ASSAM POWER GENERATION CORPORATION LTD.
অসম শক্তি উৎপাদন নিগম লিমিটেড

BIJULEE BHAWAN, PALTANBAZAR, GUWAHATI-01

‘‘REAL-TIME PPE DETECTION AND SAFETY ALERT SYSTEM’’

WINTER INTERNSHIP PROJECT REPORT

by

NAME OF THE CANDIDATE(S)

Brajakishor Tamuli,ET22BTHCS023
Shimanta Baruah,ET22BTHCS029
Simanta Hazarika,ET22BTHCS043

Department of Computer Science and Engineering
The AsaamKaziranga University,
Jorhat, Assam,January - 2025

Acknowledgement

We would like to extend our sincere gratitude to Assam Power Generation Corporation Limited (APGCL) for providing us with the opportunity to undertake this internship and work on the project titled "Real-Time Personal Protective Equipment (PPE) Detection and Safety Alert System." This experience has been invaluable in enhancing our technical knowledge and practical skills.

We are profoundly thankful to our project supervisors, Ms. Prarthana Kalita and Mr. Sibasish Das, for their unwavering guidance, encouragement, and insightful contributions throughout the duration of this project. Their expertise and support have been crucial to the successful completion of our work.

Additionally, we would like to express our appreciation to the IT Department for supplying the necessary resources and fostering a conducive learning environment during our internship. Their assistance and feedback have significantly contributed to the advancement of this project.

Finally, we are grateful to our families and friends for their constant encouragement and motivation, which kept us focused and determined throughout this journey. We would also like to thank the evaluation panel for their constructive feedback on the project, which has been instrumental in refining our work by highlighting areas for improvement.

Abstract

Ensuring workplace safety is paramount, particularly in high-risk sectors such as power generation. A fundamental safety measure involves the use of appropriate personal protective equipment, including hard-helmets and safety vests, to mitigate the risk of injuries. However, the manual oversight of hard-helmet and safety vest compliance can be both challenging and labor-intensive.

To address this issue, we introduce the AI-Powered Safety Helmet Monitoring System. This innovative project leverages artificial intelligence and computer vision technologies to automatically ascertain whether individuals in a designated work area are wearing their safety helmets. The system processes live video feeds or images, and upon detecting a person without a helmet and a vest, it promptly issues an alert.

The primary objective of this system is to enhance safety and operational efficiency in workplaces, such as those operated by Assam Power Generation Corporation Limited (APGCL), by automating the monitoring of helmet usage. This approach significantly diminishes the necessity for manual inspections, thereby ensuring compliance with safety regulations while minimizing the reliance on human intervention. This report details the design, implementation, and testing phases of the system, illustrating its potential to elevate safety standards and safeguard workers effectively.

Contents

1. Introduction

Assam Power Generation Corporation Ltd. was established following the unbundling of the Assam State Electricity Board (ASEB) in December 2004, as part of the State Power Sector Reform Programme in accordance with the Electricity Act of 2003. The certificate for the commencement of business was secured effective April 29, 2004. The final transfer scheme was executed in August 2005, accompanied by a new company balance sheet effective April 2005. The corporation is primarily tasked with maximizing energy generation to fulfill the energy demands of the state.

2. Vision and mission

APGCL envisions a future of reliable and sustainable energy for all, supporting the economic growth and quality of life in Assam. Its mission is to enhance power generation capacity through efficient resource management, innovation, and adoption of environmentally friendly practices. The corporation is dedicated to maintaining high safety standards, optimizing performance, and integrating renewable energy sources.

3. Functions

Operation & Maintenance of all existing Power Generation Plants & electrical system infrastructure associated with evacuation of power within the power station. Completion of on-going Power Projects so approved by Govt. of Assam (GoA) and arrangement of necessary power evacuation in co-ordination with Assam Electricity Grid Corporation Ltd. Renovation &Modernisation (R&M) of existing Power Stations in the State of Assam to ensure reliable power generation. To implement the energy policy of the State so set by Govt. of Assam. To acquire financial support for capacity addition in Power Generation from GoA. To obtain GoI clearances and approvals for power projects through Govt. of Assam. To acquire land for establishing power projects in support of the Power Dept., GoA. To adhere to the Regulatory Framework. To conduct Energy Audit in all Power Stations of APGCL as per Energy Conservation Act, 2001 and to take remedial measures to curb energy waste if any.

4. Objectives

Adequate availability of power by developing new Power Projects. Human Resources Development /Management -Reduction of Establishment cost. To avoid cost and time overruns on the schemes under execution through effective monitoring Systems. Best practices for Repair & Maintenance (R&M) of Power Houses & Electrical Infrastructure to improve quality and reliability of power. Other Mandatory Objectives.

5. Project Introduction

5.1 Background of the Project :

In industries, especially those dealing with high-risk operations like power generation, ensuring the safety of workers is crucial. Wearing the right protective gear, such as safety helmets, is a fundamental part of workplace safety. However, manually monitoring safety compliance can be time-consuming and prone to human error. This is where modern technology, like artificial intelligence, can make a significant difference.

The AI-Powered Safety Helmet Monitoring System was developed to automate the process of ensuring workers wear their helmets in hazardous areas. This system leverages computer vision and machine learning to detect whether individuals are wearing helmets and send real-time alerts if they are not.

5.2 Significance of the Project :

Safety measures in industrial environments are essential to protect workers from injuries and accidents. Industries such as power generation, construction, and manufacturing are inherently dangerous, with workers facing risks like falling objects, electrical hazards, and machinery related accidents.

By ensuring that all workers are wearing proper safety gear, companies can minimize the risk of accidents and create a safer working environment. In addition to reducing workplace injuries, strict safety measures also help companies comply with regulatory standards and avoid legal liabilities.

5.3 Objective of the PPE Detection System :

The main objective of this project is to design and implement a system that can automatically detect whether workers are wearing safety equipments. Using AI and computer vision, the system analyses real-time video footage or images to identify helmet usage, alerting supervisors if a worker is not wearing a helmet. This not only saves time and effort but also helps ensure compliance with safety protocols. Automate the monitoring of PPE usage to ensure adherence to occupational safety standards and regulations. This system also provide immediate notifications on email to the supervisor when PPE violations are detected, allowing for timely corrective actions.

This system is reduce the need for manual monitoring and inspections, lowering operational costs while maintaining high safety standards.

This system is especially benefitialin organizations like APGCL by enhancing workplace safety and reducing the chances of accidents due to non-compliance with helmet usage.

6. Project Planning

6.1 Planning:

The planning phase is essential for the successful execution of the PPE Detection Project. This phase encompasses the definition of project scope, objectives, timelines, resources, and risk management strategies to ensure efficient and effective achievement of project goals. The Agile methodology has been selected for this project due to its flexibility and iterative nature, which is particularly advantageous for the development and refinement of machine learning models.

6.2 Project Objectives:

The primary objective of this project is to develop and implement a PPE detection system that enhances workplace safety by automatically identifying whether workers are wearing the required personal protective equipment (PPE). This system aims to reduce workplace accidents and ensure compliance with safety regulations.

6.3 Scope:

<u>In-Scope</u>:
- Detection of PPE items, including helmets and vests.
- Real-time detection and alert system.
- User interface for monitoring and reporting.
- Data storage and retrieval system for detection results.

<u>Out-of-Scope:</u>
- The project will not encompass the installation of physical security measures (e.g., fences, barriers).
- The project will not involve the development of PPE items or the manufacturing of safety gear.
- Any integrations with third-party safety systems beyond the defined scope.

6.4 Resources:

<u>Equipment:</u>
- Cameras with above 8MP(1080p)
- O.S (Windows)
- Network Equipment

Software:
- Machine Learning Frameworks:PyTorch,, Scikit-Learn
- Project Management Tools: PyCharm

6.5 Timeline:

Phase	Start Date	End Date
Planning	17 December, 2024	24 December, 2024
Requirements	25 December, 2024	28 December, 2024
Design	28 December, 2024	31 December, 2024
Implementation	12 January, 2025	14 January, 2025
Testing	15 January, 2025	17 January, 2025
Deployment	-	-
Maintenance	ongoing	-

Potential Risks:
- Delays in hardware procurement.
- Technical challenges related to algorithm accuracy.
- Resistance to change from end-users.

Mitigation Strategies:
- Establish contingency plans for hardware delays.
- Allocate additional time for algorithm fine-tuning and testing.
- Provide comprehensive training and support to end-users to facilitate the transition.

6.6 Agile Methodology :

Agile Methodology in PPE Detection Project: In our PPE detection project, we adopted the Agile methodology to ensure a flexible, iterative, and collaborative approach to development. Agile was chosen due to its ability to adapt to changing requirements and its emphasis.

Development Process: The development process was iterative, with continuous development, testing, and integration of features. Daily standup meetings were held to discuss progress, identify obstacles, and coordinate efforts among team members.

Testing and Quality Assurance: Testing was integrated into the development process, with unit tests, integration tests, and system tests conducted to ensure the quality of the PPE detection system. Bugs were tracked and addressed during each sprint.

Collaboration and Communication Stakeholders were involved throughout the project, providing feedback and guiding development priorities. Collaboration within the development team was facilitated using PyCharm and task management software.

Outcome and Evaluation The project was successful in achieving its goals, with Agile methodology playing a key role in adapting to changing requirements and ensuring continuous feedback. The lessons learned from using Agile in this project will inform future development efforts.

By adopting the Agile methodology, the project aims to deliver a high-quality PPE detection system that effectively meets user needs and adapts to evolving requirements.

7 Requirement Analysis for PPE Detection Project

7.1 Introduction

The PPE Detection Project aims to develop a system that identifies and verifies the usage of personal protective equipment (PPE) in workplace environments using real-time video surveillance.

7.2 Stakeholder Identification

- Safety Officer: Responsible for ensuring workplace safety.
- IT Department: Responsible for system implementation and maintenance.
- Workers: End-users who need to comply with PPE regulations.
- Management Team: Oversees the project and ensures it aligns with company objectives.

7.3 Objectives and Goals

The primary objective is to enhance workplace safety by automatically detecting PPE compliance and generating alerts for non-compliance.

7.4 Functional Requirements

1. The system should detect hard hats, safety goggles, and vests in real-time from video feeds.
2. The system should generate alerts when non-compliance is detected.
3. The system should store detection logs in a database.
4. The system should provide a user interface for viewing detection results and generating reports.

7.5 Non-Functional Requirements

- The system should process video feeds in real-time with a detection accuracy of at least 95%.
- The system should be available 24/7 and have a downtime of less than 1% per month.
- The user interface should be intuitive and require minimal training for users to operate.

7.6 Technical Requirements

- Hardware: PC with a minimum of 16GB RAM, 4 CPUs, and a GPU for real-time video processing(Nvidia gpu would be great).
- Software: windows 7-11, Python, TensorFlow, and OpenCV.

7.7 Assumptions and Constraints

- Assumption: The company already has a functioning CCTV system in place.
- Constraint: The system must comply with the company's IT security policies.

7.8 Risk Analysis

- Potential Risk: Detection accuracy may be affected by poor lighting conditions in the workplace.

8 System design and methodology

8.1 Description of the Approach Used :

The AI-Powered Safety PPE Monitoring System follows a straightforward yet effective approach to ensuring that workers in an industrial setting are wearing their safety helmets. The system relies on computer vision and artificial intelligence techniques to process and analyse images or video streams in real-time, detecting whether or not PPE is being worn.

To achieve this, we implemented an image classification model that is capable of recognizing hard-helmets and safety vests in images captured from cameras placed in various industrial locations. When the system identifies an individual not wearing a helmet, it triggers an alert, notifying the safety personnel to take necessary action.

8.2 Tools, Technologies, and Hardware Used :

To build and deploy the system, we used the following tools and technologies:

- <u>Python</u>: The primary programming language used for developing the system. It offers a wide range of libraries for computer vision and machine learning. We have used the python 3.12 in the project and the IDE as Pycharm community version.

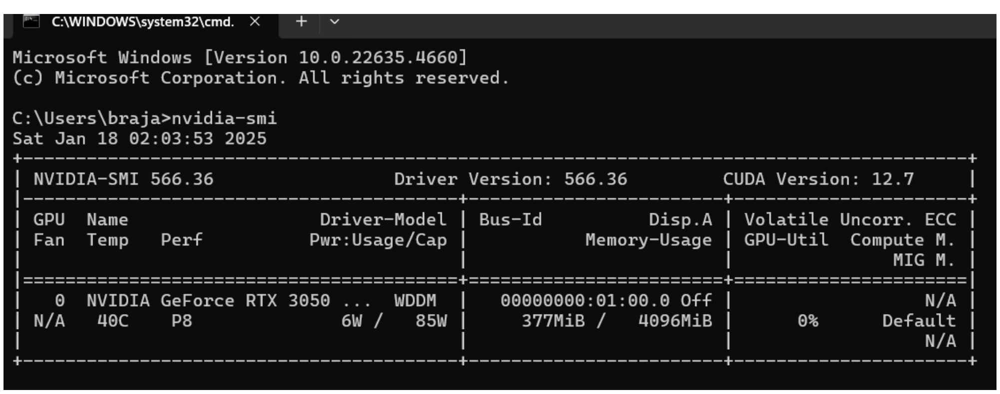
- <u>OpenCV</u>: An open-source computer vision library used for image processing tasks, such as capturing images, detecting objects, and manipulating video streams.

- <u>YOLO (You Only Look Once):</u> A state-of-the-art object detection algorithm that allows real-time detection of helmets in the video feed. We used a pre-trained YOLO model to detect the presence of helmets. We have used yolov8l model in our project.

8.2.1 How YOLO Works :

- Single Pass Detection: Unlike traditional methods that apply a classifier to multiple regions of an image, YOLO divides the image into a grid and predicts bounding boxes and class probabilities for each grid cell in a single pass.
- Fast and Efficient: YOLO is designed to be fast and efficient, making it suitable for real-time applications.
- End-to-End Training: The entire model is trained end-to-end to predict both bounding boxes and class probabilities directly from the image pixels.
- High Accuracy: YOLO achieves high accuracy in detecting objects, even in complex scenes

- <u>CUDA Deep Neural Network (cuDNN):</u> It is a GPU-accelerated library developed by NVIDIA. It provides highly optimized implementations for standard routines used in deep neural networks, such as forward and backward convolution, attention, matrix multiplication (matmul), pooling, and normalization. cuDNNis widely used in deep learning frameworks to speed up training and inference, making it an essential tool for tasks like computer vision, conversational AI, and recommendation systems.We have to install nvidia toolkit and its latest driver version for utilizing the efficiency. We used cuda version 12.7 in the project.



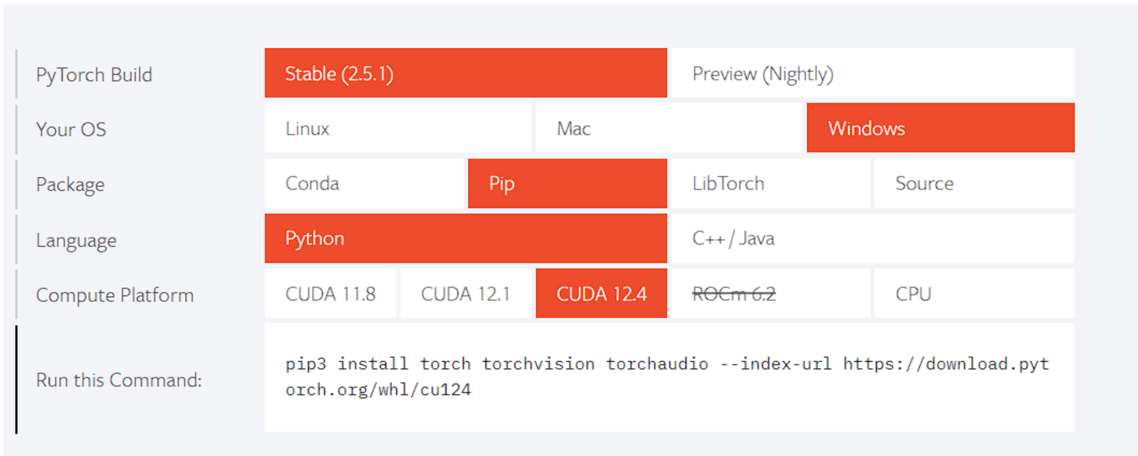
```
C:\WINDOWS\system32\cmd.
Microsoft Windows [Version 10.0.22635.4660]
(c) Microsoft Corporation. All rights reserved.

C:\Users\braja>nvidia-smi
Sat Jan 18 02:03:53 2025
+-----------------------------------------------------------------------------------------+
| NVIDIA-SMI 566.36                 Driver Version: 566.36         CUDA Version: 12.7     |
|-----------------------------------------+------------------------+----------------------+
| GPU  Name                  Driver-Model | Bus-Id          Disp.A | Volatile Uncorr. ECC |
| Fan  Temp   Perf          Pwr:Usage/Cap |           Memory-Usage | GPU-Util  Compute M. |
|                                         |                        |               MIG M. |
|=========================================+========================+======================|
|   0  NVIDIA GeForce RTX 3050 ...  WDDM  |   00000000:01:00.0 Off |                  N/A |
| N/A   40C    P8              6W /   85W |     377MiB /   4096MiB |      0%      Default |
|                                         |                        |                  N/A |
+-----------------------------------------+------------------------+----------------------+
```

Then

we have to go to the system environment settings and set the path as: C:\Program Files\NVIDIA GPU Computing Toolkit\CUDA\v12.6.

We have to swap the torch library with the cuda torch version as follows:



- <u>Hardware</u>: The system requires a camera (webcam, IP camera, etc.) to capture video footage in the industrial environment. A computer or server to processthe video feed and run the AI model, and optionally, agpu.

- <u>Libraries used:</u>

Package	**Current Version**	**Latest Version**
Jinja2	3.1.5	3.1.5
MarkupSafe	3.0.2	3.0.2
PyYAML	6.0.2	6.0.2
antlr4-python3-runtime	4.13.2	4.13.2
black	24.10.0	24.10.0
certifi	2024.12.14	2024.12.14
charset-normalizer	3.4.1	3.4.1
click	8.1.8	8.1.8
cmake	3.31.4	3.31.4
colorama	0.4.6	0.4.6
contourpy	1.3.1	1.3.1
cupy-cuda12x	13.3.0	13.3.0
cvzone	1.6.1	1.6.1
cycler	0.12.1	0.12.1
defusedxml	0.7.1	0.7.1

dlib-bin	19.24.6	19.24.6
docutils	0.16	0.21.2
fastrlock	0.8.3	0.8.3
filelock	3.16.1	3.16.1
filterpy	1.4.5	1.4.5
fonttools	4.55.3	4.55.3
fsspec	2024.12.0	2024.12.0
hydra-core	1.3.2	1.3.2
idna	3.10	3.10
imageio	2.36.1	2.36.1
imutils	0.5.4	0.5.4
jmespath	1.0.1	1.0.1
joblib	1.4.2	1.4.2
kiwisolver	1.4.8	1.4.8
lap	0.5.12	0.5.12
lazy_loader	0.4	0.4
llvmlite	0.43.0	0.43.0
matplotlib	3.10.0	3.10.0
mpmath	1.3.0	1.3.0
mypy-extensions	1.0.0	1.0.0
networkx	3.4.2	3.4.2
numba	0.60.0	0.60.0
numpy	2.0.2	2.2.1
omegaconf	2.3.0	2.3.0
opencv-python	4.10.0.84	4.11.0.86
packaging	24.2	24.2
pandas	2.2.3	2.2.3
pathspec	0.12.1	0.12.1
pillow	11.1.0	11.1.0
pip	24.3.1	24.3.1
platformdirs	4.3.6	4.3.6
psutil	6.1.1	6.1.1
py-cpuinfo	9.0.0	9.0.0
pyasn1	0.6.1	0.6.1
pyparsing	3.2.1	3.2.1

python-dateutil	2.9.0.post0	2.9.0.post0
pytz	2024.2	2024.2
requests	2.32.3	2.32.3
rsa	4.7.2	4.9
s3transfer	0.11.0	0.11.1
scikit-image	0.25.0	0.25.0
scikit-learn	1.6.1	1.6.1
scipy	1.15.1	1.15.1
seaborn	0.13.2	0.13.2
setuptools	75.8.0	75.8.0
six	1.17.0	1.17.0
spatial	0.2.0	0.2.0
sympy	1.13.1	1.13.3
threadpoolctl	3.5.0	3.5.0
tifffile	2024.12.12	2025.1.10
torch	2.5.1+cu124	2.5.1
torchaudio	2.5.1+cu124	2.5.1
torchvision	0.20.1+cu124	0.20.1
tqdm	4.67.1	4.67.1
typing_extensions	4.12.2	4.12.2
tzdata	2024.2	2024.2
ultralytics	8.3.59	8.3.63
ultralytics-thop	2.0.13	2.0.14
urllib3	2.3.0	2.3.0
wheel	0.45.1	0.45.1

- Structure of the Project :

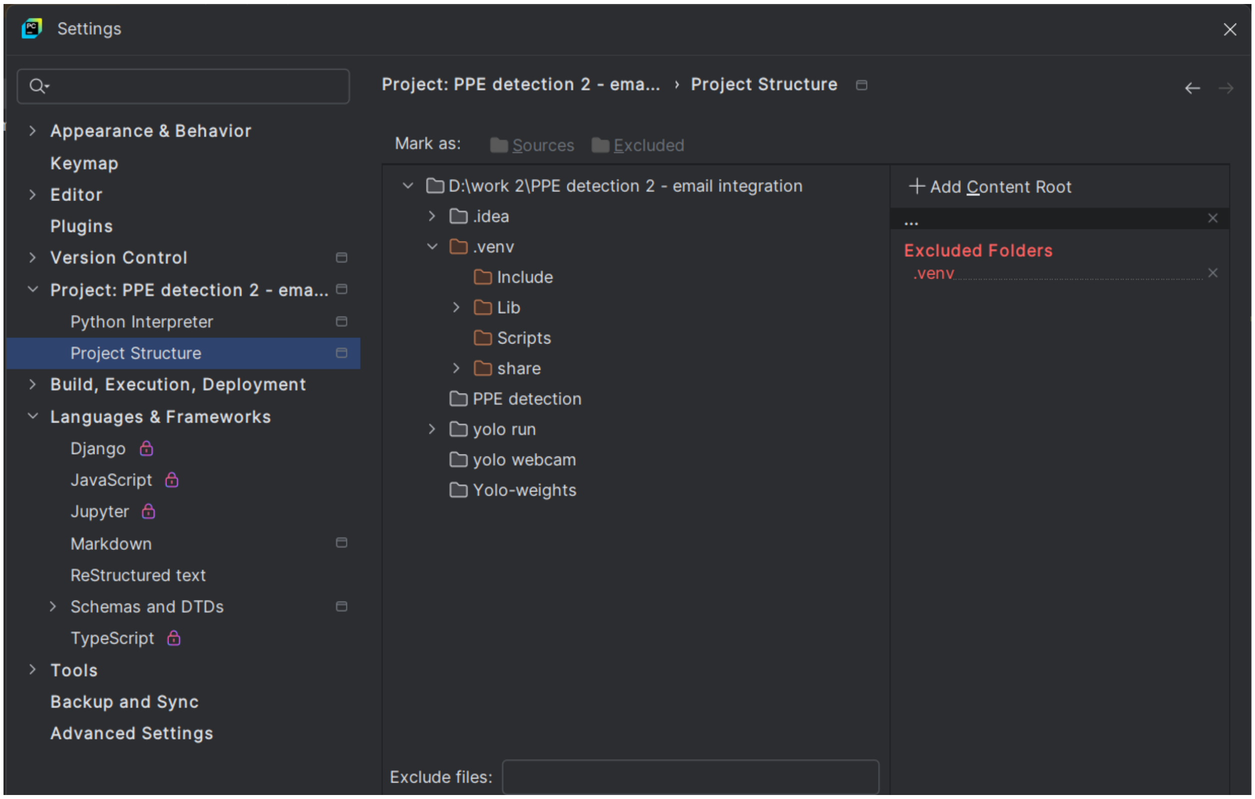


Fig.6.2: Structure of the project

9 Model Architecture :

The core of the system is a Convolutional Neural Network (CNN) architecture, which is particularly suited for tasks that involve image recognition. CNNs consist of the following layers:

Convolutional Layers: These layers apply filters to input images to extract essential features, such as edges and textures, which are used to detect whether a helmet is present.

Pooling Layers: These layers reduce the spatial dimensions of the feature maps, making the model more computationally efficient while retaining important features.

Fully Connected Layers: These layers process the extracted features and classify them into helmet or no-helmet categories.

In this project, we used YOLOv8 as a custom-trained model, fine-tuned on a dataset of workers in industrial settings to detect helmets. The model was trained to recognize helmets in various conditions, such as different angles, lighting conditions, and distances. Also we use Nvidia's CUDA dnn which is based on deep neural network for faster response.

10 Data Collection And Preproccesing

Data is a crucial component of any machine learning project. For this project, we collected a dataset of images consisting of workers in various environments, both with and without helmets. The dataset was then subjected to the following preprocessing steps:

Image Resizing: All images were resized to a fixed resolution to ensure uniformity and efficient processing by the model.

Data Augmentation: We applied augmentation techniques such as rotation, flipping, and varying brightness to create a more diverse dataset. This helps the model generalize better across different environments and avoid overfitting.

Normalization: Pixel values were normalized to a range between 0 and 1 to standardize the input, which improves the convergence of the model during training.

11 Model Training

The CNN model was trained using supervised learning, where each image in the dataset was labeled as either "helmet", "no helmet" or "vest", "No vest The training process was carried out as follows:

Training-Validation Split: The dataset was divided into two parts — 80% for training and 20% for validation. This split ensures that the model learns from the training set and is validated on unseen data during training.

Loss Function: We used a Binary Cross-Entropy Loss function to measure the error in helmet detection, optimizing it using the

Adam optimizer.

Batch Size and Epochs: A batch size of 32 and 100 epochs were used for training. Hyperparameters like the learning rate were fine-tuned to achieve optimal model performance.

During the training process, the model learns to extract key features such as the shape, color, and position of helmets from images and make predictions based on these features.

The model training was done on Google Colab for greater results. It uses Tesla T4 GPU to train the model.

12 Testing and Evaluation

After training the model, it was tested on a separate dataset that the model had not seen during training. The following metrics were used to evaluate its performance:

Accuracy: Measures the overall percentage of correct predictions made by the model.

Precision: Measures the percentage of true helmet detections out of all positive predictions made by the model.

Recall: Measures the percentage of helmets correctly detected out of all actual helmets in the test data.

F1 Score: A harmonic mean of precision and recall, used as a balanced metric for evaluating the model.

13 Workflow

The following diagram outlines the entire workflow of the Helmet Detection System:

1. Image Capture: Cameras are placed in strategic industrial locations to capture real-time video footage.

2. Image Preprocessing: The captured images are resized, normalized, and prepared for model input.

3. PPE Detection: The preprocessed images are passed through the trained CNN model. The model analyzes the images and classifies whether PPE is present.

4. Alert Generation: If a worker without a helmet is detected, the system generates an alert, notifying the safety personnel via gamail.

5. Action: Safety personnel can then intervene and take corrective actions to ensure compliance with helmet protocols.

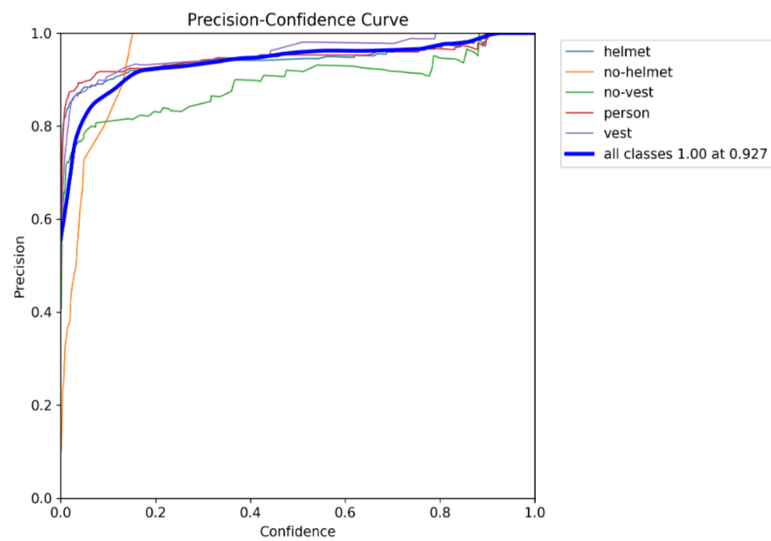
14 Performance Evaluation and Testing

The effectiveness of the AI-Powered PPE Detection System is evaluated through a series of tests designed to measure the model's accuracy, precision, recall, and robustness in various real-world conditions. These metrics help assess the system's ability to detect helmets accurately while minimizing false positives and negatives.

14.1 Precison-Confidence :

The image you uploaded is a Precision-Confidence Curve for different classes.

- X-Axis (Confidence): Ranging from 0.0 to 1.0.
- Y-Axis (Precision): Ranging from 0.0 to 1.0.



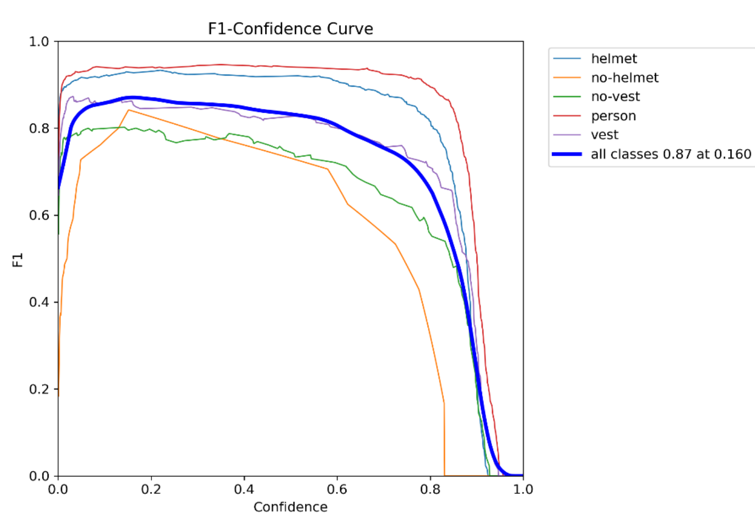
The curves represent the performance of different classes:

- Helmet (light blue)
- No-helmet (orange)
- No-vest (green)
- Person (red)
- Vest (purple)
- All classes combined (thick dark blue)

The legend indicates that the combined performance of all classes achieves a precision of 1.00 at a confidence level of 0.927. This type of curve is useful for evaluating the performance of classification models, particularly in terms of how confident the model is in its predictions and the corresponding precision.

14.2 F1 Score:

It plots the F1 score on the y-axis against the confidence on the x-axis for a classification model. The graph includes multiple coloured lines representing different classes:



- Helmet (light blue)
- No-helmet (orange)
- No-vest (green)
- Person (red)

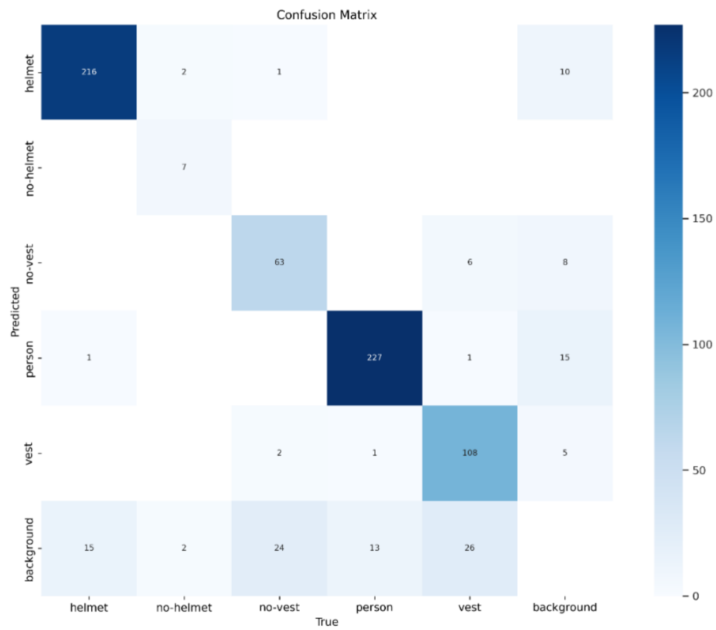
- Vest (purple)

Additionally, there's a thick blue line representing "all classes" with an F1 score of 0.87 at a confidence level of 0.160. The legend on the right side provides the color coding for each class.

This graph illustrates the model's performance across various confidence thresholds for different classes, helping us understand its accuracy and reliability.

14.3 Confusion Matrix:

It displays the performance of the model in predicting six different classes: helmet, no-helmet, no-vest, person, vest, and background.Here's a brief summary of the confusion matrix:



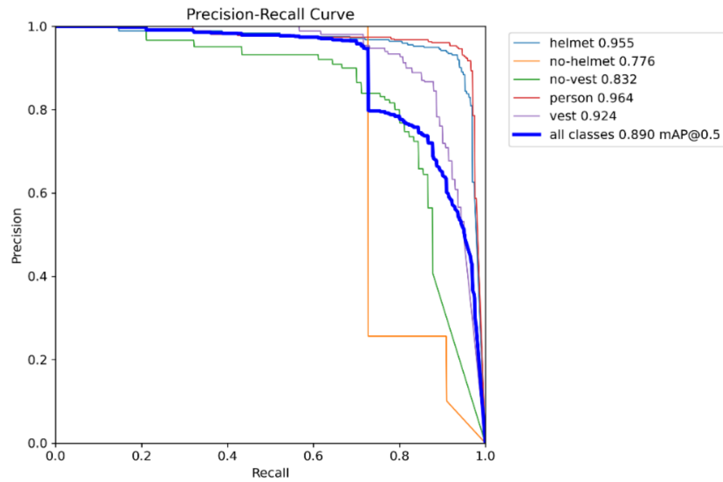
- Helmet: Correctly predicted as helmet 216 times, with 18 misclassifications.
- No-helmet: Correctly predicted as no-helmet 7 times, with 2 misclassifications.
- No-vest: Correctly predicted as no-vest 63 times, with 33 misclassifications.
- Person: Correctly predicted as person 227 times, with 32 misclassifications.
- Vest: Correctly predicted as vest 108 times, with 28 misclassifications.
- Background: Correctly predicted as background 26 times, with 20 misclassifications.

The diagonal elements represent the number of correct predictions, while the off-diagonal elements indicate the number of misclassifications.

Confusion matrices are particularly useful because they provide a detailed view of the model's performance, highlighting both its strengths and areas where it may need improvement.

14.4 Precision-Recall: The curve shows the trade-off between precision and recall for different classes in the model.

Here's a breakdown of the elements in the image:



- X-Axis (Recall): Ranging from 0.0 to 1.0.
- Y-Axis (Precision): Ranging from 0.0 to 1.0.

The legend on the right side of the image indicates the classes and their corresponding Average Precision (AP) scores:

- Helmet: 0.955
- No-helmet: 0.776
- No-vest: 0.832
- Person: 0.964
- Vest: 0.924
- All classes combined (mean Average Precision (mAP) @ 0.5): 0.890

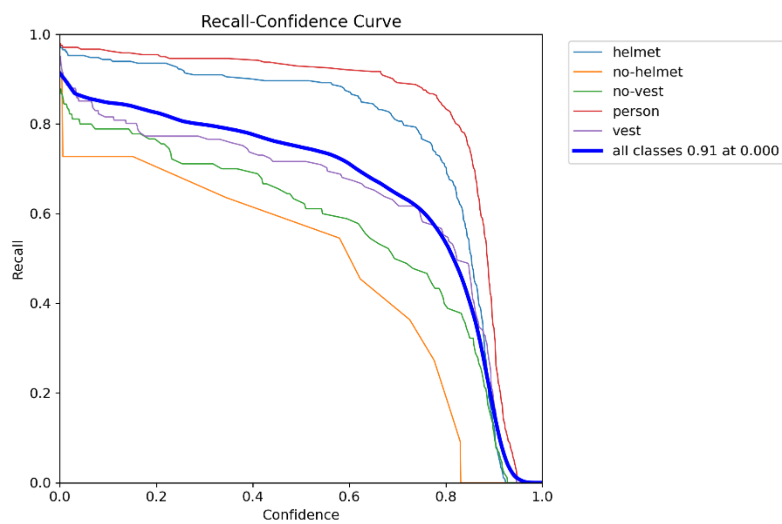
The thick blue line represents the combined performance across all classes, while the individual class performances are shown with different coloured lines.

This graph is useful for evaluating the performance of a multi-class.

classification model, particularly in understanding how well the model distinguishes between different classes.

14.5 Recall-Confidence: It plots recall on the y-axis against confidence on the x-axis for a classification model. The graph includes multiple colored lines representing different classes:

- Helmet (light blue)
- No-helmet (orange)
- No-vest (green)
- Person (red)
- Vest (purple)



Additionally, there's a thick dark blue line representing "all classes" with a recall of 0.91 at a confidence level of 0.000.

This graph illustrates how the model's recall varies with different confidence thresholds for each class, helping us understand its ability to correctly identify instances of each class.

15. Test Result :

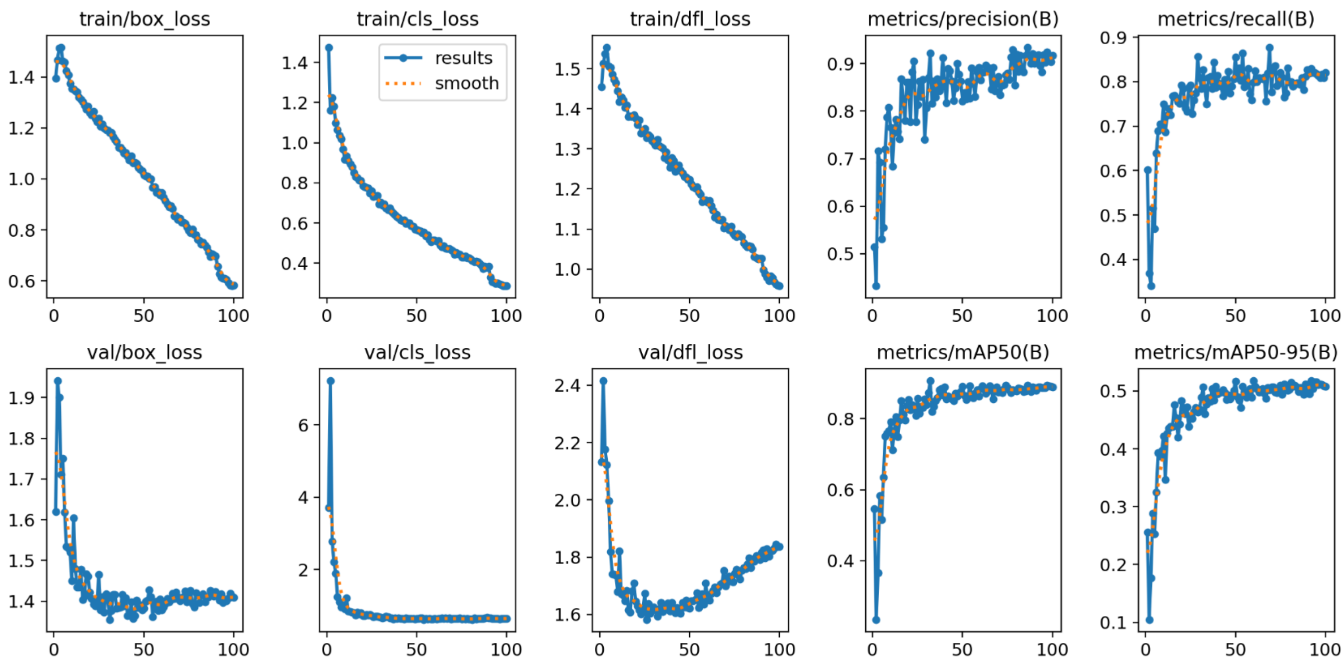


Fig. 15.1 : Training result's (Finetuning)

It isa series of line graphs showing the training and validation losses and metrics for a machine learning model over 100 epochs. Here's a brief overview of each graph:

1. train/box_loss: This graph shows the training box loss, starting at around 1.4 and decreasing steadily to around 0.6 over 100 epochs.
2. train/cls_loss: This graph shows the training classification loss, which starts at around 1.4 and decreases to around 0.4 over 100 epochs.
3. train/dfl_loss: This graph shows the training distribution focal loss, starting at around 1.5 and decreasing to around 1.0 over 100 epochs.
4. metrics/precision(B): This graph shows the precision metric, starting at around 0.5 and increasing to around 0.9 over 100 epochs.
5. val/box_loss: This graph shows the validation box loss, starting at around 1.9, decreasing sharply to around 1.4, and then fluctuating slightly over 100 epochs.
6. val/cls_loss: This graph shows the validation classification loss, which starts at around 6, decreases sharply to around 2, and then fluctuates slightly over 100 epochs.
7. val/dfl_loss: This graph shows the validation distribution focal loss, starting at around 2.4, decreasing sharply to around 1.6, and then fluctuating slightly over 100 epochs.
8. metrics/mAP50(B): This graph shows the mean Average Precision at 50% IoU, which starts at around 0.1

and increases to around 0.9 over 100 epochs.

9. metrics/mAP50-95(B): This graph shows the mean Average Precision at 50-95% IoU, starting at around 0.1 and increasing to around 0.5 over 100 epochs.

These graphs offer valuable insights into the model's performance and learning behavior during training and validation. The decreasing loss values and increasing metric values indicate that the model is learning and improving over time.

Based on the given graphs, the summary of the model's accuracy:

- **Precision**: It starts at around 0.5 and increases to around 0.9 over 100 epochs.

- **Mean Average Precision (mAP @ 50% IoU)**: Starts at around 0.1 and reaches approximately 0.9 over 100 epochs.

- **Mean Average Precision (mAP @ 50-95% IoU)**: Starts at around 0.1 and increases to around 0.5 over 100 epochs.

The mAP @ 50% IoU is a good measure of the model's accuracy and suggests that the model achieves around **90% accuracy** in detecting objects correctly by the end of the training process.

Fig. 15.2 : Picture showing the detected PPE equipments

Fig. 13.3 : Picture showing the detected PPE Helmet and Vest

Our testing plan is designed to verify that the proposed system not only meets its functional requirements but also performs efficiently in real-world construction site environments, enhancing safety and PPE compliance while respecting privacy and security constraints.

16. Error Analysis :

Despite the high accuracy, some errors were observed during the evaluation phase:

False Positives: In a few cases, the system mistakenly identified objects in the background (e.g., machinery or caps) as helmets. This was more common in environments with cluttered backgrounds or where objects had similar shapes to helmets.

False Negatives: In a few instances, helmets were not detected when the lighting was extremely poor or when the worker's head was tilted at a steep angle, obscuring the helmet from view.

To address these issues, further data augmentation (especially on edge cases) and fine-tuning of the model could help reduce the occurrence of these errors.

16.1. Comparison with Manual Monitoring :

The AI-Powered Safety Helmet Detection System was compared to manual monitoring, highlighting several advantages:

Efficiency: The system monitors helmet compliance 24/7 without the need for human intervention, making it more efficient than manual checks.

Accuracy: While human monitors are prone to fatigue and errors, especially in busy environments, the system consistently maintains high accuracy in detecting helmet usage.

Real-Time Alerts: The system sends instant alerts when it detects a violation, allowing for immediate action, whereas manual monitoring can suffer from delays.

16.2. Improvement and future Testing :

Based on the performance evaluation, the following improvements could be implemented:

Improved Data Collection: Gathering more data in challenging conditions, such as extreme lighting, unusual angles, and diverse backgrounds, can help improve the model's robustness.

Tuning Hyperparameters: Further fine-tuning of model hyperparameters, such as learning rate and batch size, could enhance its performance.

Ensemble Learning: Incorporating ensemble techniques or additional models might reduce false positives and negatives, making the system even more reliable.

17. Scope

The Safety Helmet Detection System has the potential for significant advancements and integrations with various technologies that can extend its capabilities beyond helmet detection. The system can evolve to incorporate more comprehensive safety measures, data management, and integration with other safety solutions. The following points highlight the expanded scope of this project:

17.1. Data Storage and Management via Website :

<u>Website Integration</u>: The system can be integrated with a web-based platform for centralized data storage and management. This website can store records of detected violations, compliance logs, and generate detailed reports. Such a platform would enable companies to track compliance over time, perform audits, and improve safety protocols based on data analytics.

Real-Time Dashboard: A live dashboard can be developed where safety officers can monitor multiple sites in real time, viewing helmet detection alerts and other safety metrics. This dashboard could be accessible remotely via a web interface, improving the system's accessibility and control.

Cloud Storage: Data from multiple sites can be securely stored on the cloud, allowing for scalability and easy access. The system could log timestamped incidents of safety violations, making it easier to conduct audits or provide evidence of compliance in case of an incident.

17.2. Integration with other Technologies :

Facial Recognition: The system can be integrated with facial recognition technology to track individuals' compliance with safety measures. This would allow for personalized safety tracking, ensuring that specific workers are adhering to regulations, especially in high-security or hazardous environments.

IoT and Sensor Integration: The system could be further developed to integrate with IoT devices and sensors that monitor environmental conditions such as temperature, humidity, or gas leaks.

18. Future Development

In the context of this project, there is a significant drawback where the system struggles to differentiate between a real person and a phot. This means that if someone presents a photo of a personal protective equipment (PPE) on a mobile device, the system might mistakenly detect it as the presence of actual PPE.

From a technical perspective, this issue arises due to limitations in the algorithm's ability to analize depth and motion. Current image recognition technology primarily relies on 2D image data, which can be easily deceived by high-quality photographs. The system lacks the capability to distinguish between flat, static images and dynamic , three-dimentional objects.

Solution : To address this issue, the development team could integrate additional layers of verification. One potential solution is to implement depth-sensing technology, such as **LiDAR** or **stereoscopic cameras**, which can measure the distance between the camera and the subject. This could enable the system to detect the presence of real, three-dimentional objects rather then flat images. Additionally, incorporating motion analysis algorithm can help differentiate between live, moving individuals and static photos. For instance, requiring a user to perform specific movements or gestures can provide an additional layer of validation.

Integrate depth-sensing technology (**LiDAR** or **stereoscopic cameras**) and **motion analysis algorithm** to improve the system's ability to detect real PPE and differentiate between photos and actual objects.

<u>Benefits</u> :

<u>Enhance Accuracy</u> : Depth-sensing technology can accurately measure distances, ensuring the system detects real, three dimentional objects.

<u>Improve Security</u> : Motion analysis can distinguish between live individuals and static photos, adding an extra layer of verification.

<u>Robust Solution</u> : Combining both technologies provides a comprehensive approach to accurately identify genuine PPE.

19. Conclusion

The Safety Helmet Detection System presents a significant advancement in workplace safety, particularly in high-risk industries where helmet use is mandatory. By automating the detection of safety helmets, this system minimizes the need for manual monitoring and ensures continuous compliance with safety protocols. It provides real-time alerts, enabling supervisors to take immediate action and prevent accidents. This system not only enhances safety but also helps organizations comply with regulatory requirements, reducing the risk of legal consequences and workplace injuries. The implementation of this technology leads to a safer, more efficient, and compliant work environment, where the protection of workers is prioritized.

The results of our tests have demonstrated the system’s high accuracy and reliability. However, like any technology, there are opportunities for improvement, such as reducing false detections and extending its capabilities to detect other forms of PPE. With future advancements, this system could become a core component of industrial safety monitoring worldwide, protecting workers and ensuring safe, efficient operations.

REFERENCES

1.Deng, J., et al. (2009). ImageNet Large Scale Visual Recognition Challenge. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 248- 255.

2. Redmon, J., & Farhadi, A. (2018). YOLOv3: An Incremental Improvement. arXiv preprint arXiv:1804.02767.

3. He, K., et al. (2016). Deep Residual Learning for Image Recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 770-778.

4. OpenCV: Open Source Computer Vision Library. (https://opencv.org/)

5. TensorFlow: An open-source machine learning framework. (https://www.tensorflow.org/)

6. PyTorch: An open source machine learning framework. (https://pytorch.org/)

7. LabelImg: An open source graphical image annotation tool. (https://github.com/tzutalin/labelImg)

8. GStreamer: Open-source multimedia framework. (https://gstreamer.freedesktop.org/)

9. FFmpeg: Multimedia framework for handling multimedia data. (https://www.ffmpeg.org/)

10.Twilio: Cloud communications platform for SMS alerts. (https://www.twilio.com/)

11.Iandola, F. N., et al. (2016). SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and