PROJECT ABSTRACT

Wind Turbine Blade Damage Detection using CNN

A built-to-order final-year project that classifies wind turbine blade inspection photographs into five damage categories with a fine-tuned EfficientNet-B0 CNN, with curated dataset, training notebook, interactive demo and complete documentation.

01. Title

Wind Turbine Blade Damage Detection using CNN

02. Introduction / Background

Wind turbine blades are the most damage-prone part of a turbine: rain and particle erosion wear the leading edge, cyclic loads open surface cracks, lightning strikes puncture laminates, and subsurface delamination spreads silently. Inspection is done by rope-access technicians or drones, producing thousands of photographs that must be reviewed by eye. A convolutional neural network trained to recognise damage categories turns this manual review into a fast, repeatable screening step, flagging the blades that need a closer look.

03. Problem Statement

Manual review of blade inspection photographs is slow, subjective and inconsistent across inspectors, so early-stage defects are missed and maintenance is scheduled on calendar time rather than actual blade condition. There is no widely available labelled dataset or ready reference implementation for the five most common blade damage categories, which forces every student team to start from zero.

04. Objectives

- Curate and annotate ~2,400 blade inspection photographs into five classes: healthy, crack, leading-edge erosion, lightning damage, delamination.
- Fine-tune an EfficientNet-B0 CNN with augmentation for outdoor lighting variation.
- Evaluate honestly with per-class precision/recall and a confusion matrix on a held-out split.
- Deliver an interactive web demo that runs the damage-classification pipeline on sample photos.
- Document architecture, training choices and results for the report, PPT and viva.

05. Proposed Methodology

Transfer learning is used: EfficientNet-B0 pre-trained on ImageNet provides the visual backbone, and a new classification head (global average pooling, dropout, five-way softmax) is trained on the blade dataset. Images are resized to 224×224 and normalised; augmentation (rotation ±15°, brightness jitter, horizontal flip) simulates drone-photography conditions. Training uses Adam with cosine-decay learning rate and early stopping on validation loss, with a stratified 80/20 train/validation split.

06. System Architecture / Working

- Input: blade inspection photograph (drone or rope-access).

- Preprocessing: resize 224×224, normalise, optional augmentation at train time.
- Model: EfficientNet-B0 backbone → global average pooling → dropout (0.3) → dense softmax (5 classes).
- Output: damage class with probability distribution; low-confidence cases flagged for human review.
- Demo: single-file web app reproducing the inference pipeline on sample photos.

07. Hardware / Software Requirements

- Hardware: any laptop/PC for training on a free cloud GPU; CPU-only inference (~35 ms/image expected).
- Software: Python 3, TensorFlow/Keras, NumPy, Matplotlib, scikit-learn, Jupyter Notebook, Pillow.
- Demo: any modern web browser; the single-file app runs offline.

08. Expected Outcomes

- A trained five-class blade-damage classifier with design-target validation accuracy ~93% (reported honestly after the training run).
- Per-class metrics and confusion matrix identifying the hardest defect pairs.
- Interactive demo, full training notebook, report PDF, PPT and viva Q&A.;

09. Applications

- Screening step for drone-based wind farm inspection workflows.
- Training aid for inspection technicians learning defect categories.
- Baseline for student research on renewable-energy computer vision.

10. Future Scope

- Localisation of defects with bounding boxes (detection) instead of whole-image labels.
- Severity scoring per defect to prioritise maintenance.
- Video-frame aggregation across a full blade pass.