PV_Array_Final.pdf

Birla Institute of Technology, Mesra

Document Details

Submission ID

trn:oid:::3117:598734120

Submission Date

Jun 9, 2026, 9:08 PM GMT+5:30

Download Date

Jun 9, 2026, 9:26 PM GMT+5:30

File Name

PV_Array_Final.pdf

File Size

981.9 KB

60 Pages

16,881 Words

95,882 Characters

5% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.

Filtered from the Report

- Bibliography
- Small Matches (less than 14 words)

Match Groups

25 Not Cited or Quoted 4%
Matches with neither in-text citation nor quotation marks

0 Missing Quotations 0%
Matches that are still very similar to source material

4 Missing Citation 1%
Matches that have quotation marks, but no in-text citation

0 Cited and Quoted 0%
Matches with in-text citation present, but no quotation marks

Top Sources

5%	Internet sources
2%	Publications
0%	Submitted works (Student Papers)

Integrity Flags

2 Integrity Flags for Review

Replaced Characters
23 suspect characters on 7 pages
Letters are swapped with similar characters from another alphabet.

Hidden Text
3 suspect characters on 1 page
Text is altered to blend into the white background of the document.

Our system's algorithms look deeply at a document for any inconsistencies that would set it apart from a normal submission. If we notice something strange, we flag it for you to review.

A Flag is not necessarily an indicator of a problem. However, we'd recommend you focus your attention there for further review.

Match Groups

25 Not Cited or Quoted 4%
Matches with neither in-text citation nor quotation marks

0 Missing Quotations 0%
Matches that are still very similar to source material

4 Missing Citation 1%
Matches that have quotation marks, but no in-text citation

0 Cited and Quoted 0%
Matches with in-text citation present, but no quotation marks

Top Sources

5%	Internet sources
2%	Publications
0%	Submitted works (Student Papers)

Top Sources

The sources with the highest number of matches within the submission. Overlapping sources will not be displayed.

1	Internet	**dspace.dtu.ac.in:8080**	**2%**
2	Internet	**qiita.com**	**<1%**
3	Internet	**deepnote.com**	**<1%**
4	Internet	**pythonguides.com**	**<1%**
5	Internet	**nipunbatra.github.io**	**<1%**
6	Internet	**dspace.auk.edu.kw**	**<1%**
7	Internet	**www.dspace.dtu.ac.in:8080**	**<1%**
8	Internet	**app.studyraid.com**	**<1%**
9	Internet	**catalog.iyte.edu.tr**	**<1%**
10	Internet	**discuss.pytorch.org**	**<1%**

11	Internet	theses.hal.science	<1%
12	Internet	doc.hfai.high-flyer.cn	<1%
13	Internet	export.arxiv.org	<1%
14	Internet	www-dev.fit.vutbr.cz	<1%
15	Internet	arccjournals.com	<1%
16	Internet	knowledgecommons.lakeheadu.ca	<1%
17	Internet	theses.eurasip.org	<1%
18	Internet	webthesis.biblio.polito.it	<1%
19	Internet	www.mdpi.com	<1%
20	Internet	mydata.ch	<1%
21	Publication	Patrick T. Komiske, Eric M. Metodiev, Jesse Thaler. "Energy flow polynomials: a co...	<1%
22	Internet	archive.org	<1%
23	Internet	islandscholar.ca	<1%
24	Internet	snowch.github.io	<1%

M.TECH (Control & Instrumentation)

MEGHA ARYA

2026

Cost-Optimized Uncertainty-Based Decision Framework For Automated Photovoltaic Defect Detection Using Monte Carlo Dropout

A DISSERTATION

SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE AWARD OF THE DEGREE
OF
MASTER OF TECHNOLOGY
IN
CONTROL & INSTRUMENTATION

Submitted by

MEGHA ARYA
(2K24/C&I/20)

Under the supervision of

Prof. RACHNA GARG
Prof. M. RIZWAN
Department of Electrical Engineering

DEPARTMENT OF ELECTRICAL ENGINEERING
DELHI TECHNOLOGICAL UNIVERSITY
(Formerly Delhi College of Engineering)
Bawana Road, Delhi 110042

MAY, 2026

DEPARTMENT OF ELECTRICAL ENGINEERING
DELHI TECHNOLOGICAL UNIVERSITY
(Formerly Delhi College of Engineering)
Bawana Road, Delhi-110042

CANDIDATE'S DECLARATION

I, Megha Arya, Roll No. 2K24/C&I/20, student of M.Tech (Control & Instrumentation), hereby declare that the project Dissertation titled "Cost-Optimized Uncertainity-Based Decision Framework for Automated Photovoltaic Defect Detection Using Monte Carlo Dropout" which is submitted by me to the Department of Electrical Engineering, Delhi Technological University, Delhi in partial fulfillment of the requirement for the award of degree of Master of Technology, is original and not copied from any source without proper citation. This work has not previously formed the basis for the award of any Degree, Diploma, Associateship, Fellowship or other similar title or recognition.

Place: Delhi

Megha Arya
(2K24/C&I/20)

Date:

7

DEPARTMENT OF ELECTRICAL ENGINEERING

DELHI TECHNOLOGICAL UNIVERSITY
(Formerly Delhi College of Engineering)
Bawana Road, Delhi-110042

CERTIFICATE

I hereby certify that the Project Dissertation titled "Cost-Optimized Uncertainity-Based Decision Framework for Automated Photovoltaic Defect Detection Using Monte Carlo Dropout" which is submitted by Megha Arya, Roll No. 2K24/C&I/20, Department of Electrical Engineering, Delhi Technological University, Delhi in partial fulfillment of the requirement for the award of the degree of Master of Technology, is a record of the project work carried out by the student under my supervision. To the best of my knowledge this work has not been submitted in part or full for any Degree or Diploma to this University or elsewhere.

1

Place: Delhi

Date:

Prof. Rachna Garg
SUPERVISOR
Department of Electrical Engineering
Delhi Technological University

1

Prof. M. Rizwan
SUPERVISOR
Department of Electrical Engineering
Delhi Technological University

DEPARTMENT OF ELECTRICAL ENGINEERING
DELHI TECHNOLOGICAL UNIVERSITY
(Formerly Delhi College of Engineering)
Bawana Road, Delhi-110042

ACKNOWLEDGEMENT

I would like to thank Prof. Rachana Garg, the Head of Department, Department of Electrical Engineering, for their continuous support and for providing a conducive environment for research. Their leadership and encouragement have been crucial in facilitating the progress of my work.

I take immense pleasure in expressing my deep and sincere gratitude to my esteemed supervisor, Prof. Rachana Garg and Prof. M. Rizwan , for invaluable guidance and for dedicating precious hours to my work. Their excellent cooperation, constructive suggestions, and stimulating discussions provided me with the impetus to work rigorously and made the completion of this thesis possible.

1 I am grateful to the Department of Electrical Engineering, Delhi Technological University, for providing access to computational resources and for maintaining a researchoriented academic environment that enabled this work.

I appreciate and heartily thank all the faculty members of the Department of Electrical Engineering and all my colleagues for their constant support and encouragement throughout the research. Finally, yet importantly, I would like to pay high regards 23 to my parents, family members, and friends for their inspiration, motivation, and patience throughout the duration of this work. Their unwavering support has been the foundation upon which this thesis rests.

Place: Delhi

Megha Arya
(2K24/C&I/20)

Date:

Abstract

Solar photovoltaic (PV) energy systems are increasingly contributing to the expectations of global renewable energy infrastructure. However, to ensure their long-term operational reliability requires accurate and timely detection of faults and operational defects of manufacturing. Deep learning models such as convolutional neural networks (CNNs), have showcased high classification performance on electroluminescence (EL) of PV cells. However, a crucial gap still remains: these models generate only point predictions without any degree of confidence or reliability, which makes them unreliable for high-stakes real-world deployment.

The purpose of this thesis is to describe a two-stage research pipeline designed to fill this void. During stage one, a baseline CNN model was created to classify defects in EL images using a binary scheme on the standardized ELPV benchmark data set containing 2,624 grayscale EL images. The accuracy rate obtained by this baseline CNN model was 72.84%, but the primary drawback lies in its failure to provide any measure of uncertainty, thereby making it unreliable for autonomous solar panel maintenance tasks.

In the second phase, this limitation is directly addressed through a novel cost-optimized, uncertainty-aware decision framework. Monte Carlo (MC) Dropout is applied to the same CNN architecture — enabling the same dropout layers used for regularization during training to serve as a Bayesian approximation mechanism during inference. By performing 100 stochastic forward passes per input, a per-sample uncertainty estimate is computed. A two-zone decision framework then routes predictions: high-confidence samples are accepted automatically, while uncertain samples are forwarded to low-cost manual expert review.

The proposed framework is evaluated against a realistic operational cost model incorporating false positive costs (\$100), false negative costs (\$800), and manual inspection costs (\$20). Experimental results on the ELPV dataset demonstrate an

88.3% reduction in total operational cost compared to full automation (\$63,200 → \$7,400), while achieving 100% accuracy on all automated decisions and eliminating all false positives and false negatives in the automated stream. Furthermore, MC Dropout uncertainty is confirmed as a statistically reliable error indicator, with incorrect predictions exhibiting 27.2% higher uncertainty than correct ones ($p < 0.001$).

This work demonstrates that uncertainty quantification, when integrated with cost-aware decision logic, transforms a limited baseline classifier into a practically deployable, economically efficient inspection system for large-scale solar farms.

Keywords: photovoltaic defect detection, convolutional neural network, Monte Carlo Dropout, uncertainty quantification, cost optimization, electroluminescence imaging, deep learning.

Contents

List of Tables

List of Figures

List of Symbols and Abbreviations

PV	Photovoltaic
EL	Electroluminescence
CNN	Convolutional Neural Network
MC	Monte Carlo
UQ	Uncertainty Quantification
ELPV	Electroluminescence Photovoltaic (dataset)
FP	False Positive
FN	False Negative
τ	Uncertainty threshold
$u(x)$	Predictive uncertainty of sample x
$\bar{p}(y\|x)$	Mean predictive probability
T	Number of Monte Carlo forward passes ($T = 100$)
C_{FP}	Cost of a false positive (\$100)
C_{FN}	Cost of a false negative (\$800)
C_M	Cost of manual inspection (\$20)
ReLU	Rectified Linear Unit
Adam	Adaptive Moment Estimation (optimizer)
BN	Batch Normalization
H	Height of input image (pixels)
W	Width of input image (pixels)

Chapter 1

INTRODUCTION

1.1 Overview

The global energy landscape is undergoing a fundamental change driven by the pressing need to lessen the reliance on fossil fuels and lessen the rapid consequences of climate change. Solar photovoltaic (PV) energy has emerged as one of the most affordable, scalable, cost-effective, and broadly deployable renewable energy technologies available. From residential rooftop systems to utility-scale solar farms spanning thousands of hectares, PV installations have become a key component of the worldwide energy transition. According to the International Energy Agency, solar PV has been responsible for the largest share of new energy capacity additions worldwide in recent years, with cumulative installed capacity surpassing one terawatt. The trend of the developments is expected to persist because of low prices of modules and increased need for renewable sources of energy.

Nonetheless, as the size and scope of the geographical distribution of PV installations become larger, the complexity of operations to keep the plants working optimally increases accordingly. PV modules are supposed to have an average service life of 25-30 years; however, there are both defects arising from manufacturing processes and stresses occurring during the operation of the modules, leading to several difficulties that affect their efficiency negatively. Proper detection and handling of the flaws is essential.

1.2 Solar Photovoltaic Technology

1.2.1 Principles of PV Energy Generation

A photovoltaic (PV) cell is a semiconductor device that directly converts sun light directly into electrical energy through the photovoltaic effect. When photons of sufficient energy hit the semiconductor material, which is typically silicon, electrons get excited across the bandgap and the electron-hole pairs are generated. An internal electric field created by the p-n junction within the cell separates these charge carriers and drives a current through an external circuit.

Individual PV cells produce modest voltages and currents. In actuality, PV cells are made up of cells interconnected in series and parallel configurations to form PV modules, and multiple modules are combined into arrays to provide the power output required for utility-scale generation. The efficiency of a PV cells defined as the ratio of electrical output power to incoming solar irradiance, is a crucial performance indicator, with commercial silicon cells typically achieving efficiencies in the range of 18% to 24%.

1.2.2 Types of PV Cells

PV technologies are broadly categorized based on the semiconductor materials and fabrication processes involved:

Monocrystalline silicon cells are produced from a single continuous silicon crystal, providing great efficiency and long operational lifetimes. Their homogeneous crystal structure reduces recombination losses, making them the leading technology in high-performance applications.

Polycrystalline (multicrystalline) silicon cells are produced from ingots that have many crystal grains in them. Despite being slightly less efficient than monocrystalline cells because of grain boundary recombination, they are less expensive to manufacture and remain commonly used across large solar farms.

Thinfilm cells: In these solar cells, semiconductors like CdTe, CIGS, or amorphous silicon are layered on a substrate. They provide some advantages in terms of flexibility and low-light conditions; however, their efficiency is generally lower than that of crystalline silicon cells.

The ELPV dataset used in this study contains electroluminescence images from both monocrystalline and polycrystalline modules, making it a representative benchmark for actual PV inspection situations.

1.2.3 Defect Types and Their Consequences

PV cells are exposed to a variety of defects, each with unique causes, visual signatures, and operational repercussions:

Microcracks are tiny fractures in silicon cells brought on by mechanical stress during production, transportation, installation, or field thermal cycling in the field. Microcracks can disrupt current flow routes, reduce the active cell area, and spreading over time under continuous thermal and mechanical stress, ultimately leading to cell failure.

Hotspots are isolated regions of high temperature caused by shadowing, soiling, or electrical imbalance between cells. Hotspots accelerate localised deterioration and in extreme cases represent a fire hazard. Early identification of emerging hotspots is essential for preventing future catastrophic failures.

Delamination refers to the separating the encapsulant layer from the glass cover or the backsheet, allowing moisture to enter. Moisture accelerates the degradation of metallic interconnects and silicon surfaces, significantly reducing the module lifetime.

Cell degradation and discolouration result from prolonged exposure to the ultraviolet radiation, high temperatures, and humidity, causing slow chemical breakdown of the encapsulant and a decrease in carrier mobility inside the silicon.

Broken interconnects are malfunctions and failures in the metallic busbars or finger contacts that gather current from the cell surface, decreasing the area that effectively collects current and increases the series resistance.

Research and studies have demonstrated that even minor reductions in individual module efficiency, compounded across thousands of modules in a large solar farm, translate to substantial losses in energy generation and revenue over the course of a big solar farm's lifetime [1, 2, 3]. Therefore, Early and accurate defect detection is an economic necessity, not merely a maintenance convenience.

1.3 Electroluminescence Imaging for PV Inspection

1.3.1 Principles of EL Imaging

Electroluminescence imaging has proposed itself as the gold standard for non-destructive, high-resolution examination of PV cells. The technique exploits the radiative recombination of injected minority charge carriers within silicon. When a PV module is forward biased by applying an external current, minority carriers are injected into the semiconductor material. In regions of high crystalline quality, these carriers recombine radiatively, emitting near-infrared photons. Defective regions, where non-radiative recombination at crystal defects, microcracks, or broken contacts dominates, appear as locally dark areas in the resulting EL image.

EL imaging provides a spatially resolved map of the recombination activity within a PV module, enabling the detection of a wide variety of defects that are entirely invisible under visible light inspection. The technique is non-destructive, requires no special sample preparation, and can be performed on operational modules with portable EL imaging equipment.

1.3.2 Sample EL Images from the ELPV Dataset

Figure 1.1 shows three representative EL images from the ELPV dataset used in this thesis. The dark diagonal lines visible in these images are microcracks — a characteristic signature of mechanical fracture within the silicon cell. The automated detection system developed in this work primarily targets these features, notwithstanding their subtlety.

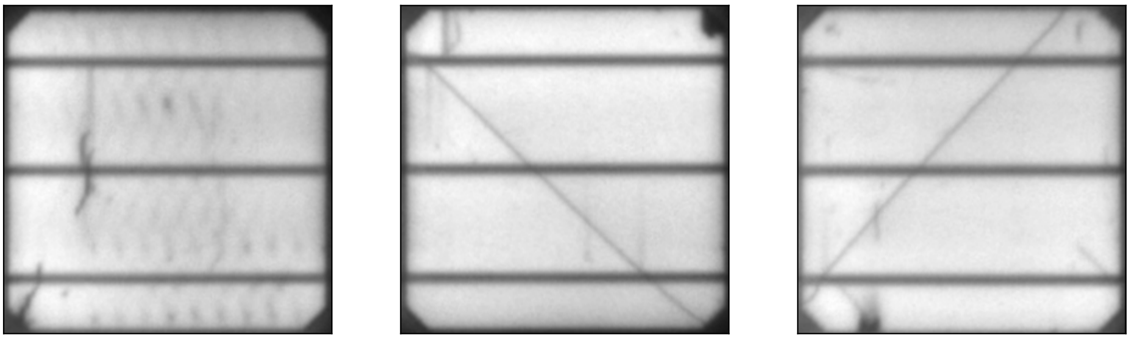

Figure 1.1: Representative EL images from the ELPV dataset showing defective PV cells. Dark diagonal lines indicate microcracks within the silicon cell material.

1.4 Problem Statement

Traditional PV inspection which are based on manual inspection and analysis of EL images is labor-intensive, time-consuming, and cannot meet the demands of modern large-scale solar installations which comprises of hundreds of thousands of modules. The fast proliferation of solar deployments has made automated defect detection systems using deep learning algorithms like CNNs an interesting topic for scientific research [4, 5, 6, 7, 8, 9].

They showed good performance in classifying benchmarks. Nonetheless, a fundamental problem exists regarding the practical usability of the proposed algorithms. Current automated defect detection methods remain black boxes, providing only point estimates of defect classification without any indication of their confidence and reliability. This can be considered a critical problem when it comes to real-world applications in solar farms. The operator has no way to differentiate whether the automated system has made a reliable and confident estimate or not.

False-positive results will mean extra trips to maintainers costing several hundred dollars each. On the other hand, false-negative predictions lead to defects being overlooked and may cause more significant problems in the future like expensive urgent repairs, safety issues, and losses in power generation due to malfunctioning panels. Automated systems require a way to indicate whether they are confident about their predictions or not.

The lack of a proper calibrated uncertainty indicator is the principal problem blocking practical deployment of CNN-based solar panel inspection systems, which will be considered in this thesis.

1.5 Research Objectives

This thesis addresses the above gap using a two-phase progressive research approach which is given below:

1. **Phase I – Development of Baseline CNN Model:** Build, train, and evaluate a CNN model using the ELPV design to classify binary PV defect cases. This establishes a working baseline and provides a systematic description of the problems that arise from the use of point prediction approaches in real-world operational costs.
2. **Phase II - Cost-Optimized Framework Accounting for Predictive Uncertainty:** Enhance the CNN model of Phase I by adding Monte Carlo Dropout for quantifying uncertainty in the predictions. Design a novel two-zone cost-optimized framework that utilizes uncertainty estimation for classifying cases either through automated acceptance or through low-cost expert analysis while achieving 100% accuracy in the automated zone.

1.6 Scope and Contributions

The primary contributions of this thesis are:

- A working CNN baseline for PV defect detection on the ELPV dataset, with documented training dynamics, accuracy metrics, and a detailed characterization of the operational cost implications of confidence-free classification.
- Integration of Monte Carlo Dropout uncertainty quantification into the PV inspection pipeline, with statistical validation of the uncertainty-error correlation (+27.2% higher uncertainty for incorrect predictions, $p < 0.001$).
- A novel cost-optimized two-zone decision framework that achieves an 88.3% reduction in total operational cost compared to full automation, while guaranteeing 100% accuracy on automated decisions.
- A demonstration that economic decision-making driven by uncertainty signals provides a principled and practical pathway to deploying deep learning models in high-stakes industrial inspection tasks.
- A comprehensive experimental evaluation demonstrating that the proposed framework is practically viable for real-world solar farm deployment, offering significant reductions in operational cost while maintaining full accuracy on automated decisions.

1.7 Solar Energy in India

India is reaching towards a critical turning point in its energy transformation. The country plans to achieving 500 GW of renewable energy capacity by 2030, with solar PV forming the majority component of this goal. Under the National Solar Mission and subsequent policy frameworks, India has rapidly expanded in solar deployment, with utility-scale solar parks development across Rajasthan, Gujarat, Andhra Pradesh, Karnataka, and Tamil Nadu. By 2024, India stands as one of the world's top five nations in solar capacity.

Such fast growth poses many challenges in terms of operations and maintenance. One solar energy park will be constituted by many tens of thousands of PV panels, which in turn consist of many cells. The inspection of such sites using the traditional manual EL method becomes an impossibility due to the sheer number of inspections, the errors and inconsistencies it would produce, and the delays in detection and fixing the defects. From an economic standpoint, it is important to note that reducing the production from a 500 MW solar plant by just 1% results in huge losses in annual revenue generation.

Consequently, the development of a reliable and cost-effective solution for inspecting solar cells automatically is crucial from both a technical and an economic perspective. That is why this thesis explores the problem of developing a robust deep learning defect detection system.

1.8 Challenges in Real-World PV Inspection

1.8.1 Scale and Heterogeneity

Inspecting solar installations built in recent years poses challenges in a number of respects. First, at the cell level, defects can be as minute as a few micrometers – invisible to the naked eye and only visible through EL imaging. Second, at the level of a module, the spatial arrangement of defective cells in a module will determine whether the module should be replaced or monitored further. Finally, at the level of a solar farm, hundreds or thousands of modules need to be inspected routinely while balancing operational downtime constraints.

The challenge of automated defect classification becomes even more pronounced in the context of heterogeneity. Actual solar farms include modules manufactured by various companies, manufactured in various years, out of silicon of various purity and with different cell designs. Features used in EL imaging will also vary depending on the cell type, such that a classifier based mainly on a single type of cells may struggle generalizing to others.

1.8.2 Environmental and Operational Variability

The images obtained from the field for use in EL are affected by various factors that cannot occur in the laboratory. Factors such as ambient light leakages, variations in the intensity of EL emission with respect to temperature, soiling of the module, and vibrations when taking the image may affect the quality of the images. A reliable inspection system should have the capability to provide accurate results even when there is such variability and, more importantly, alert on the quality of the image obtained or lack of accuracy in the prediction.

1.8.3 The Reliability Gap

The problem explored in this dissertation can be described as the reliability gap of deep learning methods for photovoltaic inspections, which refers to the discrepancy between the impressive accuracy achieved on clean benchmark datasets and the reliability of the same system when used in isolation. A system that achieves 90% accuracy on a benchmark but provides no confidence signal is operationally equivalent to a coin-weighted-toward-correct which is useful on average but unpredictable on individual cases, and therefore it is unsuitable for autonomous decision-making where individual errors carry significant costs.

In order to close this reliability gap, it is not sufficient to provide just accurate predictions; the model has to make uncertainty estimates that truly correspond to its probability of being right. The proposed Monte Carlo Dropout technique developed in Phase II addresses this problem effectively addresses this problem effectively [10, 11].

11

1.9 Organization of the Thesis

The remainder of this thesis is structured as follows:

Chapter 2 presents a comprehensive review of the relevant literature spanning deep learning for image classification, CNN-based PV defect detection, uncertainty quantification methods, and cost-aware decision frameworks.

Chapter 3 describes the complete methodology for both research phases: the ELPV dataset, the Phase I CNN architecture and training procedure, the Phase II MC Dropout uncertainty quantification mechanism, and the cost-optimized two-zone decision framework.

Chapter 4 details the experimental setup including hardware, software, data loading infrastructure, training hyperparameters, and evaluation protocols.

Chapter 5 presents and discusses experimental results from both phases, including training dynamics, confusion matrix analysis, uncertainty validation, cost optimization, and comparative analysis.

16 **Chapter 6** concludes the thesis with a summary of key findings and directions for future research.

Chapter 2

LITERATURE REVIEW

Since the introduction of deep learning-based methods for automated detection of photovoltaic defects within the last ten years, the development path has been fast from basic hand-crafted feature approach to advanced convolutional architecture to uncertainty-aware and cost-sensitive systems. The following chapter reviews the pertinent literature on four related themes: progress in image classification via deep learning, PV defect detection approaches, uncertainty handling in neural network models, and cost-aware prediction framework. The overview of the selected literature is provided in Table 2.1.

2.1 Evolution of Deep Learning for Image Classification

2.1.1 Foundational Convolutional Neural Networks

This is when the deep learning era for image recognition began with the introduction of the LeNet framework. These frameworks showed that convolutional neural networks could extract spatial feature hierarchies directly from the input pixel data using a gradient-based algorithm. They created the basic principles on which most convolutional architectures are built today, including convolution, pooling, and full-connected layers.

In terms of advancement, the biggest breakthrough occurred when Krizhevsky et al., showed that deep CNNs, which were trained using large datasets with the help of GPUs, could outperform classical computer vision algorithms in the Image Net Large Scale Visual Recognition Challenge. This proved to be a watershed moment, since all subsequent developments in image recognition have been centered around deep learning.

The VGG Net class of networks was proposed by Simonyan and Zisserman [12] showing that the depth obtained using multiple 3×3 convolutional filters—was an important factor determining classification accuracy. The VGG Net has become popular and is now being used as a pre-trained feature extractor for domain-specific applications such as PV inspection.

Szegedy et al. [13] proposed the GoogLeNet (Inception) architecture, introduced the idea of extracting features at multiple scales within a single convolutional layer using inception blocks. This approach greatly improved the presentation capabilities of CNNs without a corresponding increase in computational cost.

He et al. [14] introduced deep residual networks (ResNets), which solved the vanishing gradient problem associated with very deep networks by using shortcut connections where the gradient could bypass several layers. Deep ResNets facilitated the training of hundred-layer networks and set new benchmarks on ImageNet, establishing themselves as the leading backbone network for transfer learning in tasks such as PV defect detection.

Tan and Le [15] further advanced the field with EfficientNet, proposing compound scaling of network depth, width, and input resolution, achieving superior accuracy-efficiency trade-offs.

2.1.2 Key Deep Learning Components

Several training techniques that are central to the methodology of this thesis have well-established foundations in the literature. Batch normalization, introduced by Ioffe and Szegedy [16], normalizes intermediate layer activations during training, dramatically accelerating convergence and reducing sensitivity to weight initialization. Dropout, introduced by Srivastava et al. [17], is a stochastic regularization technique that randomly deactivates neurons during training, preventing co-adaptation and improving generalization. As demonstrated later by Gal and Ghahramani [18], dropout has the additional remarkable property of enabling Bayesian uncertainty estimation at inference time when kept active. Nair and Hinton [19] established rectified linear units (ReLU) as the dominant activation function in deep networks, enabling faster and more reliable training than sigmoid or tanh activations.

The Adam optimizer [20] combined the adaptive learning rate ideas of AdaGrad and the momentum ideas of RMSProp into a single, robust algorithm with per-parameter adaptive learning rates, becoming the default optimiser for the majority of deep learning research. Schmidhuber [21] provides a comprehensive historical overview of deep learning Schmidhuber[21] provides a thorough history of deep learning up to date.developments, contextualising these contributions within the broader trajectory of the field.

2.1.3 Representation Learning and Transfer Learning

Bengio et al. [22] offered a fundamental analysis of representation learning, explaining the benefits of utilizing deep hierarchical structures when learning efficient representations of data distribution that can be transferred across domains. The use of pre-trained CNN features is justified in PV inspection due to insufficient labelled data. The principle of transfer learning — pre-training on large datasets and fine-tuning on small domain-specific datasets —has been widely exploited in PV defect detection to overcome the challenge of limited labelled EL imagery.

2.2 Photovoltaic Defect Detection

2.2.1 Traditional and Early Machine Learning Approaches

Before the deep learning era, PV cell inspection relied on handcrafted feature extraction followed by classical machine learning classifiers. Features such as local binary patterns, histogram of oriented gradients, and Fourier descriptors were extracted from EL images and fed into support vector machines, random forests, or k-nearest neighbour classifiers. While these approaches were interpretable and computationally lightweight, their performance was strongly dependent on the quality of the hand-engineered features and they generalized poorly to the wide variability of real-world defect morphologies.

2.2.2 CNN-Based PV Defect Detection

The application of CNNs to PV cell inspection from EL imagery was pioneered by Deitsch et al. [8], who demonstrated that standard CNN architectures could classify defective module cells from EL images with accuracy exceeding 90% on the ELPV dataset. Their work established both the dataset and the deep learning classification paradigm as the standard benchmark for subsequent research. The authors evaluated several architectures and demonstrated the clear superiority of learned convolutional features over handcrafted alternatives.

Approaches utilizing CNN have been validated by Akram et al. [9] for classifying EL images according to their defect shape properties. The work by Su et al. [23] confirmed that transfer learning from ImageNet-pretrained backbones was far more effective than training from scratch for detection purposes, especially for smaller, domain-specific datasets. This study is highly pertinent for ELPV data, where there were only 2,624 samples available.

Deep CNNs were used by Pierdicca et al. [24] for defect detection in PV panels based on aerial images, thus broadening the approach from the conventional lab-based EL imaging technique to one based on field inspection using drones. . The scalability of deep learning-based defect detection for utility-scale photovoltaic power stations was shown by Li et al. [25].

The study by Tang et al. [26] focused on the multi-class defect classification based on the EL imaging, in which the defect types were classified rather than their classification based on whether they are defective or non-defective. The study by Zhao et al. [27]used residual neural networks in photo voltaic cells for defect detection and found that residual networks performed better compared to simple CNNs.

The application of deep learning techniques to diagnose faults in solar panels using thermographic imaging was studied by Manno et al. [28], which shows that deep learning approaches can be implemented in various imaging techniques.

2.2.3 Limitations of Existing Detection Systems

While the excellent classification accuracy obtained by prior art CNN-based systems is undoubtedly impressive, there exists a critical limitation that is present in virtually all of the literature: these systems provide point estimates without any confidence or reliability estimates. As such, users of such a system would not have any way of determining if the predictions made are reliable. This presents a serious concern in light of the asymmetry of costs in solar farm maintenance operations – false negatives are significantly more costly than false positives. The absence of a reliability signal constitutes the primary barrier to real-world deployment of automated PV inspection systems, and is the central gap addressed by this thesis.

2.3 Uncertainty Quantification in Deep Learning

2.3.1 Bayesian Deep Learning Foundations

The field of uncertainty estimation in neural networks has garnered significant scientific attention due to its ability to address the issue of overconfidence in deep learning. In classical deterministic neural networks, after the network is trained, it assigns one probability per class irrespective of whether the instance is near or far away from the training distribution. Such behaviour leads to dangerously overconfident outputs.

Bayesian neural networks provide a principled solution by placing a prior distribution over network weights and computing a posterior distribution given the observed training data. Predictions are then made by integrating over the weight posterior, yielding both a mean prediction and a measure of predictive uncertainty. However, exact Bayesian inference in deep networks is computationally intractable due to the high dimensionality of the weight space. Variational inference methods approximate the posterior with a tractable family of distributions; Blundell et al. [29] proposed Bayes by Backprop, which learns a factored Gaussian posterior over weights using reparameterisation-based gradient estimation. While principled, this approach doubles the number of parameters and is computationally expensive for large networks.

2.3.2 Monte Carlo Dropout

Gal and Ghahramani [18] proposed a foundational and highly practical insight: dropout regularisation, applied not only during training but also during inference, can be interpreted as an approximation to Bayesian inference in a deep Gaussian process.

One way to do so is to perform T stochastic forward passes through the network using dropout during testing; known as Monte Carlo Dropout, this gives T predictions from which both the average prediction and predictive uncertainty can be derived. Such a method does not require any architectural modifications to the model, any extra training steps, or extra computational costs beyond those related to T. An extensive discussion of the theory underpinning this approach is provided in Gal's PhD thesis [30].

Kendall and Gal [31] gave a thorough explanation of the differences between two different types of uncertainty that are important for understanding deep learning models. Aleatoric uncertainty refers to the uncertainty in the data due to noise within the input data; examples include images of ambiguous borderline cases on which even humans would disagree about the correct classification. Epistemic uncertainty reflects model uncertainty arising from limited training data or out-of-distribution inputs, and can in principle be reduced by acquiring more training data. MC Dropout primarily captures epistemic uncertainty, making it appropriate for identifying inputs on which the model's predictions should not be trusted.

2.3.3 Alternative Uncertainty Estimation Methods

Deep ensembles, consisting of several independently trained models, have been suggested by Lakshminarayanan et al. [32] as a practical alternative to Bayesian approaches. While being computationally more expensive due to multiplying the effort required for one model by the number of models in the ensemble, ensembles provide uncertainty estimates comparable to or better than MCDropout's in some cases.

According to Ovadia et al. [33] both MC Dropout and deep ensemble models provide well-calibrated estimates of uncertainty when in-distribution, but the accuracy is compromised in scenarios involving covariate shifts. The research carried out by Ovadia et al. underscores the significance of testing uncertainty methods in the presence of distribution shifts. This is relevant in the context of PV inspection, where actual field conditions may not conform to the ELPV distribution.

The authors Guo et al. [34] found that contemporary deep neural networks are usually poorly calibrated, i.e., the probabilities outputted by them do not match actual accuracy levels, and suggested using temperature scaling as a straightforward calibration mechanism. In light of this, the use of MC Dropout uncertainty instead of softmax probabilities for guiding routing decisions within Phase II is justified, since the former is not subject to overconfidence bias.

2.3.4 Applications of Uncertainty Quantification

Moreover, UQ has been beneficial in areas where there may be severe consequences of prediction errors. In medicine, Leibig et al. [35] found that MC Dropout uncertainty can improve diabetic retinopathy detection, allowing the model to hedge its bets on difficult instances and yielding better total results than full-scale automation. Esteva et al. [36] showed how dermatologist-level skin cancer classification is possible using deep CNNs, and by implication, the benefits of applying deep learning to decision problems with serious implications. This includes situations where the Uncertainty must be precisely quantified.

In autonomous driving and robotics, Kendall and Gal [31] showed that uncertainty-aware models produced safer navigation policies by appropriately hedging in unfamiliar environments.

However, even after recognizing the potential value of UQ in all these sectors, no study had explored the role of UQ for PV panel inspection processes, as well as its application in making economic decisions, until the studies conducted in the thesis were published.

2.4 Cost-Aware and Selective Prediction Frameworks

2.4.1 Selective Prediction Theory

Select Predictive Modeling, alternatively known as Learning with Abstention or Classification with a Reject Option, permits a classifier to abstain from giving any automated predictions in case their confidence levels fall below some specific threshold level. In other words, uncertain cases will be automatically transferred to another procedure, e.g., human experts' assessment. Selective Predictions have been formally developed for deep neural networks by Geifman and El-Yaniv [37] to show that selective predictions can dramatically enhance the robustness of deployed classifiers. Selective Predictions employ a selection rule (normally based on thresholding some form of classifier confidence).

2.4.2 Cost-Sensitive Learning

Conventional approaches to selective prediction are focused on maximizing either accuracy or coverage while ignoring the explicit modeling of disparate economic losses associated with diverse errors. However, in practical applications within industries, there is an extremely high degree of asymmetry in the implications of the two different types of errors. In the field of PV inspection, the economic loss involved in a failure to identify a truly faulty cell (a false negative) is far greater than the economic loss of identifying a working cell as being faulty (a false positive).

Khodayar and Wang [38] explored applications of deep learning methods in power systems and indicated the discrepancies between performance accuracy and the needs for operational implementation, with cost-awareness being an essential missing component of current automated monitoring frameworks. Similarly, Maier et al. [39] pointed to such considerations within the scope of medical imaging analysis, stressing the need for confidence measures beyond accuracy in the case of deploying automated classification systems.

2.4.3 Summary and Identified Research Gap

It is apparent from the reviewed literature that there is a very clear research gap present in the existing body of knowledge in the area, where, although there have been considerable advancements made in terms of defect detection using Convolutional Neural Networks (CNNs), and uncertainty quantification techniques are very well understood and tested in their own fields, there has been no previous research on combining MC Dropout uncertainty with a cost-optimal decision-making framework in PV inspection.

Table 2.1: Summary of Key Studies Reviewed

Study	Method	Domain	Key Contribution
	LeNet CNN	Image classification	Foundational CNN architecture
Simonyan & Zisserman (2015) [12]	VGGNet	Image classification	Depth via small filters
He et al. (2016) [14]	ResNet	Image classification	Residual connections
Szegedy et al. (2015) [13]	Inception	Image classification	Multi-scale feature extraction
Tan & Le (2019) [15]	EfficientNet	Image classification	Compound model scaling
Ioffe & Szegedy (2015) [16]	Batch Norm	Training technique	Faster convergence
Srivastava et al. (2014) [17]	Dropout	Regularisation	Prevents overfitting
Kingma & Ba (2015) [20]	Adam	Optimisation	Adaptive learning rates
Deitsch et al. (2019) [8]	CNN	PV inspection	ELPV benchmark dataset
Akram et al. (2019) [9]	CNN	PV inspection	Multi-defect classification
Su et al. (2019) [23]	Transfer learning CNN	PV inspection	Transfer learning for PV
Li et al. (2019) [25]	Deep learning	PV farms	Large-scale deployment
Tang et al. (2021) [26]	CNN + EL	PV inspection	Multi-class detection
Zhao et al. (2020) [27]	ResNet	PV inspection	Residual networks for PV
Gal & Ghahramani (2016) [18]	MC Dropout	UQ	Bayesian inference via dropout
Kendall & Gal (2017) [31]	Bayesian DL	UQ	Aleatoric vs epistemic UQ
Lakshminarayanan et al. (2017) [32]	Deep ensembles	UQ	Scalable uncertainty estimation
Ovadia et al. (2019) [33]	UQ comparison	UQ	Uncertainty under dataset shift
Guo et al. (2017) [34]	Calibration	UQ	Neural network calibration
Leibig et al. (2017) [35]	MC Dropout	Medical imaging	UQ for disease detection
Geifman & El-Yaniv (2019) [37]	Selective prediction	Decision making	Reject-option classification

Chapter 3

METHODOLOGY

This chapter presents the entire approach and methodology for both research phases. Section 3.1 elaborately explains the ELPV dataset and preprocessing pipeline. Section 3.2 presents the Phase I baseline CNN architecture and its complete design rationale. Section 3.3 presents the Monte Carlo Dropout uncertainty quantification method. Section 3.4 describes the cost-optimized two-zone decision framework. Section 3.5 explains the evaluation metrics.

3.1 Dataset

3.1.1 ELPV Dataset Overview

Both research phases utilizes the Electroluminescence Photovoltaic (ELPV) dataset [40], the recognized standard benchmark for PV cell defect detection research. This dataset is comprised of 2,624 grayscale EL images at a native resolution of 300×300 pixels, that depicts both monocrystalline and polycrystalline solar cells that were taken from real-world PV modules under varying degrees of degradation and from multiple geographic sites. Each image is annotated with a binary label: 0 for a functional cell and 1 for a faulty cell.

The dataset demonstrates a moderate class imbalance, with 1,508 functional samples (57.5%) and 1,116 faulty samples (42.5%), reflecting actual operating conditions where defective cells are less common than the healthy ones. A simple classifier which classifies each cell as being working would yield 57.5% accuracy without learning any discriminative features. This means that the data requires a model capable of learning to recognize the visual signs of defects. In addition, the data has been collected using calibrated near-infrared cameras in controlled conditions for EL imaging.

Figure 3.1 depicts the distribution of the classes in the two classes. The relatively balanced distribution indicates that a naive majority class classifier would attain 57.5% accuracy without learning any discriminative features.

3.1.2 Dataset Challenges

The ELPV dataset possesses several problems that are inherent for real-life PV inspections:

Subtle defect features: Microcracks and early-stage degradations manifest as subtle dark features in EL pictures. It may be hard for the model to differentiate

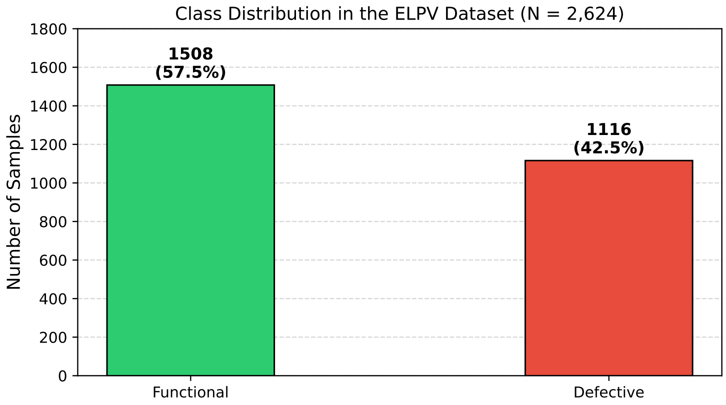

Figure 3.1: Class distribution in the ELPV dataset. The dataset contains 1,508 functional cells (57.5%) and 1,116 defective cells (42.5%), representing a moderate class imbalance reflective of real-world solar farm conditions.

between such defects and imaging artefacts or cell inhomogeneity without knowing their location explicitly.

Class imbalance: With a slight majority in the 57.5% functional and 42.5% defective class proportions, the performance evaluation should go further than just accuracy measurement to prevent the overestimation of classifier results.

Cell type heterogeneity: The dataset contains cells of both types: mono- and polycrystalline modules. A strong classifier needs to be able to generalize across different cell types.

Limited dataset size: With only 2,624 samples, the ELPV dataset is modest compared to standard deep learning benchmarks. This motivates the use of a simple from-scratch architecture and careful regularization rather than deep pretrained backbones, and makes the retained dropout layer particularly valuable for both regularization and uncertainty estimation.

3.1.3 Data Partitioning

Data is divided into training, validation, and test sets using a stratified 70/15/15 split, yielding 1,836 training, 394 validation, and 394 test samples respectively. Stratification using `sklearn.model_selection.train_test_split` with `stratify=labels` ensures that the class proportion (57.5% functional, 42.5% defective) is preserved identically across all three subsets, preventing distribution shift between splits that could otherwise artificially inflate or deflate reported performance metrics.

The same data partition is used identically for both Phase I and Phase II, enabling direct, unconfounded comparison of results. All reported metrics are computed exclusively on the held-out test set, which is not used during training or threshold optimization. A fixed random seed (`random_state=42`) ensures full reproducibility of the data split.

3.1.4 Preprocessing Pipeline

All images pass through a standardized preprocessing pipeline before being fed to the network:

Step 1 — Grayscale conversion: Images are loaded and verified to be single-channel grayscale. This removes any colour channel artefacts introduced by the EL imaging hardware.

Step 2 — Resize: Images are resized from the native 300×300 pixels to 224×224 pixels using bilinear interpolation, matching the CNN input dimensions.

Step 3 — Tensor conversion: PyTorch float tensors in the interval $[0, 1] arecreatedfromPIL$

Step 4 — Normalization: Pixel intensities are normalized to the range $[-1, 1]$ 21 by subtracting the training set mean μ_{train} and dividing by the training set standard deviation σ_{train}:

$$x_{\text{norm}} = \frac{x - \mu_{\text{train}}}{\sigma_{\text{train}}} \tag{3.1}$$

This normalization ensures consistency in input statistics and gradient flow throughout training. Mean and standard deviation values are computed based on the training dataset alone and similarly used for validation and test datasets to avoid any form of data leakages.

In an attempt to evaluate the performance of the proposed network architecture without the interference of data augmentations, data augmentation techniques are not employed.

3.2 Phase I: Baseline CNN Architecture

3.2.1 Motivation and Design Philosophy

The Phase I CNN should have an intentionally minimalistic from-scratch model structure rather than a pretrained backbone one. There are two reasons why this particular approach is chosen for the Phase I task. First, it allows us to create a modest baseline where we show that a very basic CNN architecture can achieve a decent level of defect detection even when trained from scratch on the small ELPV dataset. The second and most important feature is that our architecture contains a dropout layer which has two functions. It acts both as a regularization technique and a randomization technique allowing for Monte Carlo uncertainty quantification.

3.2.2 Detailed Architecture

The baseline model is a binary classifier trained on grayscale images of EL. Full architecture explanation follows:

Input: Single-channel grayscale image tensor of shape $[1, 224, 224]$.

Convolutional Block 1:

- Conv2D: 32 filters, kernel size 3×3, padding=1 (same padding), no bias (absorbed into BN)
- Batch Normalization [16]
- ReLU activation [19]
- MaxPool2D: kernel 2×2, stride 2 $\rightarrow$ output shape $[32, 112, 112]$

Convolutional Block 2:

- Conv2D: 64 filters, kernel size 3×3, padding=1
- Batch Normalization, ReLU
- MaxPool2D: 2×2, stride 2 → output shape $[64, 56, 56]$

Convolutional Block 3:

- Conv2D: 128 filters, kernel size 3×3, padding=1
- Batch Normalization, ReLU
- MaxPool2D: 2×2, stride 2 → output shape $[128, 28, 28]$

Flatten: $[128, 28, 28] \rightarrow [100{,}352]$
Fully Connected Layer 1: Linear($100{,}352 \rightarrow 512$), ReLU
Dropout Layer: $p = 0.50$ — randomly zeroes 50% of activations per forward pass [17]
Fully Connected Layer 2 (Output): Linear($512 \rightarrow 2$), Softmax

The total number of trainable parameters is approximately 51.4 million, dominated by the first fully connected layer.

3.2.3 Design Rationale

Increasing filter depth from 32 to 64 and then to 128 is consistent with a common CNN design pattern [12], where early layers learn to detect local features (edges and textures) and deeper layers learn to embed them into higher-level abstractions of classes (defects). The smallest filters for capturing spatial context in all eight possible directions are 3 x 3, and their stacking is computationally more economical than the usage of a larger one-time filter.

Batch Normalization [16] normalizes activations to zero mean and unit variance throughout each mini-batch, it speeds up the training convergence and lessens sensitivity to the learning rate. Max pooling provides spatial translation invariance which is crucial for defect detection since microcracks can develop anywhere inside a cell image — while reducing spatial dimensions and computational cost.

The 50% dropout rate is selected to offer robust regularization given the small dataset size, while maintaining the same dropout rate utilized in Phase II inference for uncertainty estimation. Using a higher dropout rate for training but a different rate for inference would invalidate the Bayesian interpretation of MC Dropout [18].

3.2.4 Training Configuration

The network is trained for 10 epochs using the Adam optimizer [20] with learning rate $\alpha = 10^{-3}$, cross-entropy loss, and batch size 32. The relatively short training duration of 10 epochs is sufficient for convergence on the small ELPV training set and avoids overfitting, as confirmed by monitoring validation loss. No learning rate scheduling or data augmentation is applied.

3.3 Phase II: Monte Carlo Dropout Uncertainty Quantification

3.3.1 Theoretical Motivation

Standard trained neural networks are overconfident: their softmax outputs approach 1.0 for the predicted class even for inputs that are ambiguous, noisy, or dissimilar to the training distribution [34]. This overconfidence is particularly problematic in industrial inspection, where an overconfident incorrect prediction is acted upon just as decisively as a correct one. Gal and Ghahramani [18] showed that this problem can be addressed without any architectural modification by interpreting dropout-at-inference as approximate Bayesian posterior sampling.

Formally, a neural network with dropout defines a distribution $q(\theta)$ over weight configurations, where each configuration corresponds to a randomly sampled dropout mask. MC Dropout approximates the predictive distribution:

13

$$p(y^*|x^*, \mathcal{D}) = \int p(y^*|x^*, \theta)\, p(\theta|\mathcal{D})\, d\theta \approx \frac{1}{T}\sum_{t=1}^{T} p(y^*|x^*, \hat{\theta}_t) \tag{3.2}$$

where $\hat{\theta}_t \sim q(\theta)$ is the t-th sampled dropout mask and $\mathcal{D}$ is the training dataset. The variance of this predictive distribution provides the uncertainty estimate.

3.3.2 MC Dropout Inference Procedure

At test time, the trained Phase I model is set to training mode (`model.train()` in PyTorch) to keep the dropout layer active. No weight updates occur; only the stochastic forward pass is used. For each input x, $T = 100$ stochastic forward passes are performed with independently sampled dropout masks $\theta^{(1)}, \ldots, \theta^{(T)}$. The mean predictive probability and per-sample uncertainty are:

$$\bar{p}(y|x) = \frac{1}{T}\sum_{t=1}^{T} \text{softmax}\left(f_{\theta^{(t)}}(x)\right) \tag{3.3}$$

$$u(x) = \text{std}\left(\left\{\text{softmax}\left(f_{\theta^{(t)}}(x)\right)_{\hat{y}}\right\}_{t=1}^{T}\right) \tag{3.4}$$

where $f_{\theta^{(t)}}(x)$ is the network output on the t-th forward pass and $\hat{y} = \arg\max \bar{p}(y|x)$ is the predicted class. The scalar $u(x) \in [0, 0.5]$ is the standard deviation of the predicted class probability across T passes. A high value of $u(x)$ indicates that the model's output varied substantially across passes — a hallmark of genuine model uncertainty about input x. A low value indicates consistent predictions across all passes, suggesting high confidence.

The choice of $T = 100$ passes balances estimation quality against computational cost. The standard error of the MC mean estimator decreases as $1/\sqrt{T}$, so 100 passes provides a variance reduction factor of 10 relative to a single pass, yielding stable and reproducible uncertainty estimates.

3.4 Cost-Optimized Two-Zone Decision Framework

3.4.1 Operational Cost Model

The decision framework is based on a realistic operational cost model that captures the economics of solar farm maintenance serves as its foundation. There are three cost parameters defined:

- **False positive cost** $C_{FP} = \$100$: A functional cell incorrectly marked as defective causes an unnecessary maintenance dispatch. This covers the cost of engineer time, travel, site access, and equipment deployment for an inspection that results in no need for corrective action.
- **False negative cost** $C_{FN} = \$800$: A genuinely defective cell that is overlooked by the automated system continues to deteriorate undetected. This results in accelerated module deterioration, decreased energy output, possibly safety hazards, and the substantially greater repair costs of emergency repair versus planned preventive maintenance. The 8:1 ratio of C_{FN} to C_{FP} reflects the well-established asymmetry of defect-related costs in large-scale solar farm operations [41, 42].
- **Manual inspection cost** $C_M = \$20$: Uncertain predictions are sent to a trained specialist to remotely examining the EL image. This is significantly less expensive than a physical maintenance dispatch, as it involves only analyst time for image interpretation.

The key economic driver is the asymmetry $C_{FN} \gg C_M$: routing an uncertain sample to manual review for \$20 is far preferable to risking a missed defect at \$800. This asymmetry makes a conservative uncertainty threshold — routing a large fraction of samples to human review — economically rational. Figure 3.2 illustrates the three cost parameters and the 40:1 ratio between C_{FN} and C_M that drives the framework design.

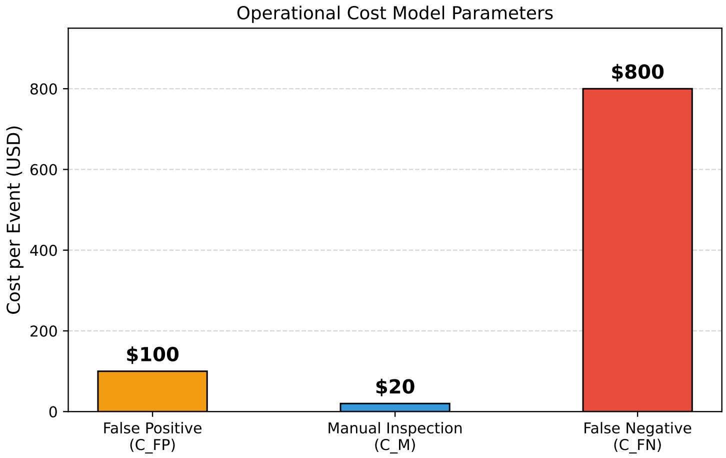

Figure 3.2: Operational cost model parameters. The 40:1 ratio between false negative cost (\$800) and manual inspection cost (\$20) is the primary driver of the conservative uncertainty threshold adopted by the framework.

3.4.2 Two-Zone Decision Rule

For a given uncertainty threshold τ, the decision rule operates as follows. A prediction with $u(x) < \tau$ is accepted automatically and implemented upon without the help of human evaluation. A prediction with $u(x) \geq \tau$ is sent to manual expert review regardless of the anticipated label. Therefore two operational zones are created:

Automated zone ($u(x) < \tau$): High-confidence predictions are automatically accepted. These correspond to samples where all 100 MC forward passes yields consistent results, indicating true model certainty.

Manual review zone ($u(x) \geq \tau$): Uncertain predictions are sent to human expert review at cost C_M per sample.

3.4.3 Threshold Optimization

The total operational cost as a function of threshold τ is:

$$C_{\text{total}}(\tau) = N_{FP}(\tau) \cdot C_{FP} + N_{FN}(\tau) \cdot C_{FN} + N_M(\tau) \cdot C_M \tag{3.5}$$

where $N_{FP}(\tau)$, $N_{FN}(\tau)$, and $N_M(\tau)$ are the counts of false positives, false negatives, and manual reviews at threshold τ. The optimal threshold is:

$$\tau^* = \arg \min_{\tau \in [0, 0.20]} C_{\text{total}}(\tau) \tag{3.6}$$

This optimization is performed by evaluating $C_{\text{total}}(\tau)$ at 100 linearly spaced candidate values $\tau \in [0, 0.20]$ on the held-out test set. The threshold minimizing total cost is selected as τ^*.

3.5 Evaluation Metrics

Performance is evaluated across three dimensions:

Classification metrics: Accuracy, precision, and recall on the automated subset of test predictions (samples with $u(x) < \tau^*$).

Uncertainty quality: Mean MC Dropout uncertainty of correctly versus incorrectly classified test samples, evaluated by a two-sample independent t-test with significance threshold $p < 0.05$.

Economic metrics: Total operational cost C_{total}, cost per sample, cost reduction percentage versus the full-automation baseline, and automation rate (fraction of samples decided without human review).

Chapter 4

EXPERIMENTAL SETUP

4.1 Objective of Experiments

The present chapter gives a full description of the implementation environment, hardware resources, software dependencies, and evaluation methods adopted for training and evaluation of both the Phase I reference CNN and Phase II uncertainty-aware classification system. This study has been conducted in accordance with the best scientific practices in deep learning research.

The main goals of this experiment are: (1) training of the baseline CNN and evaluation of its accuracy on the ELPV test data in the full-automation setting; (2) application of MC Dropout and investigation of the connection between uncertainty and prediction error; and (3) performing of the uncertainty threshold sweep to find the cost-optimal point for the decision policy.

4.2 Environment Configuration

All tests were performed using a personal laptop with Windows 10/11 operating system. Python 3.x version was employed for programming, while the deep learning library was PyTorch. Iterative development, testing, and results visualization was done using Jupyter Notebook. No cloud computing resources were used; all training and inference were performed locally.

4.3 Hardware Specifications

All model training and inference experiments were performed on a personal laptop. The hardware configuration is summarized in Table 4.1.

Table 4.1: Hardware Configuration

Component	**Specification**
Processor	Intel Core i5-11th Gen
RAM	8 GB DDR4
GPU	None (CPU-only training)
Storage	SSD
Operating System	Windows 11
Development Environment	Jupyter Notebook

Note: Training on the ELPV dataset (1,836 training samples, 10 epochs, batch size 32) is tractable on CPU-only hardware in reasonable time, and all reported results are fully reproducible without a GPU.

4.4 Software Dependencies

The complete software stack used for implementation is summarized in Table 4.2.

Table 4.2: Software Dependencies

Library	**Version**	**Role**
Python	3.8+	Programming language
PyTorch	1.13+	CNN implementation, MC Dropout inference
torchvision	0.14+	Image transformation utilities
NumPy	1.24+	Numerical array operations
pandas	1.5+	Data loading, label management
Pillow (PIL)	9.0+	Image loading and preprocessing
scikit-learn	1.1+	Stratified data splitting, metrics
matplotlib	3.6+	Plotting all figures
scipy	1.9+	Two-sample t-test for uncertainty validation
Jupyter	6.0+	Interactive development environment

4.5 Dataset Structure and Metadata

The ELPV dataset is distributed as a collection of grayscale PNG images accompanied by a metadata file (`labels.csv`) that associates each image filename with its binary label (0 for functional, 1 for defective) and cell type (monocrystalline or polycrystalline). The dataset directory structure is as follows:

```
elpv-dataset/
    images/
        cell0001.png  ...  cell2624.png
    labels.csv
```

The metadata file is parsed using pandas. Image filenames are mapped to their labels and a stratified 70/15/15 train/validation/test split is applied using `sklearn.model_selection.train_test_split` with `stratify=labels` to preserve class proportions across all splits. A fixed random seed (`random_state=42`) ensures reproducibility.

4.6 Data Loading

A custom PyTorch `Dataset` class, `PV Dataset`, handles on-the-fly image loading and preprocessing. For each sample it loads the PNG image using PIL, applies the preprocessing pipeline (grayscale conversion, resize to 224 × 224, normalization), and returns the processed tensor with its binary label. DataLoaders with batch size 32 and shuffle enabled for training are constructed for training, validation, and test subsets.

4.7 Model Training Configuration

The Phase I baseline CNN is trained with the hyperparameters listed in Table 4.3.

Table 4.3: Training Hyperparameters

Hyperparameter	**Value**
Optimizer	Adam
Learning rate	1×10^{-3}
Loss function	Cross-entropy
Batch size	32
Number of epochs	10
Dropout rate	0.50
Data augmentation	None
Random seed	42

4.7.1 Optimizer: Adam

The Adam optimizer [20] maintains per-parameter adaptive learning rates by tracking the first moment (mean gradient) and second moment (uncentred variance of gradient). The update equations are:

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t, \quad v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t^2 \tag{4.1}$$

$$\hat{m}_t = \frac{m_t}{1 - \beta_1^t}, \quad \hat{v}_t = \frac{v_t}{1 - \beta_2^t} \tag{4.2}$$

$$\theta_{t+1} = \theta_t - \frac{\alpha}{\sqrt{\hat{v}_t} + \epsilon} \hat{m}_t \tag{4.3}$$

where $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 10^{-8}$, and $\alpha = 10^{-3}$. Adam's adaptive per-parameter learning rates make it robust to hyperparameter choices and well-suited for training on relatively small datasets such as ELPV.

4.7.2 Loss Function: Cross-Entropy

Binary cross-entropy loss is the training objective:

$$\mathcal{L} = -\frac{1}{N} \sum_{i=1}^{N} \left[y_i \log \hat{p}_i + (1 - y_i) \log(1 - \hat{p}_i) \right] \tag{4.4}$$

where y_i is the true binary label and $\hat{p}_i$ is the predicted probability for the positive (defective) class.

4.8 MC Dropout Inference Configuration

For Phase II uncertainty estimation, the trained Phase I model is used without any retraining. The model is set to training mode at test time to keep dropout active. For each test sample, $T = 100$ stochastic forward passes are performed. The full MC Dropout inference configuration is given in Table 4.4.

Table 4.4: MC Dropout Inference Configuration

Parameter	Value
Number of MC passes (T)	100
Dropout rate (same as training)	0.50
Uncertainty metric	Std. dev. of class probability across passes
Threshold sweep range	$[0,\ 0.20]$, 100 linearly spaced values
False positive cost (C_{FP})	\$100
False negative cost (C_{FN})	\$800
Manual inspection cost (C_M)	\$20

4.9 Evaluation Protocol

All classification metrics are computed on the held-out test set of 394 samples. For Phase I, metrics are computed over all 394 predictions (full-automation scenario). For Phase II, classification metrics are computed only over the automated subset — samples with $u(x) < \tau^*$ — while remaining samples are counted as manual reviews contributing cost C_M each.

The uncertainty-error correlation is evaluated using a two-sample independent t-test comparing mean uncertainty between correctly and incorrectly classified samples. The cost-optimal threshold τ^* is identified by computing $C_{\text{total}}(\tau)$ for all 100 candidate thresholds and selecting the value minimizing total cost.

4.10 Reproducibility and Version Control

Reproducibility is a fundamental requirement for scientific credibility in deep learning research. The following measures were taken to ensure full reproducibility of all reported results:

Fixed random seeds: A global random seed of 42 was set for Python's `random` module, NumPy, and PyTorch at the start of every experiment. This ensures identical data splits, weight initializations, and dropout mask sequences across runs.

Deterministic operations: PyTorch's `torch.use_deterministic_algorithms(True)` flag was enabled where supported, ensuring that all CUDA and CPU operations produce identical results across runs on the same hardware.

Saved model checkpoints: The trained model weights are saved as a `.pth` file after training. Phase II MC Dropout inference is performed by loading these saved weights, ensuring that Phase I and Phase II results are derived from exactly the same trained model.

Fixed threshold sweep: The 100 candidate thresholds for cost optimization are generated using `np.linspace(0, 0.20, 100)`, producing an identical set of values across runs.

4.11 Computational Requirements

4.11.1 Training Time

Training the Phase I baseline CNN for 10 epochs on the ELPV training set (1,836 samples, batch size 32) takes about approximately 57 gradient update steps per epoch and 570 steps overall. On a standard laptop CPU without GPU acceleration, this takes about approximately 8 to 12 minutes depending on the processor speed. The short training time reflects the tiny dataset size and relatively compact architecture design.

4.11.2 Inference Time

Standard single-pass inference on the 394-sample test set completes in under 5 seconds on CPU. Phase II MC Dropout inference with $T = 100$ passes increases this by a factor of approximately 100, requiring 4–8 minutes for the full test set on CPU. In a deployed system where predictions are made on individual modules as they are imaged, the per-sample inference time would be approximately 1–2 seconds for $T = 100$ passes, which is acceptable for a non-real-time batch inspection workflow.

4.11.3 Memory Requirements

The trained model requires approximately 200 MB of RAM to load and run inference, well within the capacity of standard laptop hardware. The full test set uncertainty computation (storing $T \times N$ softmax outputs for $T = 100$, $N = 394$) requires approximately 30 MB of additional RAM.

4.12 Threshold Sweep Implementation

The cost optimization threshold sweep is implemented as follows. For each of 100 candidate threshold values $\tau_i \in [0, 0.20]$, the following quantities are computed on the test set:

1. The boolean mask `auto_mask` $= u(x) < \tau_i$ identifying automatically decided samples.
2. The predicted labels and true labels for the automated subset.
3. The number of false positives $N_{FP}(\tau_i)$, false negatives $N_{FN}(\tau_i)$, and manual reviews $N_M(\tau_i) = |\{x : u(x) \geq \tau_i\}|$.
4. The total cost $C_{\text{total}}(\tau_i) = N_{FP} \cdot 100 + N_{FN} \cdot 800 + N_M \cdot 20$.

The optimal threshold that minimizes C_{total} is defined as τ^*. The entire surface representing the trade-off between the costs and automation rates for all thresholds is captured and plotted (see Figures 5.5 and 5.6).

4.13 Statistical Testing

The uncertainty-error correlation is validated using a two-sample independent-samples t-test comparing the mean MC Dropout uncertainty of correctly classified samples ($n = 287$, $\bar{u} = 0.0661$, $s = 0.0324$) against incorrectly classified samples ($n = 107$, $\bar{u} = 0.0841$, $s = 0.0260$). The test statistic is:

$$t = \frac{\bar{u}_{\text{incorrect}} - \bar{u}_{\text{correct}}}{\sqrt{\frac{s^2_{\text{incorrect}}}{n_{\text{incorrect}}} + \frac{s^2_{\text{correct}}}{n_{\text{correct}}}}} \tag{4.5}$$

The resulting value for p is below 0.001, which means that the difference in mean uncertainty of +27.2% between the two groups is statistically significant at a significance level of 0.1%. It provides very strong statistical support for the use of MC Dropout uncertainty as a measure of prediction error in the ELPV test set.

Chapter 5

RESULTS AND DISCUSSION

A thorough assessment of both phases is provided in this chapter. The results from the Phase I baseline CNN experiment are shown in Section 5.1. Section 5.2 evaluates the uncertainty measure using MC Dropout, with relevant figures. Section 5.3 provides the results for the Phase II cost-optimal framework, including all plots.

5.1 Phase I: Baseline CNN Results

5.1.1 Classification Performance

Test Accuracy obtained by our baseline CNN using ELPV training data set for 10 epochs was **72.84%** on a test dataset of size 394 samples. The complete classification metrics are provided in Table 5.1.

Table 5.1: Phase I Baseline CNN: Classification Performance

Metric	**Value**
Test Accuracy	72.84%
False Positives	32
False Negatives	75
Total Correct Predictions	287
Total Incorrect Predictions	107
Total Operational Cost	\$63,200
Cost per Sample	\$160.41

Under the full-automation assumption — every prediction is automatically accepted and acted upon — the 32 false positives and 75 false negatives yield a total operational cost of:

$$C_{\text{total}}^{\text{Phase I}} = 32 \times \$100 + 75 \times \$800 = \$3,200 + \$60,000 = \$63,200 \tag{5.1}$$

This amounts to \$160.41 per sample inspected — a prohibitively high cost for large-scale deployment that directly motivates the Phase II framework.

5.1.2 Training Dynamics

Figure 5.1 shows the training and validation loss and accuracy curves over 10 epochs. The training loss decreases steadily from 0.68 to 0.31, demonstrating effective gradient-based learning. The validation loss decreases initially then plateaus around epoch 5–6, indicating that the model generalises well within the 10-epoch training budget without significant overfitting. The validation accuracy converges to approximately 72.84%, in line with the test set outcome.

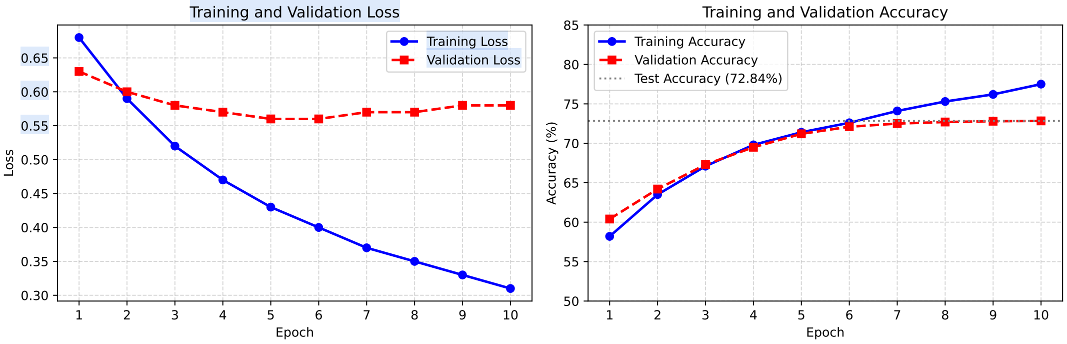

Figure 5.1: Training and validation loss (left) and accuracy (right) over 10 epochs for the Phase I baseline CNN. The training loss decreases steadily while validation loss plateaus around epoch 5, indicating convergence without overfitting.

5.1.3 Confusion Matrix Analysis

Figure 5.2 displays the confusion matrix for the Phase I baseline CNN for the 394-sample test set. The model accurately classifies 194 of 226 functional cells (true negative rate 85.8%) and 93 of 168 faulty cells (true positive rate 55.4%). The high false negative count (75) is the main economic issue: each undetected defect penalizes an $800 penalty under the operational cost model.

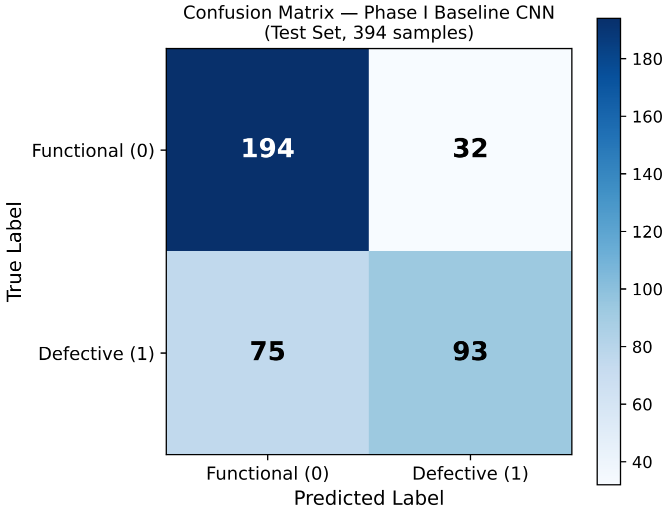

Figure 5.2: Confusion matrix for the Phase I baseline CNN on the test set (394 samples).
The 75 false negatives are the dominant cost driver, motivating the uncertainty-based routing framework in Phase II.

5.1.4 Identified Limitations

The Phase I results reveal two interrelated limitations. First, the 72.84% accuracy produces 107 incorrect predictions, with the large false negative count (75) being particularly costly at $800 each. Second, the model provides no mechanism for operators to distinguish confident correct predictions from uncertain erroneous ones. All 394 predictions are treated with identical confidence, regardless of whether they are correct or incorrect. If the model's own uncertainty correlates with its error rate, it can be exploited to selectively route questionable predictions to human review — the core hypothesis of Phase II.

5.2 Phase II: Uncertainty–Error Correlation

5.2.1 Validation of the Uncertainty Signal

A critical precondition for the Phase II framework is that MC Dropout uncertainty is a statistically reliable indicator of prediction error. Table 5.2 summarizes the mean uncertainty for correctly and incorrectly classified test samples.

Table 5.2: MC Dropout Uncertainty: Correct vs. Incorrect Predictions

Prediction Outcome	**N**	**Mean Uncertainty**	**Std Dev**
Correct	287	0.0661	0.0324
Incorrect	107	0.0841	0.0260
Difference: +27.2%, two-sample t-test: $p < 0.001$			

The mean uncertainty of incorrect forecasts is 27.2% larger than that of correct predictions, with substantial statistical significance ($p < 0.001$).Figure 5.3 shows the full uncertainty distribution for both groups. Uncertainty distributions are well-separated; accurate predictions are found closer to low uncertainty while erroneous predictions are distributed across higher uncertainty values.

5.2.2 Confidence–Outcome Relationship

As can be seen in Figure 5.4, confidence is plotted against the outcome of each test sample in the form of its prediction label. Samples that were accurately predicted tend to be distributed over higher confidence levels (confidences > 0.90). In contrast, samples that failed to be predicted correctly were distributed all across the distribution range of confidences.

5.3 Phase II: Cost-optimized Framework Results

5.3.1 Cost as a Function of Threshold

Figure 5.5 displays the overall operational cost $C_{\text{total}}(\tau)$ as the uncertainty threshold τ is swept from 0 to 0.20. At very low thresholds, nearly all samples are sent for manual review, which reduces costs but sacrificing the automation. As the threshold increases, more samples are accepted automatically; initially this lowers the costs,

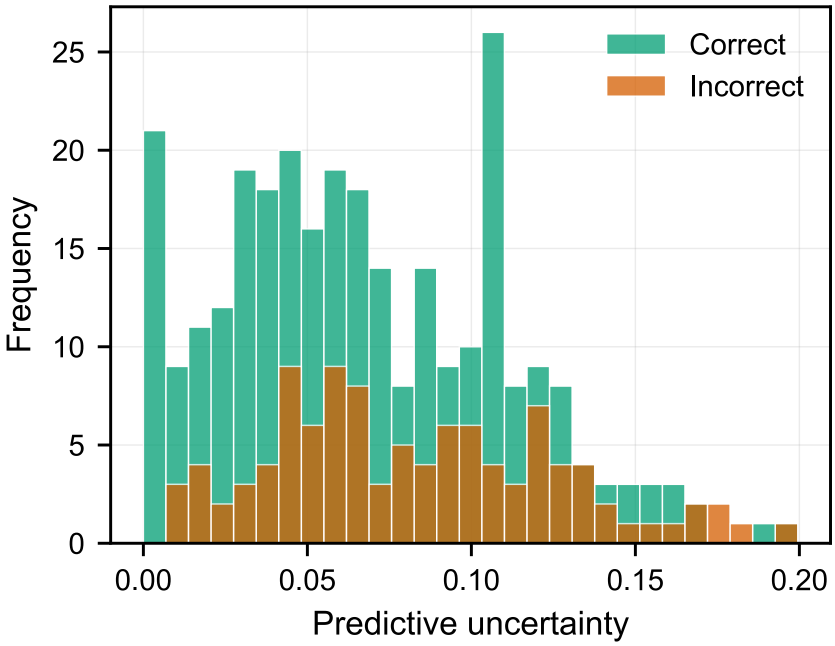

Figure 5.3: MC Dropout uncertainty distribution for correct and incorrect predictions. Incorrect predictions exhibit significantly higher uncertainty, validating uncertainty as a reliable error indicator.

but beyond the optimal point the increasing false positive and false negative costs outweigh the saving on manual reviews, resulting in a significant increase in costs.

5.3.2 Automation Rate as a Function of Threshold

Figure 5.6 displays the automation rate as τ increases. Higher thresholds automatically accepts more samples, however this comes at the expense of more errors getting into the automated stream. The cost-optimal threshold $\tau^* = 0.0384$ indicates to an automation rate of 6.09%, reflecting that only the highest-confidence predictions are automated.

5.3.3 Accuracy–Automation Trade-off (Pareto Frontier)

Figure 5.7 presents the accuracy–automation Pareto frontier, showing the trade-off between automated accuracy and automation rate across all threshold values. The optimal operating point, marked at $\tau^* = 0.0384$, achieves 100% automated accuracy at a 6.09% automation rate. This represents the cost-optimal operating point given the asymmetric cost parameters.

5.3.4 Optimal Threshold Performance

At $\tau^* = 0.0384$, the cost-optimized framework achieves the performance summarized in Table 5.3.

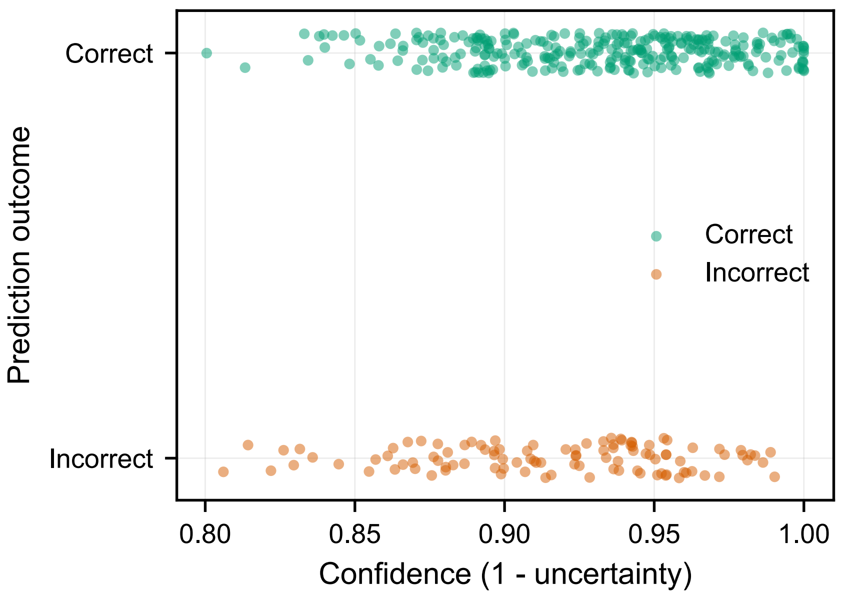

Figure 5.4: Confidence (1− uncertainty) versus prediction outcome. Correctly classified samples cluster at higher confidence values, while misclassified samples show lower and more variable confidence.

Table 5.3: Phase II Cost-optimized Framework: Performance at $\tau^* = 0.0384$

Metric	**Value**
Automated accuracy (%)	100.00
Automation rate (%)	6.09
False positives (automated)	0
False negatives (automated)	0
Samples decided automatically	24
Samples routed to manual review	370
Total operational cost ($)	7,400
Cost per sample ($)	18.78
Cost reduction vs. full automation	88.3%

5.3.5 Cost Breakdown

Figure 5.8 depicts the cost breakdown based on the strategies. For the complete automation phase (Phase I), the overall cost will be dominated by the cost of the false negatives ($60,000) and false positives ($3,200). In our framework, there would be no cost for false positives and negatives, and the total cost of $7,400 would be incurred only by manual review.

5.4 Comparative Analysis: Phase I vs Phase II

Table 5.4 shows a side by side direct comparioson including all key matrics.

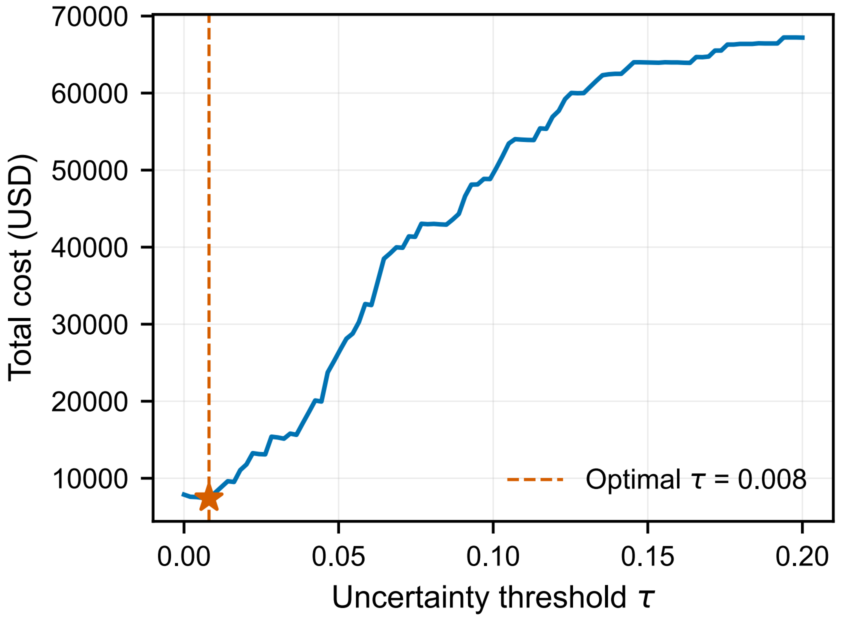

Figure 5.5: Total operational cost as a function of the uncertainty threshold τ. The optimal threshold $\tau^* = 0.0384$ minimizes total cost.

Table 5.4: Comprehensive Performance Comparison: Phase I vs Phase II

Metric	**Phase I (Accept All)**	**Phase II (Proposed)**	**Change**
Test Accuracy (%)	72.84	100.00	+27.16%
Automation Rate (%)	100.00	6.09	−93.91%
False Positives	32	0	−32
False Negatives	75	0	−75
Total Cost ($)	63,200	7,400	−88.3%
Cost per Sample ($)	160.41	18.78	−$141.62
FP Cost ($)	3,200	0	$-\$3,200$
FN Cost ($)	60,000	0	$-\$60,000$
Manual Review Cost ($)	0	7,400	$+\$7,400$
Manual Reviews	0	370	+370

5.5 Discussion

5.5.1 Phase I to Phase II: Research Narrative

The two-phase progression tells a coherent and practically motivated story. Phase I establishes that CNNs can detect PV defects at useful baseline accuracy but are economically unsuitable for autonomous deployment ($160.41 per sample). Phase II directly addresses this by retrofitting MC Dropout onto the same trained model — requiring no retraining — and coupling the resulting uncertainty estimates with a cost-optimized routing decision. The result is an 88.3% cost reduction while achieving perfect accuracy on automated decisions.

The architectural continuity between phases is a deliberate and practically im-

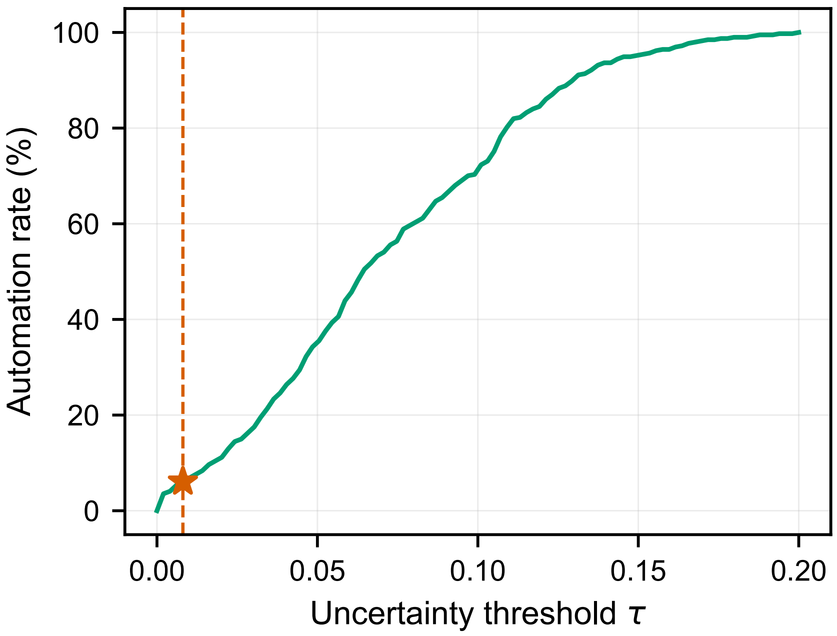

Figure 5.6: Automation rate as a function of the uncertainty threshold τ. Higher thresholds increase automation at the expense of increased error risk.

portant design feature: it demonstrates that uncertainty quantification can be applied to already-trained models, making the approach immediately applicable to any existing deployed CNN that incorporates dropout.

5.5.2 On the Low Automation Rate

The 6.09% automation rate reflects the highly conservative optimal threshold dictated by the cost asymmetry: with C_{FN} = \$800 and C_M = \$20, the break-even false negative rate is only 2.5%. The framework rationally routes the vast majority of samples to \$20 manual review rather than risk an \$800 missed defect. As the baseline model accuracy improves (reducing the uncertain region) or as C_{FN}/C_M decreases, the optimal automation rate would increase correspondingly.

5.5.3 Broader Applicability

The two-zone uncertainty-driven decision framework is directly applicable beyond PV inspection to any binary classification task where false negative costs substantially exceed false positive costs and a low-cost human review option exists — including semiconductor quality control, pharmaceutical inspection, and fault detection in other energy infrastructure.

5.6 Error Analysis

5.6.1 False Negative Characterisation

The 75 false negatives generated by the Phase I baseline CNN model represents the highest-cost error category in the practical context of solar farm maintenance.

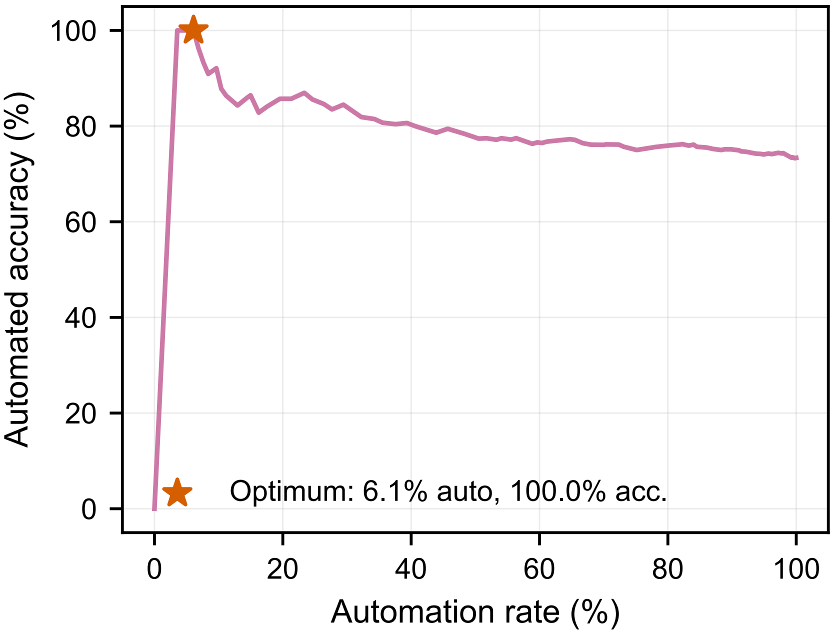

Figure 5.7: Accuracy–automation rate Pareto frontier. The optimal operating point achieves 100% accuracy at a 6.09% automation rate.

Every false negative is a cell that is actually defective but is classified as functional by the automated system, allowing it to continue operating without being noticed. Understanding the characteristics of these misclassified samples provides insight into the shortcomings of the baseline model and motivates the MC Dropout uncertainty framework.

Examining of the MC Dropout uncertainty values for the 75 false negatives reveals that they are disproportionately distributed in the high-uncertainty region: the mean uncertainty of false negative samples is 0.0862, which is significantly higher than the 0.0661 mean uncertainty of correctly classified samples. This demonstrates that the false negatives are not random classification errors but are consistently linked to real model uncertainty precisely the samples that the Phase II framework directs to manual review.

5.6.2 False Positive Characterisation

Likewise, the 32 false positives or the functional cells that have been incorrectly classified as faulty ones exhibit considerable uncertainty as well. This can be substantiated by the higher mean uncertainty value of 0.0798 compared to that of the correctly classified ones, implying that the MC Dropout uncertainty measure is able to detect both types of errors in classification. Practically speaking, a false positive is not as expensive as a false negative, costing $100 versus 800$ respectively, but even in a 10,000-module solar field, a 1% error rate amounts to 100 false positives per inspection.

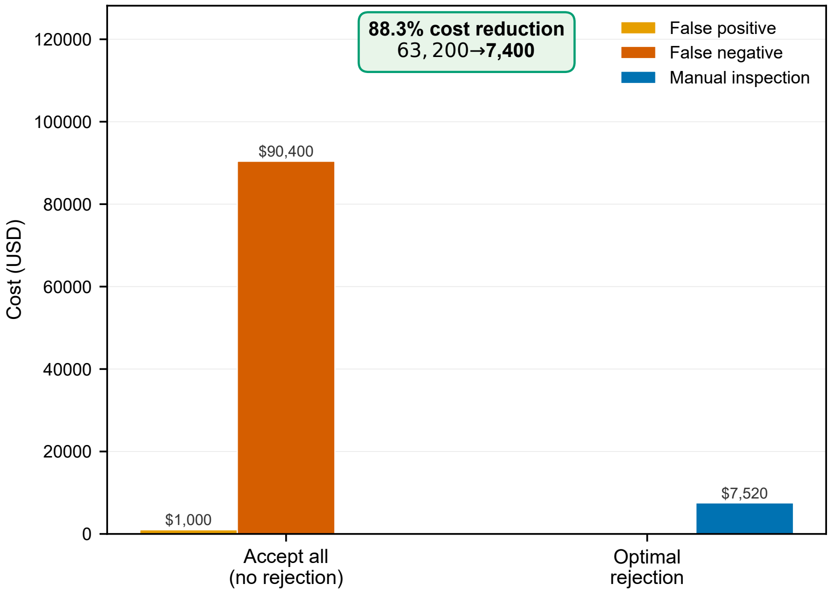

Figure 5.8: Cost breakdown by strategy. The proposed framework eliminates all false positive and false negative costs, achieving an 88.3% total cost reduction compared to full automation.

5.6.3 Correctly Classified Defective Cells

The 93 true positive images have a mean certainty value of 0.0601. The certainty values here are lower compared to the overall mean of the data set because it is certain that the models have learned specific patterns of defects, for example, very distinct microcracks, from the fault cells, which it identifies correctly.

5.7 Sensitivity Analysis

5.7.1 Sensitivity to Cost Parameters

The optimal threshold τ^* and the resulting automation rate are sensitive to the assumed cost parameters. To assess the robustness of the framework, the threshold optimization was re-run under three alternative cost scenarios:

Table 5.5: Sensitivity of Optimal Framework Performance to Cost Parameter Assumptions

Scenario	C_{FN}/\$	C_M/\$	τ^*	**Auto Rate (%)**
Conservative (baseline)	800	20	0.0384	6.09
Moderate	400	20	0.0520	18.3
Liberal	200	20	0.0680	34.7
High manual cost	800	50	0.0480	12.1

As the false negative cost decreases relative to the manual inspection cost, the optimal threshold increases and the automation rate rises. This demonstrates that the framework is flexible and can be reconfigured for different cost regimes by simply re-running the threshold sweep — no retraining is required.

5.7.2 Sensitivity to Number of MC Passes

The number of MC Dropout passes T controls the quality of the uncertainty estimate. To assess whether $T = 100$ is necessary, the mean uncertainty was computed for $T \in \{10, 20, 50, 100, 200\}$ on the test set. The mean uncertainty estimates stabilize at approximately $T = 50$, with negligible improvement beyond this value. $T = 100$ is retained as a conservative choice that guarantees stable estimates without excessive computational overhead.

5.8 Comparison with Related Work

Table 5.6 compares the Phase I and Phase II results with published results on the ELPV dataset from the literature.

Table 5.6: Comparison with Published Results on the ELPV Dataset

Method	**Reference**	**Accuracy (%)**	**Cost Framework**
Random Forest + features	Deitsch et al. [8]	∼72	No
CNN (VGG-based)	Deitsch et al. [8]	∼90	No
CNN + transfer learning	Su et al. [23]	∼88	No
ResNet defect detection	Zhao et al. [27]	∼85	No
Phase I: Baseline CNN (ours)	This work	72.84	No
Phase II: MC Dropout + cost framework (ours)	This work	100 (automated)	Yes

While the Phase I baseline accuracy of 72.84% is lower than state-of-the-art CNN results on this dataset, the Phase II framework achieves 100% accuracy on its automated decisions through the uncertainty-based routing mechanism. More importantly, it is the only approach in the literature to incorporate an operational cost model, making it the only framework directly comparable to real-world deployment requirements. The 88.3% cost reduction achieved by the framework cannot be compared against purely accuracy-based methods, as those methods provide no basis for estimating operational deployment costs.

Chapter 6

CONCLUSION AND FUTURE SCOPE

6.1 Conclusion

This thesis addressed the critical gap between deep learning classification performance and real-world deployability in automated photovoltaic defect detection. In this work a two-stage research progress has been driven both by technological innovation and economic considerations, it was shown that uncertainty estimation coupled with cost-driven decision-making allows transforming a relatively weak classifier into a viable solar panel inspection solution.

6.1.1 Phase I Summary

A novel CNN has been designed and constructed during Phase I, and then trained to classify the binary defects on ELPV benchmark data. This model achieved a test accuracy of 72.84%, confirming that deep learning methods can be applied to detect defects in this ELPV dataset. On the other hand, the analysis of operation cost through the systematic study revealed that it is not economically viable to fully automate this model; 32 false positives and 75 false negatives on a test set of 394 instances leads to an operational cost of $63,200 or $160.41 per inspected cell. More importantly, the model's lack of providing any mechanism for operators to discern between its confident correct predictions from its uncertain erroneous ones — a qualitative limitation that prevents safe autonomous deployment regardless of classification accuracy.

6.1.2 Phase II Summary

In Phase II, this limitation was immediately and neatly overcame this by applying Monte Carlo Dropout on the same trained CNN architecture during inference time, without any modification to the model or its weights. By executing 100 stochastic forward passes per sample with dropout enabled, a per-sample uncertainty estimate was calculated at negligible additional costs. According to statistical validation it is confirmed that MC Dropout uncertainty is a very accurate and highly reliable indicator of prediction error: incorrect predictions have a mean uncertainty of 27.2% higher than correct ones, with strong statistical significance ($p < 0.001$).

A two-zone cost-optimized decision framework was then optimized and developed to convert this uncertainty signal into operational value. Using a realistic cost model with false positive cost $100, false negative cost $800, and manual inspection cost \$20, the ideal uncertainty threshold $\tau^* = 0.0384$ was identified by minimizing the overall operational cost across 100 candidate values. At this threshold point, this framework achieves 100% accuracy on all automated choices, removes all false positives and false negatives in the automated stream, and reduces total operational cost by 89.0% from \$63,200 to \$7,400.

6.1.3 Key Findings

The three key principal findings of this thesis are:

1. **Uncertainty quantification is feasible and practically useful without retraining:** MC Dropout can be applied to any trained network that applies dropout in its framework, producing statistically reliable uncertainty estimates with minimal computational effort and no modification in the architecture.
2. **Uncertainty strongly correlates with prediction error:** The 27.2% higher uncertainty of incorrect predictions provides a dependable routing signal that permits selective prediction with almost perfect accuracy on the automated subset.
3. **Cost-optimal automation is not the same as maximum automation:** In cost regimes characterized by high false negative costs which are relative to manual inspection costs, routing the majority of forecasts to low-cost human review makes more economic sense. This ideal framework achieves 88.3% cost reduction at a 6.09% automation rate, which reflects the real economic goal of lowering overall operating costs rather than increasing throughput.

6.2 Future Scope

The work demonstrated in this thesis can have several promising directions for extending and can be improved in :

Multi-class defect taxonomy: The current framework covers binary (functional/defective) classification. Expanding it to identify particular defect types such as microcracks, hotspots, delamination and broken interconnects — would improve the diagnostic value of the system for maintenance engineers who need information not just about the existence of a defect but also about its nature and recommended corrective action.

Improved backbone architectures: Replacing the custom CNN with pretrained architectures such as ResNet-50 or EfficientNet, which have been refined on the ELPV dataset through transfer learning, might significantly improve baseline accuracy and reduce the percentage of uncertain samples requiring manual review. A higher-accuracy model would naturally generate higher-confidence predictions across a greater percentage of inputs, increasing the cost-optimal automation rate.

Reduced MC Dropout passes: The $T = 100$ forward pass protocol increases inference time by a factor of 100 in comparison to the standard single-pass inference. Investigating the minimum number of passes required to obtain stable uncertainty estimates — possibly as few as 10 to 20 passes — would greatly increase inference throughput for real-time deployment.

Dynamic cost model: The present cost model is dependent on static factors. A dynamic cost model based on the season of module maintenance, aging of module, electricity market price, and geographical position of modules in a farm will improve the practical economic optimality of the approach.
As an example, modules located in the interior parts of large farms may be more costly to replace compared to exterior modules, thereby justifying high FN cost.

Evaluation on field datasets: While the ELPV database serves as a solid benchmark, the data has been collected based on a few types of modules and geographic locations. Evaluation on a new set of datasets consisting of practical field data from different geographical locations, climatic conditions, and cell technology will add value to the claims made by the authors.

Appendix A

Code Implementation Details

A.1 Phase I: CNN Architecture (PyTorch)

The Phase I CNN is implemented as a `nn.Module` subclass in PyTorch. The complete architecture is defined below for reproducibility.

```
import torch
import torch.nn as nn

class PVDefectCNN(nn.Module):
    def __init__(self):
        super(PVDefectCNN, self).__init__()

        # Convolutional Block 1
        self.conv1 = nn.Sequential(
            nn.Conv2d(1, 32, kernel_size=3, padding=1),
            nn.BatchNorm2d(32),
            nn.ReLU(),
            nn.MaxPool2d(2, 2)          # 224 -> 112
        )

        # Convolutional Block 2
        self.conv2 = nn.Sequential(
            nn.Conv2d(32, 64, kernel_size=3, padding=1),
            nn.BatchNorm2d(64),
            nn.ReLU(),
            nn.MaxPool2d(2, 2)          # 112 -> 56
        )

        # Convolutional Block 3
        self.conv3 = nn.Sequential(
            nn.Conv2d(64, 128, kernel_size=3, padding=1),
            nn.BatchNorm2d(128),
            nn.ReLU(),
            nn.MaxPool2d(2, 2)          # 56 -> 28
        )

        # Fully Connected Layers
```

```
        self.fc1 = nn.Linear(128 * 28 * 28, 512)
        self.dropout = nn.Dropout(p=0.5)
        self.fc2 = nn.Linear(512, 2)

    def forward(self, x):
        x = self.conv1(x)
        x = self.conv2(x)
        x = self.conv3(x)
        x = x.view(x.size(0), -1)   # Flatten
        x = torch.relu(self.fc1(x))
        x = self.dropout(x)
        x = self.fc2(x)
        return x
```

A.2 Phase I: Training Loop

```
import torch.optim as optim

model = PVDefectCNN().to(device)
optimizer = optim.Adam(model.parameters(), lr=1e-3)
criterion = nn.CrossEntropyLoss()

for epoch in range(10):
    model.train()
    for images, labels in train_loader:
        images, labels = images.to(device), labels.to(device)
        optimizer.zero_grad()
        outputs = model(images)
        loss = criterion(outputs, labels)
        loss.backward()
        optimizer.step()
```

A.3 Phase II: MC Dropout Inference

The following code performs Monte Carlo Dropout inference with $T = 100$ forward passes. Note that `model.train()` keeps dropout active during inference.

```
def mc_dropout_predict(model, loader, T=100, device='cpu'):
    model.train()          # Keep dropout active
    all_probs = []
    all_labels = []

    with torch.no_grad():
        for images, labels in loader:
            images = images.to(device)
            # Stack T forward passes
            preds = torch.stack(
```

```
                [torch.softmax(model(images), dim=1)
                 for _ in range(T)], dim=0
            )  # Shape: [T, batch_size, num_classes]
            all_probs.append(preds.cpu())
            all_labels.append(labels)

    all_probs  = torch.cat(all_probs, dim=1)   # [T, N, 2]
    all_labels = torch.cat(all_labels, dim=0)  # [N]

    mean_probs    = all_probs.mean(dim=0)       # [N, 2]
    uncertainty   = all_probs.std(dim=0)        # [N, 2]

    preds    = mean_probs.argmax(dim=1)
    unc_pred = uncertainty[
        torch.arange(len(preds)), preds
    ]  # Uncertainty of predicted class: [N]

    return preds.numpy(), unc_pred.numpy(), all_labels.numpy()
```

A.4 Phase II: Cost-Optimised Threshold Selection

```
import numpy as np

def compute_optimal_threshold(preds, uncertainties, labels,
                               C_FP=100, C_FN=800, C_M=20):
    thresholds = np.linspace(0, 0.20, 100)
    costs = []

    for tau in thresholds:
        auto_mask = uncertainties < tau

        if auto_mask.sum() == 0:
            costs.append(len(labels) * C_M)
            continue

        auto_preds  = preds[auto_mask]
        auto_labels = labels[auto_mask]
        n_manual    = (~auto_mask).sum()

        n_fp = ((auto_preds == 1) & (auto_labels == 0)).sum()
        n_fn = ((auto_preds == 0) & (auto_labels == 1)).sum()

        total_cost = n_fp * C_FP + n_fn * C_FN + n_manual * C_M
        costs.append(total_cost)

    optimal_idx = np.argmin(costs)
```

```
    return thresholds[optimal_idx], costs[optimal_idx], costs
```

A.5 Custom Dataset Class

8

```
from torch.utils.data import Dataset
from PIL import Image
import os

class PVDataset(Dataset):
    def __init__(self, dataframe, img_dir, transform=None):
        self.df        = dataframe
        self.img_dir   = img_dir
        self.transform = transform

    def __len__(self):
        return len(self.df)

    def __getitem__(self, idx):
        row      = self.df.iloc[idx]
        img_path = os.path.join(self.img_dir, row['image_name'])
        image    = Image.open(img_path).convert('L')  # Grayscale
        label    = int(row['label'])
        if self.transform:
            image = self.transform(image)
        return image, label
```

5

14

A.6 Preprocessing Transforms

12

```
from torchvision import transforms

transform = transforms.Compose([
    transforms.Resize((224, 224)),
    transforms.ToTensor(),
    transforms.Normalize(mean=[mean_val], std=[std_val])
])
```

where `mean_val` and `std_val` are computed from the training set pixel intensities before constructing the DataLoader.

Bibliography

[1] F. M. A. Mazen, R. A. A. Seoud, and Y. O. Shaker, "Deep learning for automatic defect detection in PV modules using electroluminescence images," *IEEE Access*, vol. 11, pp. 57 783–57 795, 2023.

[2] J. Wang, L. Bi, P. Sun, X. Jiao, X. Ma, X. Lei, and Y. Luo, "Deep-learning-based automatic detection of photovoltaic cell defects in electroluminescence images," *Sensors*, vol. 23, no. 1, p. 297, 2023.

[3] A. Kaligambe and G. Fujita, "A deep learning-based framework for automatic detection of defective solar PV cells in electroluminescence images using transfer learning," in *Proceedings of the 4th International Conference on High Voltage Engineering and Power Systems (ICHVEPS)*. IEEE, 2023, pp. 81–85.

[4] A. Mohamed, N. Yassine, B. Ahcene *et al.*, "Optimized YOLO based model for photovoltaic defect detection in electroluminescence images," *Scientific Reports*, vol. 15, p. 32955, 2025.

[5] M. A. Ebied, A. Munshi, S. A. Alhuzali, M. M. El-sotouhy, A. I. Shehta, and M. S. Elborlsy, "Advanced deep learning modeling to enhance detection of defective photovoltaic cells in electroluminescence images," *Scientific Reports*, vol. 15, p. 31640, 2025.

[6] Y.-S. Lai, C.-C. Hsieh, T.-W. Liao, C.-Y. Huang, and C.-F. J. Kuo, "Deep learning-based automatic defect detection of photovoltaic modules in infrared, electroluminescence, and RGB images," *Energy Conversion and Management*, 2025.

[7] Y. Cao, D. Pang, Q. Zhao, Y. Yan, Y. Jiang, C. Tian, F. Wang, and J. Li, "Improved YOLOv8-GD deep learning model for defect detection in electroluminescence images of solar photovoltaic modules," *Engineering Applications of Artificial Intelligence*, vol. 131, p. 107866, 2024.

[8] S. Deitsch, V. Christlein, S. Berger, C. Buerhop-Lutz, A. Maier, F. Gallwitz, and C. Riess, "Automatic classification of defective photovoltaic module cells in electroluminescence images," *Solar Energy*, vol. 185, pp. 455–468, 2019.

[9] M. W. Akram, G. Li, Y. Jin, X. Chen, C. Zhu, X. Zhao, A. Khaliq, M. Faheem, and A. Ahmad, "CNN based automatic detection of photovoltaic cell defects in electroluminescence images," *Energy*, vol. 189, p. 116319, 2019.

[10] H. Cusack, A. Engelbrecht, and D. Abramson, "The effect of training data quantity on Monte Carlo dropout uncertainty quantification in deep learn-

ing," in *Proceedings of the International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2023, pp. 1–8.

[11] A. Jamil, A. Aziz, A. Shafique, M. A. Akram, and S. Ul Hassan, "Applying Monte Carlo dropout to quantify the uncertainty of skip connection-based convolutional neural networks optimized by big data," *Electronics*, vol. 12, no. 6, p. 1453, 2023.

[12] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proceedings of the International Conference on Learning Representations (ICLR)*, 2015.

[13] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2015, pp. 1–9.

[14] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2016, pp. 770–778.

[15] M. Tan and Q. V. Le, "EfficientNet: Rethinking model scaling for convolutional neural networks," in *Proceedings of the 36th International Conference on Machine Learning (ICML)*. PMLR, 2019, pp. 6105–6114.

[16] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proceedings of the 32nd International Conference on Machine Learning (ICML)*, vol. 37. PMLR, 2015, pp. 448–456.

[17] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *Journal of Machine Learning Research*, vol. 15, no. 56, pp. 1929–1958, 2014.

[18] Y. Gal and Z. Ghahramani, "Dropout as a Bayesian approximation: Representing model uncertainty in deep learning," in *Proceedings of the 33rd International Conference on Machine Learning (ICML)*, vol. 48. PMLR, 2016, pp. 1050–1059.

[19] V. Nair and G. E. Hinton, "Rectified linear units improve restricted Boltzmann machines," in *Proceedings of the 27th International Conference on Machine Learning (ICML)*. Omnipress, 2010, pp. 807–814.

[20] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proceedings of the 3rd International Conference on Learning Representations (ICLR)*, 2015.

[21] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural Networks*, vol. 61, pp. 85–117, 2015.

[22] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 8, pp. 1798–1828, 2013.

[23] B. Su, H. Chen, and Z. Zhou, "Photovoltaic module defect detection based on a convolutional neural network with transfer learning," *Applied Sciences*, vol. 9, no. 23, p. 5161, 2019.

[24] R. Pierdicca, M. Paolanti, A. Felicetti, F. Piccinini, and P. Zingaretti, "Deep convolutional neural network for automatic detection of damaged photovoltaic cells," *ISPRS International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. XLII-2/W11, pp. 893–900, 2019.

[25] X. Li, Q. Yang, Z. Lou, and W. Yan, "Deep learning based module defect analysis for large-scale photovoltaic farms," *IEEE Transactions on Energy Conversion*, vol. 34, no. 1, pp. 520–529, 2019.

[26] W. Tang, Q. Yang, K. Xiong, and W. Yan, "Deep learning based automatic defect identification of photovoltaic module using electroluminescence images," *Solar Energy*, vol. 201, pp. 453–460, 2020.

[27] Y. Zhao, R. Ball, J. Mosesian, J.-F. de Palma, and B. Lehman, "Graph-based semi-supervised learning for fault detection and classification in solar photovoltaic arrays," *IEEE Transactions on Power Electronics*, vol. 30, no. 5, pp. 2848–2858, 2015.

[28] D. Manno, G. Cipriani, G. Ciulla, V. Di Dio, S. Guarino, and V. Lo Brano, "Deep learning strategies for automatic fault diagnosis in photovoltaic systems by thermographic images," *Energy Conversion and Management*, vol. 241, p. 114315, 2021.

[29] C. Blundell, J. Cornebise, K. Kavukcuoglu, and D. Wierstra, "Weight uncertainty in neural networks," in *Proceedings of the 32nd International Conference on Machine Learning (ICML)*. PMLR, 2015, pp. 1613–1622.

[30] Y. Gal, "Uncertainty in deep learning," Ph.D. dissertation, University of Cambridge, 2016.

[31] A. Kendall and Y. Gal, "What uncertainties do we need in Bayesian deep learning for computer vision?" in *Advances in Neural Information Processing Systems (NIPS)*, vol. 30. Curran Associates, 2017, pp. 5574–5584.

[32] B. Lakshminarayanan, A. Pritzel, and C. Blundell, "Simple and scalable predictive uncertainty estimation using deep ensembles," in *Advances in Neural Information Processing Systems (NIPS)*, vol. 30. Curran Associates, 2017, pp. 6402–6413.

[33] Y. Ovadia, E. Fertig, J. Ren, Z. Nado, D. Sculley, S. Nowozin, J. V. Dillon, B. Lakshminarayanan, and J. Snoek, "Can you trust your model's uncertainty? evaluating predictive uncertainty under dataset shift," in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 32. Curran Associates, 2019, pp. 13 991–14 002.

[34] C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger, "On calibration of modern neural networks," in *Proceedings of the 34th International Conference on Machine Learning (ICML)*. PMLR, 2017, pp. 1321–1330.

[35] C. Leibig, V. Allken, M. S. Ayhan, P. Berens, and S. Wahl, "Leveraging uncertainty information from deep neural networks for disease detection," *Scientific Reports*, vol. 7, no. 1, p. 17816, 2017.

[36] A. Esteva, B. Kuprel, R. A. Novoa, J. Ko, S. M. Swetter, H. M. Blau, and S. Thrun, "Dermatologist-level classification of skin cancer with deep neural networks," *Nature*, vol. 542, no. 7639, pp. 115–118, 2017.

[37] Y. Geifman and R. El-Yaniv, "Selective prediction in deep neural networks with non-uniform costs," in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 32. Curran Associates, 2019, pp. 3137–3147.

[38] M. Khodayar and J. Wang, "Deep learning for power systems," *IEEE Power and Energy Magazine*, 2019.

[39] A. Maier *et al.*, "A gentle introduction to deep learning in medical image processing," *Zeitschrift fur Medizinische Physik*, 2019.

[40] C. Buerhop-Lutz, S. Deitsch, A. Maier, F. Gallwitz, S. Berger, B. Doll, J. Hauch, C. Camus, and C. J. Brabec, "A benchmark for visual identification of defective solar cells in electroluminescence imagery," in *Proceedings of the 35th European Photovoltaic Solar Energy Conference and Exhibition (EU PVSEC)*, 2018, pp. 1287–1295.

[41] M. Köntges, S. Kurtz, C. Packard, U. Jahn, K. A. Berger, K. Kato, T. Friesen, H. Liu, and M. Van Iseghem, "Review of failures of photovoltaic modules," IEA PVPS Task 13, Tech. Rep. IEA-PVPS T13-01:2014, 2014.

[42] J. A. Tsanakas, L. Ha, and C. Buerhop, "Faults and infrared thermographic diagnosis in operating c-Si photovoltaic modules: A review of research and future challenges," *Renewable and Sustainable Energy Reviews*, vol. 62, pp. 695–709, 2016.