2. Student’s Full Name:Prakhar Sharma

3. Github Repo Link: <https://github.com/pra123134/pra121314-ML-Deeplearning/tree/main>

4. CRS Name: Artificial Intelligence

5. Course Name: ML AND DEEP LEARNING

6. School Name: Ryan Global School Kharghar

Submission Details (as per WACP AI Submission Criteria)

**Smarter Safety: Making PPE Monitoring Easier and More Reliable with AI**

**How We Built a Real-Time PPE Detection System Using YOLOv8 and Streamlit**

**Summary**

Keeping workers safe at construction and industrial sites is a serious responsibility. But even with rules and procedures in place, it's not always easy to make sure everyone is wearing the right protective gear—like hard hats, masks, and vests—at all times. Human supervisors are limited by what they can see, how much time they have, and how quickly they can act. Mistakes or missed violations can lead to dangerous situations.

In this project, we set out to solve this problem using artificial intelligence (AI) and computer vision. We built a system that can automatically detect whether workers are wearing the required safety gear in real time. It uses a model called YOLOv8 to spot different types of PPE in images and a tool called Streamlit to create an easy-to-use dashboard where safety officers can quickly see who’s following the rules—and who isn’t. Our goal was simple: to reduce the risk of accidents by giving site managers better tools to do their job.

**Why This Project Matters**

Construction zones are fast-moving and full of risks. Workers often move between tasks and areas, and supervisors can't be everywhere at once. Manually checking for PPE violations takes time and may not catch problems quickly enough to prevent harm. That’s where our system comes in.

We designed a smart solution that watches for compliance automatically. By using AI, we can give real-time alerts when something's wrong, helping teams respond faster and more effectively. The system isn’t just accurate—it’s also designed to be used by real people on the job, with a simple interface and clear visual feedback.

**Phase 1: Understanding the Problem and Preparing the Data**

**What We Learned from Real Construction Sites**

Before building anything, we needed to understand the real-world challenges. We talked to workers and supervisors and looked at how PPE compliance is currently monitored. Some common problems stood out:

* It's hard to keep track of every worker all the time.
* Manual checks can miss violations.
* There’s often a delay between noticing a problem and fixing it.

**Getting the Data Ready**

We used a labeled image dataset that shows workers wearing or not wearing PPE. Each image includes bounding boxes around objects like hard hats, masks, and vests. Here's how we got the data ready for our model:

* We resized images to a standard size (224x224 pixels).
* We normalized pixel values so the model could process them easily.
* We added some variety (like flipping and brightness changes) to make sure the model could handle real-world changes.
* We checked that each image had correct labels and a good mix of examples across different categories: compliant, partially compliant, and non-compliant.

**Exploring the Data**

Before training the model, we explored the dataset to understand what we were working with:

* We counted how often each PPE class appeared.
* We looked for class imbalances that might affect accuracy.
* We checked for errors in labeling and fixed anything suspicious.

**Phase 2: Building the Model and Creating the Interface**

**Training Our Detection Model**

We used YOLOv8 because it's fast and accurate, even on challenging images. The model was trained to spot:

* When PPE is present: hard hat, mask, vest
* When PPE is missing: no hard hat, no mask, no vest

To make it easier for supervisors to understand, we added a compliance color code:

* **Green:** Fully compliant
* **Yellow:** Missing one or two items
* **Red:** Not wearing any required gear

Our training process included tools to improve performance, like early stopping, label smoothing, and real-time augmentation.

**Measuring How Well It Worked**

We didn’t just train the model—we also tested it. We measured how well it could:

* Identify true violations (recall)
* Avoid false alarms (precision)
* Balance both (F1 score)
* Match predictions with actual objects (IoU)

We also visualized results to make them easier to understand:

* Side-by-side examples of model vs. ground truth
* Heatmaps showing what the model focused on
* Confusion matrices to analyze mistakes

**Making It Easy to Use with Streamlit**

We built a clean and interactive web dashboard using Streamlit. This lets supervisors use the system without needing technical skills. The dashboard includes:

* A button to upload images
* Real-time detection with color-coded boxes
* Compliance summaries (counts by color category)
* Graphs showing trends over time
* Alerts for urgent issues

**Keeping the System Updated**

Even a great model needs updates. To keep it reliable, we designed a plan for continuous improvement:

* Monitor for data drift or changes in job site conditions
* Add new training examples when needed
* Adjust thresholds as PPE rules change
* Gather user feedback to improve future versions

**How It All Fits Together**

Here’s a snapshot of our full development process:

**Start → Understand the Problem → Collect & Clean Data → Explore the Dataset → Train the Model → Test & Evaluate → Build the Dashboard → Deploy for Use → Monitor & Improve → Repeat**

### 📘 Storyboard Panels: PPE Monitoring System

#### Panel 1: The Problem

**Title: *Why PPE Monitoring Matters*** **Visual:** Construction site scene with workers, some with and some without PPE.  
 **Caption:** Manual safety checks miss violations. Our goal: detect PPE in real time using AI.

#### Panel 2: Project Vision

**Title: *AI to the Rescue* Visual:** Flowchart showing a camera feeding into a computer vision system with traffic light indicators (green/yellow/red).  
 **Caption:** We use YOLOv8 + Streamlit to create an intelligent PPE detection system.

#### Panel 3: Real-World Challenges

**Title: *Learning from the Field* Visual:** Interviews with workers, supervisors looking overwhelmed.  
 **Caption:** We studied job sites to understand gaps in current safety monitoring.

#### Panel 4: Data Preparation

**Title: *Building the Dataset* Visual:** Grid of labeled images with bounding boxes on PPE items.  
 **Caption:** We cleaned, resized, and augmented images for training. Categories: compliant, partially compliant, non-compliant.

#### Panel 5: Model Training

**Title: *Training YOLOv8* Visual:** Computer screen showing training progress (loss curves, images, detection boxes). **Caption:** The model learns to detect PPE items and classify worker compliance.

#### Panel 6: Evaluation Metrics

**Title: *How Well Does It Work?* Visual:** Graphs: precision-recall, confusion matrix, heatmap. **Caption:** We tested for precision, recall, IoU, and F1 score to validate accuracy.

#### Panel 7: User Interface

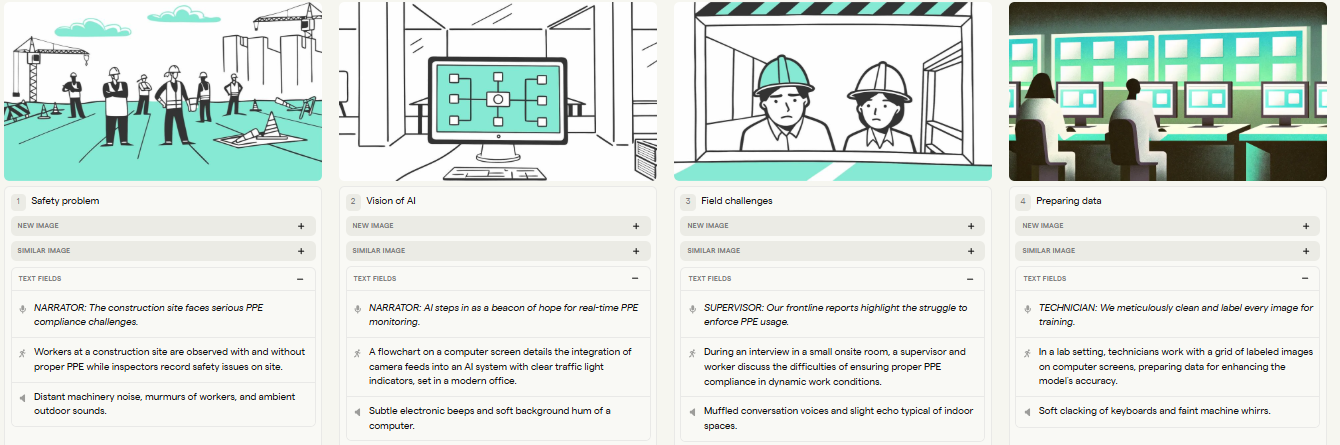
**Title: *Streamlit Dashboard* Visual: Screenshot of the dashboard with color-coded alerts and PPE status.  
 Caption: Supervisors get real-time feedback with an easy-to-use interface.**

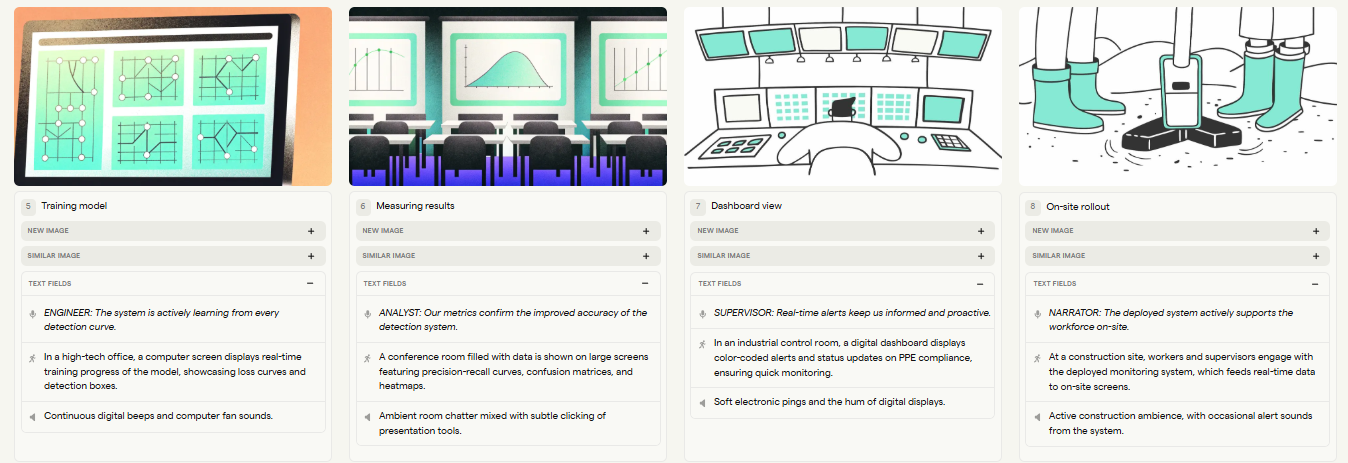
#### Panel 8: Deployment & Feedback

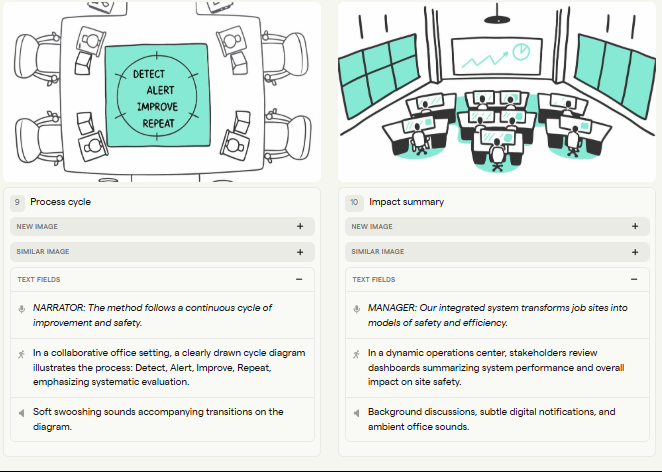
**Title: *Putting It to Work* Visual: Workers on-site with system in use; pop-up alerts on dashboard.  
 Caption: We deployed the system and planned for ongoing updates and user feedback.**

#### Panel 9: The Big Picture

**Title: *Safer Sites Through Smarter Systems* Visual: Cycle diagram showing “Detect → Alert → Improve → Repeat”  
 Caption: Continuous learning keeps the system reliable and ready for changing job sites.**

****

****

****