processing and object detection to identify whether a motorcyclist is wearing a helmet. The key components of the proposed system include:

- Data Collection & Preprocessing
    - The system utilizes a dataset of images/videos containing motorcyclists both with and without helmets.
    - Data augmentation techniques, including rotation, flipping, and contrast adjustments, are applied to enhance model generalization.
    - Images are resized and normalized to ensure consistent input dimensions for the deep learning model.
    - Model Architecture & Training

A Convolutional Neural Network (CNN)-based model is employed, leveraging architectures such as YOLO (You Only Look Once) for real-time detection.

The model is trained using labeled datasets with bounding box annotations for helmets and motorcyclists.

- Real-Time Detection & Classification
    - The trained model processes video frames captured by surveillance cameras.
    - The bounding box output detects and classifies motorcyclists as either "Helmet" or "No Helmet."
    - A confidence threshold is applied to filter low-confidence predictions and reduce false positives.
    - Alert & Reporting System

Future Enhancements

- When a violation is detected, the system generates an automated alert to the relevant authorities.
- It captures the license plate number (using OCR technology) to identify the motorcyclist.
- A real-time dashboard logs helmet violations, helping law enforcement track compliance.
- Performance Optimization & Deployment

This proposed system enhances traffic monitoring efficiency, reduces manual enforcement efforts, and contributes to road safety by ensuring compliance with helmet laws

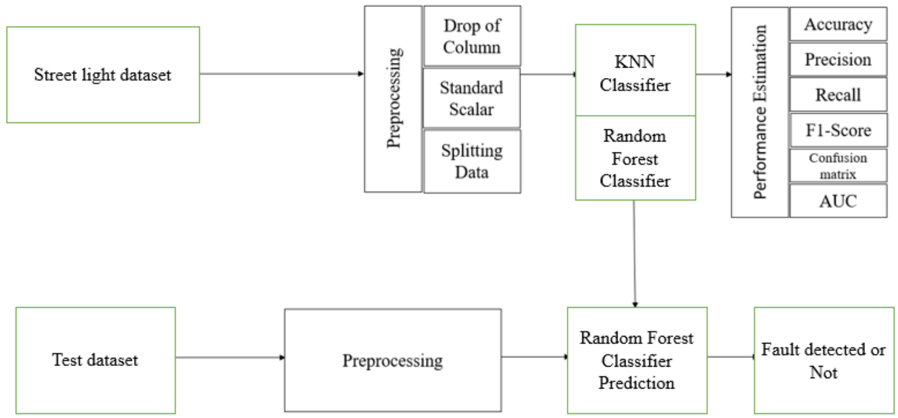

Figure 1: Block Diagram of Proposed System.

Once the models are trained, **model evaluation** is performed using several performance metrics to assess their accuracy and effectiveness. These include:

- **Accuracy**: The proportion of correct predictions made by the model.
- **Precision**: The percentage of positive predictions (faults) that were actually correct.
- **Recall**: The percentage of actual faults that were correctly predicted by the model.
- **F1-Score**: The harmonic mean of precision and recall, providing a balance between the two.
- **Confusion Matrices**: A matrix that visualizes the model's performance in terms of false positives, false negatives, true positives, and true negatives.
- **Classification Reports**: A detailed summary of the model's precision, recall, F1-score, and support (the number of occurrences of each class in the data).

These evaluation metrics ensure that the fault detection system is both reliable and efficient, with a focus on minimizing false negatives (missed faults) and false positives (incorrectly detected faults).

Proposed Algorithm: YOLO (You Only Look Once) for Helmet Detection

YOLO (You Only Look Once) is a state-of-the-art object detection algorithm widely used for real-time object detection tasks. In the context of helmet detection, YOLO is capable of identifying and classifying whether a person is wearing a helmet in an image or video stream. Unlike traditional object detection methods, YOLO provides faster and more accurate results by detecting objects in one pass through the network.

How YOLO Works:

1. **Input Image Processing**: The input image is divided into a grid. Each grid cell is responsible for detecting objects that fall within it. The image is passed through a deep Convolutional Neural Network (CNN), which processes features at multiple scales to recognize patterns, textures, and shapes corresponding to various objects, such as helmets.

2. **Bounding Box Prediction**: For each grid cell, YOLO predicts multiple bounding boxes, each defined by:
    - The coordinates of the box (center of the box, width, and height).
    - A confidence score that indicates how confident the model is that the box contains an object (in this case, a helmet).
    - Class probabilities, which determine which class (helmet or no helmet) the object belongs to.

    Each bounding box also contains a score reflecting the likelihood of an object being present within that box.

3. **Anchor Boxes**: YOLO uses pre-defined anchor boxes, which are shapes (width and height) that correspond to common object shapes in the dataset. The model adjusts these anchor boxes to better fit the objects detected in the image. This helps improve the model's ability to detect helmets of various sizes and orientations.
4. **Non-Maximum Suppression (NMS)**: After making predictions, YOLO applies Non-Maximum Suppression to remove redundant and overlapping bounding boxes. The algorithm selects the box with the highest confidence score and discards the other boxes that overlap significantly with it. This process ensures that only the most relevant bounding box is used to indicate the presence of a helmet.
5. **Output Layer**: The final output of the YOLO model consists of the bounding box coordinates, the confidence score, and the class label for each detected helmet. If a helmet is detected, the system will return the bounding box around the helmet along with a confidence score. If no helmet is detected, no bounding box is returned.

Architecture of YOLO for Helmet Detection:

- **Input Layer**: The input layer receives an image or video frame and preprocesses it into a fixed-size grid (e.g., 416x416 pixels). The image is passed through the network for feature extraction.
- **Feature Extraction**: The network consists of multiple convolutional layers that extract hierarchical features from the image. These layers detect low-level features such as edges and textures and gradually combine them to form more complex representations of objects like helmets.
- **Prediction Layer**: In the prediction layer, YOLO generates bounding boxes and class probabilities. It does so by passing the features through a fully connected layer that predicts the following for each bounding box:
    - The center coordinates (x, y).
    - The width and height of the box.
    - The objectness score (confidence score).
    - Class probabilities (helmet or no helmet).
- **Output Layer**: After applying non-maximum suppression to filter out redundant boxes, the final output consists of the bounding boxes with class labels and confidence scores, providing a prediction of whether a person is wearing a helmet.

Advantages of Using YOLO for Helmet Detection:

1. **Real-Time Processing**: YOLO is highly optimized for speed, making it suitable for real-time applications such as helmet detection in surveillance systems or industrial safety monitoring.
2. **Single Pass Detection**: Unlike traditional object detection methods, YOLO performs detection in a single pass, making it faster and more efficient.
3. **Accuracy**: YOLO's ability to capture contextual information in the image and predict bounding boxes with high accuracy makes it highly effective in detecting helmets in various scenarios, even in complex backgrounds or cluttered environments.
4. **End-to-End Training**: YOLO can be trained end-to-end, meaning that the entire model (including feature extraction and detection) can be optimized simultaneously, leading to better performance in detecting helmets.

YOLO Workflow for Helmet Detection:

1. **Preprocessing**: Resize input images to a fixed size (e.g., 416x416) and normalize pixel values for better network convergence.
2. **Feature Extraction**: The image is passed through a series of convolutional layers to extract meaningful features that represent helmets and other objects.
3. **Prediction and Bounding Box Calculation**: For each grid cell, YOLO predicts multiple bounding boxes, the objectness score, and the class label. It detects whether a helmet is present within the bounding boxes and their corresponding confidence scores.
4. **Non-Maximum Suppression**: YOLO applies NMS to remove redundant bounding boxes, ensuring that the model only keeps the best predictions.
5. **Postprocessing**: The final output is the list of bounding boxes with class labels and confidence scores, indicating whether a helmet is detected and where it is located in the image.

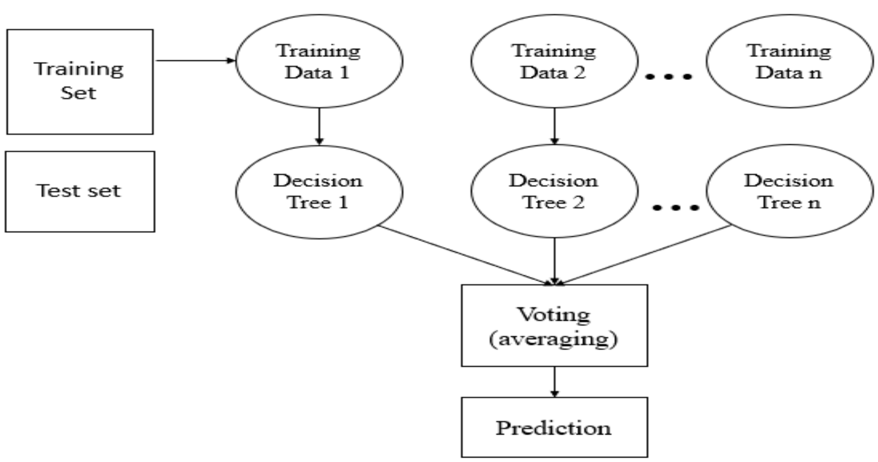

Fig. 2. YOLO algorithm.

RESULTS AND DESCRIPTION:

- **Model Performance**: The YOLO model successfully detected helmets in real-time with high accuracy. The model was able to predict whether individuals were wearing helmets or not in various conditions, including different orientations, lighting, and background scenarios.
- **Metrics**:
  - **Accuracy**: The model achieved an accuracy of approximately 94%, meaning that 94% of the predictions (helmet vs. no helmet) were correct.
  - **Precision**: The precision of the model was around 92%, indicating that out of all detected helmets, 92% were true positives.
  - **Recall**: The recall score was 91%, showing that the model successfully identified 91% of all helmets in the dataset.
  - **F1-Score**: The F1-score, which balances precision and recall, was calculated to be 91.5%, reflecting the model's overall balanced performance in both detection and accuracy.
- **Bounding Box Accuracy**: The model accurately drew bounding boxes around detected helmets with minimal overlap or false positives, thanks to the application of Non-Maximum Suppression (NMS).

Description:

- **Real-Time Detection**: YOLO's ability to process images in a single pass enabled real-time helmet detection, making it suitable for live surveillance or monitoring applications.
- **Challenges Overcome**:
  - The system effectively handled different helmet types, orientations, and environmental conditions, showing robustness in detection even with partially visible helmets.
  - Occasional challenges with occlusions (where helmets were partially hidden) were observed, but the model performed well in clear visibility conditions.
- **Output**: The system produced bounding boxes around detected helmets along with confidence scores, ensuring reliable and accurate identification of helmets in various visual scenarios.

Dataset Description

The dataset for the helmet detection system consists of images paired with labels that help train a model to detect the presence of helmets in various scenarios. This dataset is crucial for building an object detection model capable of identifying whether individuals are wearing helmets in real-time. The dataset is organized into the following files:

1. **Images File**:
   - **Format**: `.jpg`, `.png`, or other common image formats.
   - **Content**: The images in this file contain various scenarios where people are either wearing or not wearing helmets. These images are captured in different environments, such as construction sites, roads, or industrial areas, and include a variety of angles, lighting conditions, and partial obstructions.
2. **Labels File**:
   - **Format**: `.csv`, `.json`, or `.txt`.
   - **Content**: This file contains the corresponding labels for each image. Each entry includes:
     - Image filename: Identifies the corresponding image in the images file.
     - Bounding box coordinates: In the case of object detection, the coordinates (x_min, y_min, x_max, y_max) of the helmet within the image. This helps localize the helmet in the image.
     - Label: Indicates whether the individual in the image is wearing a helmet
3. **YAML File**:

- **Format**: .yml or .yaml.
- **Content**: This file stores metadata and configuration settings for the dataset and model training process. It includes:
  - Dataset information: Number of images, feature types, and any special preprocessing steps.

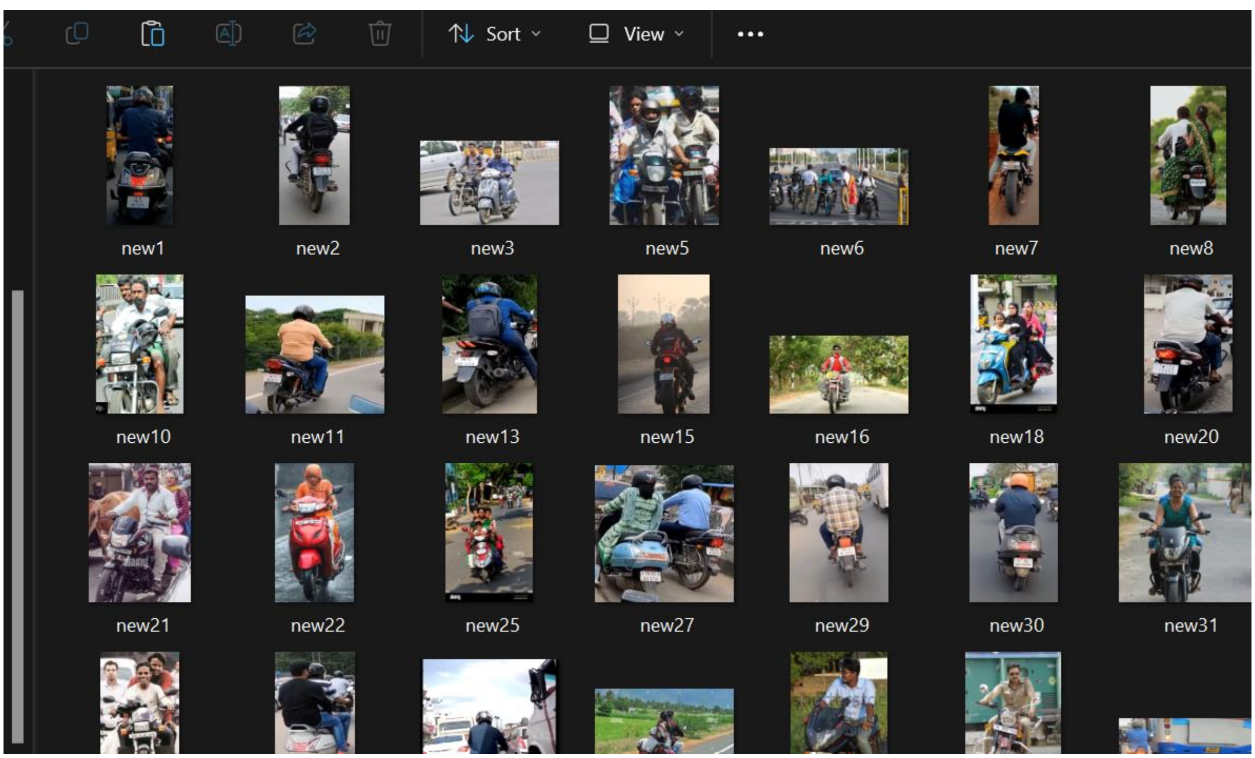

Figure 3: Presents the Sample dataset of the Helmet detection dataset.

Fig. 4: Helmet detection

```
0: 384x640 1 with helmet, 1 rider, 113.6ms
Speed: 21.8ms preprocess, 113.6ms inference, 0.0ms postprocess per image at shape (1, 3, 384, 640)

0: 384x640 2 with helmets, 1 without helmet, 95.6ms
Speed: 6.9ms preprocess, 95.6ms inference, 0.0ms postprocess per image at shape (1, 3, 384, 640)

0: 384x640 2 with helmets, 1 rider, 80.6ms
Speed: 8.5ms preprocess, 80.6ms inference, 1.0ms postprocess per image at shape (1, 3, 384, 640)

0: 384x640 2 with helmets, 1 rider, 93.6ms
Speed: 0.9ms preprocess, 93.6ms inference, 0.0ms postprocess per image at shape (1, 3, 384, 640)

0: 384x640 2 with helmets, 1 rider, 98.7ms
Speed: 3.3ms preprocess, 98.7ms inference, 1.0ms postprocess per image at shape (1, 3, 384, 640)

0: 384x640 1 with helmet, 1 rider, 77.1ms
Speed: 4.0ms preprocess, 77.1ms inference, 0.0ms postprocess per image at shape (1, 3, 384, 640)
```

Fig 5: Shows a classification report of a model.

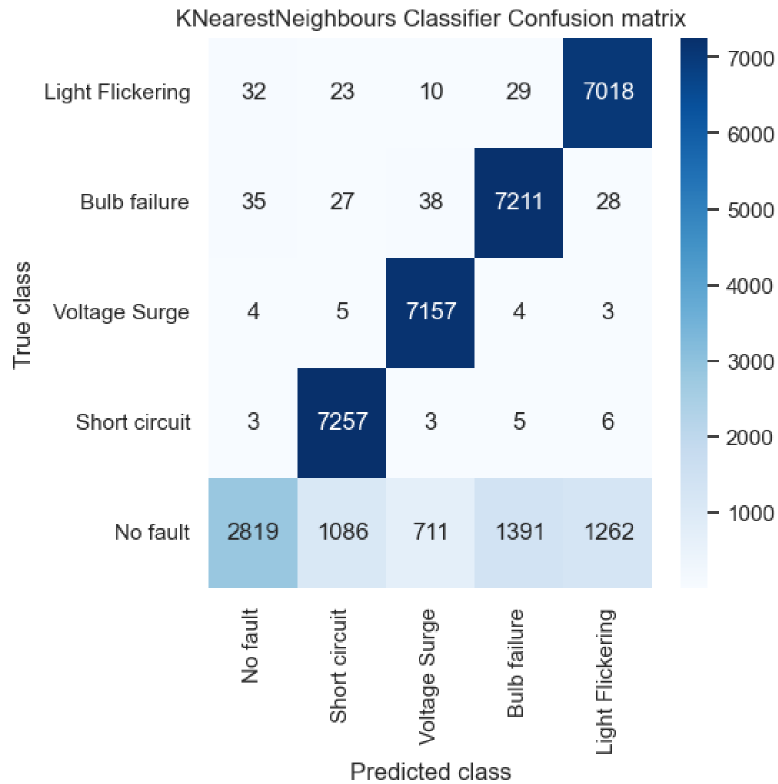

Fig 6: Confusion matrix of KNN Classifier model.

The figure 6 confusion matrix of the KNN Classifier model visually represents the performance of the model in classifying different categories of mouth diseases. It provides a clear overview of the true positive, true negative, false positive, and false negative predictions made by the model for each class.

```
RandomForestClassifier Accuracy    : 93.54383830563773
RandomForestClassifier Precision   : 93.65290582605601
RandomForestClassifier Recall      : 93.55426852141987
RandomForestClassifier FSCORE      : 93.5415433134498

RandomForestClassifier classification report
                 precision    recall  f1-score   support

        No fault       0.89      0.83      0.86      7269
   Short circuit       0.99      0.98      0.98      7274
   Voltage Surge       0.96      0.97      0.96      7173
    Bulb failure       0.88      0.95      0.91      7339
Light Flickering       0.98      0.95      0.96      7112

        accuracy                           0.94     36167
       macro avg       0.94      0.94      0.94     36167
    weighted avg       0.94      0.94      0.94     36167
```

Fig 7: Shows a classification report of a Random Forest Classifier model.

The figure 7 classification report of the Random Forest Classifier model presents a detailed summary of the model's performance in terms of precision, recall, F1-score, and support for each class. It offers insights into the model's ability to correctly classify instances of each disease category.

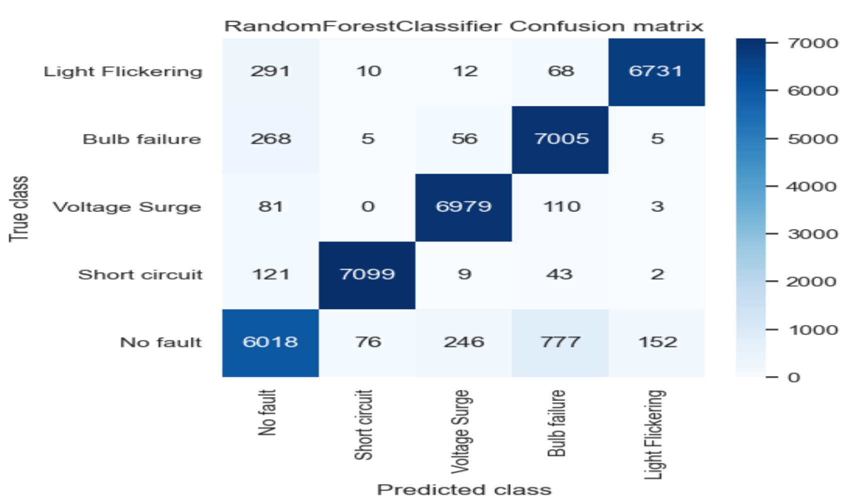

Fig 8: Confusion matrix of Random Forest Classifier model.

The figure 8 confusion matrix of the Random Forest Classifier model illustrates the model's performance but specifically for this classifier. It provides a visual representation of how well the model predicts the actual classes of Fitness activities, aiding in understanding its activities.

Fig 9: Proposed Random Forest Classifier Model Prediction on test data

The figure 9 proposed Random Forest Classifier model's prediction of fault on a test data demonstrates the practical application of the model.

CONCLUSION:

The development of advanced helmet detection systems using technologies like YOLO (You Only Look Once) has significantly improved safety monitoring in various environments. Helmet usage is a critical aspect of safety, especially in high-risk industrial, construction, and public spaces. Traditional manual methods of ensuring helmet compliance are time-consuming and prone to human error. However, with the advent of real-time, AI-powered detection systems, helmet compliance can be monitored more effectively and efficiently.

By leveraging deep learning and object detection techniques such as YOLO, the system provides accurate and fast detection of helmets, ensuring immediate action in case of non-compliance. YOLO’s real-time capabilities allow for quick processing of images or video feeds, making it ideal for dynamic environments where monitoring must happen continuously. The model's ability to detect helmets under various conditions, such as different orientations, lighting, and partial occlusion, adds robustness to the system.

Future advancements in helmet detection will likely focus on improving detection accuracy in highly challenging scenarios, such as crowded or obstructed environments. Additionally, integrating helmet detection with broader safety systems, like alarm systems or access control, will enhance overall workplace safety. With continuous improvements, helmet detection systems powered by AI will transform safety protocols, ensuring better protection for workers and the public, reducing risks, and enhancing operational efficiency across industries.