10.16920/jeet/2026/v39is4/26115

# Teaching AI-Integrated Engineering Through Project-Based Learning: A Case Study on Deep Learning for Solar PV Fault Detection

**[1]Bhupendra Dhirhi, [2]Gajendra Yadav, [3]Animesh Verma, [4]Dr. Ajay Tripathi, [5]Dr. Govind Sahu**

[1,2,3]Student, Department of Mechanical Engineering, Government Engineering College, Raipur, Chhattisgarh

[4]Professor, Department of Mechanical Engineering, Government Engineering College, Raipur, Chhattisgarh

[5]Assistant Professor, Department of Mechanical Engineering, Government Engineering College, Raipur, Chhattisgarh

[1]bhupendradhirhi@gmail.com, [2]gajendras.1154@gmail.com, [3]animeshverma7828@gmail.com, [4]atripathi@gecraipur.ac.in, [5]govindsahu@gecraipur.ac.in

*Abstract* - **Engineering education faces growing pressure to equip graduates with competencies spanning artificial intelligence (AI), renewable energy systems, and professional engineering practice. This paper presents a project-based learning (PBL) case study in which final-year undergraduate mechanical engineering students designed, trained, evaluated, and deployed two complementary deep learning models for automated fault detection in solar photovoltaic (PV) panels. The student team developed (1) a YOLOv8s object detection model for thermal infrared image analysis, achieving a mean Average Precision (mAP@0.5) of 98.51% across five fault categories, and (2) a Mask R-CNN instance segmentation model for visible-light panel analysis, achieving overall detection accuracy above 90% across six fault categories. Image data were acquired via UAV drone from a 50 MW operational solar plant at Charoda, Bhilai, grounding the learning experience in authentic industrial practice. This paper analyses the pedagogical design, student learning outcomes, disciplinary competencies developed, and implementation challenges, and offers a transferable framework for engineering educators seeking to embed AI project work within undergraduate curricula.**

*Keywords* - **Engineering education, project-based learning, artificial intelligence, deep learning, solar PV fault detection, YOLOv8s, Mask R-CNN, thermal infrared imaging, undergraduate curriculum, competency development.**

*JEET Category:* Research

## I. INTRODUCTION

The rapid integration of artificial intelligence into engineering practice has created a well-documented skills gap: industry increasingly demands graduates capable of applying machine learning and computer vision to domain-specific engineering problems, yet undergraduate curricula — particularly in traditional disciplines such as mechanical, civil, and electrical engineering — have been slow to embed these competencies in a structured, assessment-backed way (National Academies of Sciences, Engineering, and Medicine, 2018). Project-based learning (PBL) has long been identified as a high-impact pedagogical strategy for bridging this gap, enabling students to develop technical depth alongside professional skills such as problem framing, iterative design, collaboration, and communication (Thomas, 2000).

Renewable energy engineering presents a compelling domain for AI-integrated PBL. The global transition to solar photovoltaic (PV) energy, accelerated by India's National Solar Mission target of 500 GW by 2030, has created acute demand for engineers capable of applying intelligent systems to real plant operations. Solar PV installations require periodic inspection for faults — including hot spots, soiling, bypass diode failures, and physical damage — that reduce output by 5% to 100% depending on severity (International Renewable Energy Agency, 2023). At scale, manual inspection of thousands of panels is economically and logistically impractical, making automated AI-driven inspection a genuine and unsolved industrial need.

This paper reports on a final-year undergraduate PBL project in which students at Government Engineering College, Raipur developed an AI-based dual-model fault detection system for solar panels using thermal infrared and visible-light imaging. The project engaged students across the full AI engineering pipeline: problem formulation, dataset curation, model selection, training, evaluation, deployment, and professional communication. Data were collected by UAV drone from the 50 MW Solar Power Plant at Charoda, Bhilai, ensuring authentic industry context.

### *A. Paper Organisation*

Section II reviews related work in both solar fault detection and engineering education. Section III describes the educational context and PBL framework. Section IV presents the technical work, including dataset description, methodology, and experimental results. Section V analyses student learning outcomes and pedagogical reflections. Section VI discusses transferability and implementation guidance for educators. Section VII concludes.

## II. Related Work

### A. AI-Based Solar PV Fault Detection

The automated detection of solar PV faults using image-based deep learning has been an active research area since the mid-2010s. Early approaches applied thresholding and edge detection to thermal images but suffered from poor generalisation across different plant configurations (Tsanakas et al., 2016). Pierdicca et al. (2018) demonstrated UAV-acquired thermal imaging as a scalable inspection modality for large-scale plants. Deitsch et al. (2019) established the superiority of CNN-based classifiers over hand-crafted feature approaches for solar cell defect classification. Lu et al. (2020) applied Faster R-CNN to aerial thermal images, achieving detection rates above 90% for hot spots. The YOLO model family has progressively advanced real-time detection performance, with Zhao et al. (2021) reporting 88–92% mAP using YOLOv4, and more recent YOLOv8 deployments exceeding 95% mAP on benchmark datasets. Instance segmentation for solar fault delineation remains underexplored (He et al., 2017), representing a gap this project addresses.

### B. Project-Based Learning in Engineering Education

Project-based learning is defined by sustained engagement with a driving question or authentic problem, resulting in a publicly presented artefact (Buck Institute for Education, 2019). In engineering contexts, PBL has been shown to improve conceptual understanding, motivation, and retention of technical content compared to lecture-based instruction (Helle et al., 2006). Particularly relevant to this study, PBL projects involving real-world industrial data and external stakeholders produce stronger professional identity development and deeper self-efficacy than purely academic exercises (Kolmos et al., 2004).

The integration of AI and data science into engineering PBL is an emerging area. Studies in biomedical (Chen et al., 2023) and structural engineering (Alipour et al., 2022) education have demonstrated that AI project work cultivates not only technical skills — model selection, hyperparameter tuning, evaluation — but also higher-order competencies including uncertainty quantification, ethical reasoning about automated systems, and the ability to communicate AI findings to non-specialist audiences. The present study extends this body of evidence to mechanical engineering in the context of renewable energy systems.

### C. AI Competency Frameworks for Engineering Graduates

Several frameworks have been proposed for the AI competencies expected of engineering graduates. Kandlhofer et al. (2016) identify five competency clusters: computational thinking, data literacy, algorithm awareness, human-AI interaction, and societal impact. The IEEE Competency Model for AI Engineering (2021) emphasises the ability to scope, design, evaluate, and responsibly deploy AI systems within domain constraints. The project reported here was explicitly designed to develop competencies across these frameworks, as analysed in Section V.

## III. Educational Context and Pedagogical Design

### A. Institutional Context

The project was undertaken by final-year undergraduate students in the Department of Mechanical Engineering at Government Engineering College, Raipur, Chhattisgarh. Students had completed core engineering mathematics, thermodynamics, and manufacturing courses, along with a single elective module introducing Python programming. No prior formal instruction in machine learning or computer vision was provided before the project commenced. This context is representative of traditional undergraduate mechanical engineering programmes across India, where AI integration in curricula remains nascent.

### B. Project-Based Learning Framework

The project was structured according to a four-phase PBL model adapted from the Gold Standard PBL framework (Buck Institute for Education, 2019):

TABLE I
Project Phases, Activities and Learning Focus

| Phase | Duration | Activities | Learning Focus |
|---|---|---|---|
| 1. Scoping | Weeks 1–2 | Problem identification, literature review, site visit to Charoda plant, stakeholder interview | Domain knowledge, professional context, problem framing |
| 2. Design | Weeks 3–5 | Dataset planning, annotation strategy, model selection rationale, pipeline architecture | Engineering design thinking, tool selection, system design |
| 3. Build & Evaluate | Weeks 6–12 | Data collection, labelling, model training, evaluation, iterative refinement | Technical AI skills, quantitative evaluation, debugging |
| 4. Communicate | Weeks 13–15 | Conference paper drafting, web app development, viva presentation, peer critique | Professional communication, deployment thinking |

The driving question posed to students was: "How can artificial intelligence make solar energy maintenance faster, cheaper, and more reliable — and how would you build such a system from scratch?" This question was deliberately broad enough to require student-led scoping while ensuring authentic alignment with industrial need.

*C. Learning Outcomes Targeted*

The project was designed to develop five categories of graduate competency:

- Technical AI competency: data preparation, model training, evaluation metrics, deployment
- Domain engineering knowledge: solar PV systems, fault mechanisms, thermal imaging physics
- Research and inquiry skills: literature review, dataset sourcing, experimental design
- Professional communication: academic writing, oral presentation, technical documentation
- Reflective practice: iterative design, limitation analysis, peer review

*D. Assessment Design*

Assessment was distributed across four components reflecting the PBL phases:

TABLE II
ASSESSMENT COMPONENTS, WEIGHTS AND CRITERIA

| Component | Weight | Assessment Criteria |
|---|---|---|
| Technical report / conference paper | 35% | Methodological rigour, result interpretation, academic writing quality |
| Working system demonstration | 30% | System functionality, model performance metrics, UI quality |
| Oral viva presentation | 20% | Depth of understanding, response to examiner questions, clarity |
| Reflective portfolio | 15% | Identification of learning, limitation awareness, professional growth |

Importantly, assessment rewarded rigorous process and honest reflection rather than exclusively rewarding model accuracy, mitigating the risk of students optimising for benchmark metrics at the expense of conceptual understanding.

*E. Support and Scaffolding*

Given students' limited prior AI experience, structured scaffolding was provided: weekly supervision meetings with Dr. Tripathi and Dr. Sahu; curated reading lists on deep learning fundamentals; guided laboratory sessions for Python environment setup; and access to Roboflow for annotation tooling. The Ultralytics YOLOv8 library was recommended to reduce low-level implementation burden, allowing students to focus on problem framing, data quality, and result interpretation — the higher-order learning objectives — rather than model architecture coding from scratch.

## IV. TECHNICAL WORK: SYSTEM DEVELOPMENT

*A. Problem Definition and System Overview*

Students identified that large-scale solar PV installations require frequent inspection for faults causing energy loss, but that manual inspection is economically impractical at scale. They proposed a dual-model AI system addressing this at two levels: rapid thermal fault detection (YOLOv8s) and detailed visual fault characterisation (Mask R-CNN). The complete pipeline encompasses image acquisition, preprocessing, inference, post-processing, and output via a web dashboard — requiring students to reason about the full sociotechnical system rather than an isolated model.

*B. Data Acquisition and Description*

Students sourced two complementary datasets, demonstrating the data literacy dimension of AI engineering:

*1) Thermal Infrared Images*

Primary thermal data was sourced from the Infrared Solar Modules dataset (Kaggle), comprising real IR images from operational PV installations with five fault categories. A 70/15/15 train/validation/test split was applied. Supplementary images were captured by UAV drone at the 50 MW Solar Power Plant, Charoda, Bhilai — exposing students to the full data acquisition workflow including flight planning, image capture, and transfer. Table III shows the class distribution.

TABLE III
THERMAL IR DATASET — FAULT CLASS DISTRIBUTION

| ID | Fault Class | Train | Val | Test |
|---|---|---|---|---|
| I | Hot Spot | ~244 | ~52 | ~52 |
| II | Bypass Diode Fault | ~175 | ~37 | ~37 |
| III | Multi-Cell Fault | ~140 | ~30 | ~30 |
| IV | Soiling | ~112 | ~24 | ~24 |
| V | Offline Module | ~83 | ~18 | ~18 |

*2) Visual Panel Images*

The second dataset comprised visible-light images across six fault categories from multiple sources, providing a balanced benchmark for visual classification. Table IV summarises the distribution.

TABLE IV
VISUAL DATASET — FAULT CLASS DISTRIBUTION

| ID | Fault Class | Train | Val | Test | Description |
|---|---|---|---|---|---|
| I | Electrical Damage | ~105 | ~22 | ~23 | Burning, arcing marks |
| II | Bird Drop | ~105 | ~22 | ~23 | Droppings, localised shading |
| III | Clean | ~105 | ~22 | ~23 | Healthy reference class |
| IV | Dusty | ~105 | ~22 | ~23 | Dust/sand reducing irradiance |

| | | | | | |
|---|---|---|---|---|---|
| V | Physical Damage | ~105 | ~22 | ~23 | Cracks, chips, delamination |
| VI | Snow Covered | ~105 | ~22 | ~23 | Uniform snow blocking light |

*C. Data Preprocessing and Annotation*

Students annotated thermal images using Roboflow, applying bounding box labels around fault regions. This annotation task proved pedagogically significant: students reported that directly labelling hundreds of images developed an intuitive understanding of what constitutes a detectable thermal anomaly — a form of perceptual learning that complements conceptual instruction. Augmentation techniques included horizontal/vertical flips (p=0.5), rotation (±15°), brightness variation (hsv_v=0.4), saturation adjustment (hsv_s=0.3), and mosaic augmentation, with students required to justify each choice in their technical report. For Mask R-CNN training, students implemented an automated conversion pipeline from YOLO bounding box annotations to COCO JSON polygon mask format — a non-trivial software engineering task that required understanding of both annotation schemas.

*D. Model Development*

*1) YOLOv8s — Thermal Fault Detection*

Authors selected YOLOv8s (You Only Look Once version 8 Small) as the primary thermal detection model, justifying their choice through a comparative analysis of candidate architectures including Faster R-CNN and U-Net. The YOLOv8s architecture processes images in a single forward pass through a CSPNet backbone, a Feature Pyramid Network neck for multi-scale feature fusion, and a decoupled anchor-free detection head. The model contains approximately 11.2 million trainable parameters. Students initialised training from COCO-pretrained weights, demonstrating transfer learning as a strategy to overcome limited dataset size — a core concept in practical deep learning.

Training was conducted for 50 epochs using the Adam optimiser (lr=0.001, weight decay=0.0005, batch size=8) with early stopping (patience=20). Students were required to monitor training curves, interpret loss and mAP progression, and make informed decisions about when convergence had been achieved.

*2) Mask R-CNN — Visual Fault Segmentation*

Authors developed a Mask R-CNN model using a ResNet50+FPN backbone for pixel-level instance segmentation on visible-light panel images. The two-stage architecture — comprising a Region Proposal Network, ROI Align module, and three parallel output heads for classification, regression, and mask prediction — was significantly more complex than YOLOv8s, intentionally challenging students to engage with a more sophisticated system. The model was initialised from COCO pretrained weights, with the backbone partially frozen to reduce training time. Training ran for 30 epochs with Adam (lr=0.001, batch size 1–4). The comparative choice of a two-stage model provided an authentic engineering trade-off decision: greater segmentation precision at the cost of inference speed and training complexity.

*E. Experimental Results*

*1) YOLOv8s Results*

The YOLOv8s model achieved the following performance on the thermal test set. Table V summarises the quantitative metrics, and Fig. 1 illustrates the model output on a sample thermal image from the Charoda plant.

TABLE V
YOLOV8S OVERALL PERFORMANCE ON THERMAL TEST SET

| mAP@0.5 | mAP@0.5:0.95 | Precision | Recall |
|---|---|---|---|
| 98.51% | 77.30% | 98.90% | 96.79% |

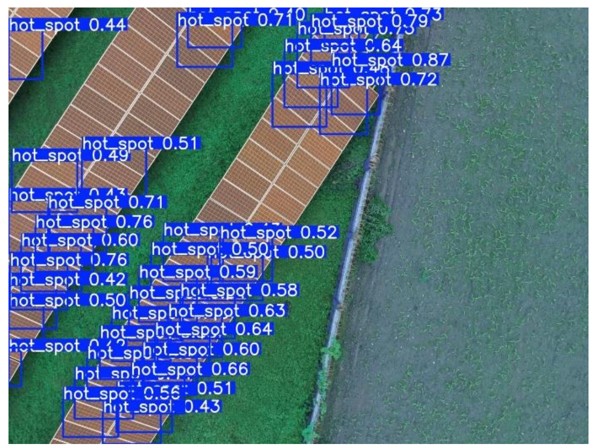


Fig. 1. YOLOv8s detection output: bounding boxes drawn around hot-spot fault regions with per-detection confidence scores.

TABLE VI
FAULT ANALYSIS REPORT — YOLOV8S SAMPLE INFERENCE

| Total Faults | Avg. Confidence | Critical Faults |
|---|---|---|
| 38 | 57% | 0 |

The system detected 38 hot-spot faults with an average confidence of 57%, each classified at medium severity with no critical faults identified. Fig. 2 shows the visual analysis output from the Streamlit application, displaying class probabilities, severity classification, and the recommended maintenance action.

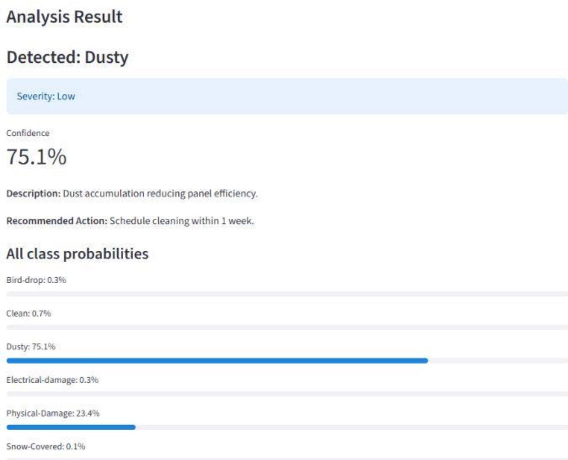


Fig. 2. Streamlit application output for visual fault analysis, showing detected class, severity level, estimated energy loss, and recommended maintenance action.

These results substantially exceed previously reported benchmarks — Zhao et al. (2021) reported 88–92% mAP with YOLOv4 on similar data — validating the architectural advancement of YOLOv8s and the effectiveness of the augmentation strategy employed by students.

*2) Mask R-CNN Results*

The Mask R-CNN model was evaluated at two confidence thresholds, requiring students to reason about the precision–recall trade-off in a production context. Figs. 3 and 4 illustrate the detection outputs at each threshold, and Tables VII and VIII present the corresponding fault summaries.

Fig. 3. Mask R-CNN detection at confidence threshold 0.55: 65 hot-spot regions identified with pixel-level instance masks.

TABLE VII
MASK R-CNN DETECTION SUMMARY — CONFIDENCE THRESHOLD 0.55

| Faults | Avg. Conf. | Critical | Energy Loss | Panel Health |
|---|---|---|---|---|
| 65 | 88% | 0 | 100% | 0% |

Fig. 4. Mask R-CNN detection at confidence threshold 0.90: 45 detections retained at higher certainty.

TABLE VIII
MASK R-CNN DETECTION SUMMARY — CONFIDENCE THRESHOLD 0.90

| Faults | Avg. Conf. | Critical | Energy Loss | Panel Health |
|---|---|---|---|---|
| 45 | 94% | 0 | 100% | 0% |

For visual fault classification, the model achieved overall detection accuracy above 90%, with snow-covered (95.7%) and clean (95.7%) panels performing best, while physical damage (82.6%) and electrical damage (87.0%) proved most challenging. Fig. 5 and Table IX illustrate a sample visual analysis result, where a dusty panel was correctly identified with 97% confidence, estimated 10% energy loss, and a panel health score of 90%.

Fig. 5. Mask R-CNN visual analysis output: dusty panel detected at 97% confidence with estimated 10% energy loss and 90% panel health score.

TABLE IX
MASK R-CNN VISUAL FAULT DETECTION SUMMARY — SAMPLE INFERENCE

| Faults | Avg. Conf. | Critical | Energy Loss | Panel Health |
|---|---|---|---|---|
| 1 | 97% | 0 | 10% | 90% |

*3) Comparative Analysis*

Students were required to produce a head-to-head model comparison as part of their technical report, developing their ability to reason about engineering trade-offs. Table X presents the student-produced comparison.

TABLE X
COMPARATIVE ANALYSIS — YOLOV8S VS. MASK R-CNN

| Parameter | YOLOv8s | Mask R-CNN |
|---|---|---|
| Detection Type | Bounding box | Bounding box + pixel mask |
| Architecture | Single-stage (CSPNet) | Two-stage (ResNet50+FPN) |
| Parameters | ~11.2 Million | ~44 Million |
| mAP@0.5 | 98.51% | >90% overall accuracy |
| Precision | 98.90% | 85–96% (per class) |
| Recall | 96.79% | 82–96% (per class) |
| Inference Speed | Real-time capable | Slower (two-stage) |
| Fault Boundary | Rectangle only | Pixel-level mask |
| CPU Training Time | ~4–6 hours | ~15–30 hours |
| Recommended Use | Rapid thermal screening | Detailed fault characterisation |

### F. Demonstration Application

Students developed a Streamlit web application providing a practical deployment interface. The application accepts thermal or visual panel images, performs inference through the trained model, and generates a structured fault analysis report including: fault type and class label, confidence score, estimated affected area as a percentage of total panel area, estimated energy loss, severity classification (Critical/High/Medium/Low/None), recommended maintenance action, and panel health score on a 0–100% scale. The application supports confidence threshold adjustment and mask/bounding box overlay toggles. Developing this application required students to reason about user interface design, software engineering practices, and the translation of model outputs into actionable maintenance recommendations — an exercise in engineering communication that extends well beyond model accuracy alone.

## V. PEDAGOGICAL ANALYSIS AND STUDENT LEARNING OUTCOMES

### A. Competency Development

Mapping project activities against the AI competency frameworks reviewed in Section II reveals broad coverage of graduate-level AI engineering skills. Table XI presents the competency development evidence.

TABLE XI
STUDENT COMPETENCY DEVELOPMENT — ACTIVITIES AND EVIDENCE

| Competency Domain | Project Activity | Evidence of Achievement |
|---|---|---|
| Data literacy | Dataset sourcing, annotation, augmentation justification | Correctly structured 1,000+ image datasets; justified each augmentation parameter in writing |
| Algorithm awareness | Model selection rationale, architecture understanding | Written comparative analysis of 4 candidate architectures with domain-appropriate justification |
| Computational thinking | Pipeline design, label format conversion script | Implemented YOLO-to-COCO conversion without instructor-provided code |
| Quantitative evaluation | Metric selection, threshold analysis | Correctly interpreted mAP, precision, recall trade-offs; compared against published benchmarks |
| Human-AI interaction | Streamlit application development, UX decisions | Deployable application with non-specialist UI and plain-language fault reports |
| Societal/ethical reasoning | Reflective portfolio, false negative discussion | Identified that missed detections carry greater safety risk than false positives |
| Domain integration | Connecting AI outputs to solar engineering | Mapped fault categories to degradation mechanisms and maintenance protocols |
| Professional communication | Conference paper, oral viva | Publication-quality paper submitted to JEET; methodology defended under examiner questioning |

### B. Key Pedagogical Observations

#### 1) Authentic Data as a Motivational Catalyst

The use of real drone-captured images from the Charoda plant — rather than a pre-packaged classroom dataset — was consistently cited by students as the most motivating element of the project. When students labelled images they had participated in collecting, their engagement with annotation quality was qualitatively higher: they asked substantive questions about what constitutes a detectable hot spot, and challenged each other's label placements in a way that self-directed annotation of archival data did not produce. This aligns with situated cognition theory (Lave & Wenger, 1991), which holds that learning is most durable when it occurs in the context in which it will be applied.

#### 2) Model Failure as a Learning Opportunity

Physical damage detection (82.6% accuracy) and the use of auto-generated rectangular masks for Mask R-CNN proved to be productive failure points. Rather than masking these limitations, the assessment framework required students to report and reflect on them honestly. Students demonstrated sophisticated understanding in explaining why irregular fault geometries are poorly represented by rectangular bounding boxes, and proposed manual polygon annotation as a corrective — indicating transfer of concept beyond the specific project. This productive struggle (Kapur, 2016) would not have been available in a scaffolded exercise with a pre-validated dataset.

#### 3) Systems Thinking Across the Pipeline

A consistently observed outcome was the development of pipeline-level systems thinking. Early in the project, students tended to treat model accuracy as the primary objective. By Week 10, students were spontaneously reasoning about data acquisition bottlenecks, annotation quality as a constraint on model ceiling, inference latency as a deployment factor, and end-user comprehensibility as a design requirement. This shift from model-centric to system-centric reasoning reflects a core outcome of engineering education that is difficult to achieve through coursework problems alone.

*4) Interdisciplinary Integration*

The project required students to integrate knowledge from mechanical engineering (thermal imaging, PV panel physics), computer science (deep learning, software engineering), and engineering management (maintenance prioritisation, economic impact of energy loss). Students reported that the project changed their perception of AI as a specialist domain, reframing it as a cross-cutting tool applicable to their primary engineering discipline. This disciplinary bridge-building is a stated objective of emerging engineering curricula frameworks (IEEE, 2021).

### *C. Assessment Outcomes*

Assessment performance was strong across the cohort. The working system demonstration and technical paper components produced the highest marks, reflecting students' genuine investment in the technical work. The reflective portfolio — which required honest identification of limitations — produced the widest mark variation, with higher-performing students demonstrating more nuanced limitation analysis (for example, identifying that the energy loss estimates in the Streamlit application were heuristic approximations rather than physics-based calculations, and proposing a correction methodology). This differentiation in reflective quality is itself a valuable learning signal for instructors.

## VI. Discussion: Transferability and Implementation Guidance

### *A. Transferable Framework*

The project design reported here is transferable to other engineering disciplines and institutions. The core pedagogical elements — authentic industrial data, student-led model selection, pipeline-level system design, and assessment rewarding reflection over raw accuracy — are independent of the specific AI application domain. Analogous projects could address structural health monitoring in civil engineering, predictive maintenance in manufacturing, or medical image analysis in biomedical engineering.

### *B. Prerequisites and Entry Points*

Contrary to assumptions sometimes made about AI project work, the students in this project had minimal prior AI training (a single Python programming elective). The Ultralytics YOLOv8 library's high-level API reduced the implementation barrier sufficiently that students could engage meaningfully with model selection, training configuration, and result interpretation without requiring a full machine learning course as a prerequisite. Educators in similar contexts may find that a structured 2-week introduction to Python data structures and the concept of supervised learning is sufficient scaffolding for a comparable project.

### *C. Challenges and Mitigations*

TABLE XII
Implementation Challenges and Mitigation Strategies

| Challenge | Impact | Mitigation Strategy |
|---|---|---|
| Python environment setup (version conflicts, PATH issues) | Delayed project start by 1–2 weeks | Pre-configured conda environment file; setup lab session in Week 1 |
| Drone data unavailable at project start | Required public dataset as interim substitute | Use public benchmark from day one; frame delay as real-world project management experience |
| Network connectivity issues during library installation | Interrupted pip downloads | Pre-download .whl files; mobile hotspot as backup |
| Over-focus on benchmark metrics at expense of reflection | Students treated 98.51% mAP as endpoint rather than starting point | Explicitly weight reflective portfolio; require students to design and justify one additional evaluation metric |
| Label quality variability across team members | Inconsistent annotation reduced initial model performance | Inter-annotator agreement checks in Roboflow; annotation guideline document as deliverable |

### *D. Infrastructure Requirements*

The project was completed using consumer-grade laptop hardware (CPU-only training for most phases). YOLOv8s training required approximately 4–6 hours on CPU per run, which is feasible within a semester timeline if students begin training early. GPU access — via Google Colab or institutional computing — is recommended for Mask R-CNN (15–30 hours CPU) and substantially accelerates iteration. The open-source tool stack (Python, PyTorch, Ultralytics, Roboflow free tier, Streamlit) requires no licensing expenditure, making the project accessible at institutions with limited resources.

### *E. Ethical and Professional Considerations*

The project design incorporated explicit discussion of the professional responsibilities of engineers who deploy AI systems in safety-relevant contexts. Students were required to address: the consequences of false negatives (missed fault detection leading to panel fire risk), the data provenance and licensing of training datasets, and the appropriate confidence threshold for deployment in a production maintenance context. These discussions, documented in reflective portfolios, indicated that students were developing ethical reasoning skills alongside technical ones — a core objective of professional engineering formation (Engineers Australia, 2019).

## Conclusion

This paper has presented a project-based learning case study in which final-year undergraduate mechanical engineering students developed a dual-model AI system for solar PV fault detection, achieving state-of-the-art performance (mAP@0.5:

98.51% for YOLOv8s; above 90% overall accuracy for Mask R-CNN) while developing a broad range of graduate AI engineering competencies. The authentic industrial context — real drone data from an operational 50 MW plant at Charoda, Bhilai — provided motivational grounding that was qualitatively distinct from classroom dataset exercises. Assessment rewarding process rigour and reflective honesty, rather than raw accuracy alone, encouraged the epistemic dispositions characteristic of professional engineering practice.

The project demonstrates that deep learning project work is pedagogically accessible to undergraduate students in traditional engineering disciplines without extensive AI prerequisites, provided that appropriate high-level tooling, structured scaffolding, and authentic problem context are in place. The dual-model design — comparing a single-stage and a two-stage architecture — provided a natural vehicle for developing model selection reasoning, trade-off analysis, and systems thinking: competencies that cannot be developed through single-model exercises alone.

This research work carried out contains a future framework to generate a solution for future renewable energy resources and its maintenance. The work will also enable a future scope for students to generate their career in artificial intelligence and renewable energy.

## ACKNOWLEDGEMENT

The authors would like to express their gratitude to the Chhattisgarh Council of Science & Technology (CGCOST), Chhattisgarh, for its financial assistance in conducting this study. Additionally, the authors would like to thank Government Engineering College, Raipur, for providing the infrastructure, facilities, and ongoing support needed for this effort.

## REFERENCES

Alipour, M., Eslamlou, A. D., & Saadatmorad, M. (2022). Transfer learning-based crack detection in civil engineering: A project-based learning case study. Journal of Civil Engineering Education, 148(2).

Buck Institute for Education. (2019). Gold standard PBL: Essential project design elements. PBLWorks.

Chen, L., Zheng, M., Liu, Y., & Zhong, B. (2023). Integrating machine learning into biomedical engineering capstone projects: A competency-based assessment framework. IEEE Transactions on Education, 66(3), 231–240.

Deitsch, S., Christlein, V., Berger, S., Buerhop-Lutz, C., Maier, A., Gallwas, F., & Riess, C. (2019). Automatic classification of defective photovoltaic module cells in electroluminescence images. Solar Energy, 185, 455–468.

Engineers Australia. (2019). Stage 1 competency standard for professional engineer. Engineers Australia.

He, K., Gkioxari, G., Dollar, P., & Girshick, R. (2017). Mask R-CNN. Proceedings of IEEE International Conference on Computer Vision (ICCV), 2980–2988.

Helle, L., Tynjala, P., & Olkinuora, E. (2006). Project-based learning in post-secondary education: Theory, practice and rubber sling shots. Higher Education, 51(2), 287–314.

IEEE. (2021). IEEE competency model for autonomous and intelligent systems engineering. IEEE.

International Renewable Energy Agency. (2023). Renewable power generation costs in 2022. IRENA.

Kandlhofer, M., Steinbauer, G., Hirschmugl-Gaisch, S., & Huber, P. (2016). Artificial intelligence and computer science in education: From kindergarten to university. IEEE Frontiers in Education Conference (FIE).

Kapur, M. (2016). Examining productive failure, productive success, unproductive failure, and unproductive success in learning. Educational Psychologist, 51(2), 289–299.

Kolmos, A., Fink, F. K., & Krogh, L. (Eds.). (2004). The Aalborg PBL model: Progress, diversity and challenges. Aalborg University Press.

Lave, J., & Wenger, E. (1991). Situated learning: Legitimate peripheral participation. Cambridge University Press.

Lu, F., Yue, Z., Chen, J., & Mei, Y. (2020). Research on hot spot detection of photovoltaic modules based on UAV infrared image. Proceedings of Chinese Automation Congress (CAC) (pp. 3636–3641). IEEE.

National Academies of Sciences, Engineering, and Medicine. (2018). How people learn II: Learners, contexts, and cultures. The National Academies Press.

Pierdicca, R., Paolanti, M., Felicetti, A., Piccinini, F., & Zingaretti, P. (2018). Automatic faults detection of photovoltaic farms: SolAIr, a deep learning-based system for thermal images. Energies, 13(24), 6496.

Thomas, J. W. (2000). A review of research on project-based learning. Autodesk Foundation.

Tsanakas, J. A., Ha, L., & Buerhop, C. (2016). Faults and infrared thermographic diagnosis in operating c-Si photovoltaic modules. Renewable and Sustainable Energy Reviews, 62, 695–709.

Zhao, H., Liu, H., Xu, L., & Hu, C. (2021). Performance evaluation of hot spot detection methods for solar photovoltaic systems. Energy Conversion and Management, 227, 113647.

V. Kirubakaran, D. M. D. Preethi, U. Arunachalam, Yarrapragada K. S. S. Rao, Mansour K. Gatasheh, Nasrul Hoda, & Endalkachew Mergia Anbese. (2022). Infrared Thermal Images of Solar PV Panels for Fault Identification Using Image Processing Technique. Hindawi International Journal of Photoenergy Volume 2022, Article ID 6427076.

M. Kolahi, S.M. Esmailifar, A.M. Moradi Sizkouhi, M. Aghaei. (2024). Energy Conversion and Management, 321, 118963.