# ABSTRACT

Solar energy has emerged as a critical renewable energy source, driving the adoption of photovoltaic (PV) technologies, including land-based and floating systems. The widespread deployment of these systems necessitates effective monitoring and maintenance strategies to ensure optimal performance and longevity.

The research addresses the need for intelligent inspection systems capable of identifying various defects, including hot spots, cell cracks, diode failures, soiling, vegetation blocking, shadowing, and offline modules. This research aims to develop and evaluate deep learning models for automated anomaly detection in solar photovoltaic systems using thermal imaging.

Using the publicly available InfraredSolarModules dataset, which contains over 20,000 thermal images representing 11 different anomaly types, three state-of-the-art architectures were evaluated: Vision Transformer (ViT-B32), Data-Efficient Image Transformer (DeiT-B16), and ConvNeXt-Tiny.

The methodology encompasses a comprehensive evaluation across three classification scenarios: binary classification (anomaly vs. normal), 11-class classification (anomaly types only), and 12-class classification (normal and anomaly types). Image processing and offline data augmentation techniques were applied for anomaly enhancement and class balancing tasks for multi-class classification scenarios.

The results demonstrate that DeiT-B16 achieved the highest overall performance with 94.92% accuracy for 12-class classification and 94.37% for 11-class classification, while ConvNeXt-Tiny showed excellent efficiency with 99.05% accuracy for binary classification, 94.33% for 12-class classification, and 94.16% for 11-class classification using 67% fewer parameters than transformer models. ViT-B32 delivered competitive performance with the fastest training speed (1.7 min/epoch). DeiT-B16 represents the optimal choice for accuracy-critical applications, while ConvNeXt-Tiny offers superior parameter efficiency for resource-constrained deployments.

This research contributes to the advancement of automated solar panel inspection systems and provides a foundation for implementing intelligent maintenance strategies for photovoltaic systems.

# SAMMENDRAG

Solenergi har blitt en sentral fornybar energikilde og driver implementeringen av solcelleanlegg (PV-teknologier), inkludert både bakkemonterte og flytende systemer. Den utstrakte utplasseringen av slike systemer krever effektive overvåkings- og vedlikeholdsstrategier for å sikre optimal ytelse og lang levetid.

Dette forskningsarbeidet adresserer behovet for intelligente inspeksjonssystemer som kan identifisere ulike defekter, inkludert varme punkter (hot spots), cellebrudd, diodefeil, tilsmussing, vegetasjonsblokkering, skyggeeffekter og frakoblede moduler. Målet er å utvikle og evaluere dyp læringsmodeller for automatisk anomalioppdagelse i solcelleanlegg ved hjelp av termiske bilder.

Ved å benytte det offentlig tilgjengelige datasettet InfraredSolarModules, som inneholder over 20,000 termiske bilder med 11 forskjellige anomalityper, ble tre moderne arkitekturer evaluert: Vision Transformer (ViT-B32), Data-Efficient Image Transformer (DeiT-B16) og ConvNeXt-Tiny.

Metoden inkluderer en omfattende evaluering innenfor tre klassifiseringsscenarier: binær klassifisering (anomalier vs. normale), 11-klasses klassifisering (kun anomalier), og 12-klasses klassifisering (normale og anomalityper). Bildebehandling og offline dataforsterkning ble benyttet for å forbedre anomalier og balansere klasser i flerklasses scenarier.

Resultatene viser at DeiT-B16 oppnådde den høyeste totale nøyaktigheten med 94.92% for 12-klasses klassifisering og 94.37% for 11-klasses klassifisering. Samtidig viste ConvNeXt-Tiny høy effektivitet med 99.05% nøyaktighet for binær klassifisering, 94.33% for 12-klasses og 94.16% for 11-klasses klassifisering, med 67% færre parametere enn transformer-modellene. ViT-B32 leverte konkurransedyktig ytelse med den raskeste treningstiden (1.7 minutter per epoke). DeiT-B16 er dermed det optimale valget for applikasjoner der nøyaktighet er kritisk, mens ConvNeXt-Tiny er best egnet for ressursbegrensede implementeringer.

Denne studien bidrar til videreutviklingen av automatiserte inspeksjonssystemer for solcellepaneler og legger et grunnlag for implementering av intelligente vedlikeholdsstrategier for PV-systemer.

# PREFACE

This thesis represents the culmination of my master's degree journey in ICT-Simulation and Visualization at NTNU, and I am deeply grateful for the opportunity to explore this fascinating field that bridges technology and human understanding.

First of all, I want to thank my supervisor, Assoc. Prof. Saleh Abdel-Afou Alaliyat, for all his help and knowledge that made this research possible. His support during difficult times and his way of making things clear when I was confused have been very important for my learning.

I am equally grateful to my co-supervisor, Sr. Researcher and Scientist Mohammadreza Aghaei, whose deep knowledge and practical insights in this domain have significantly enriched this work. His constructive feedback had pushed me to think beyond conventional boundaries and explore new possibilities.

My sincere thanks also go to my colleagues in the Department of ICT, who have created an intellectually stimulating environment. I want to acknowledge my family especially my mom and siblings, and friends, whose unwavering support has been my instrumental throughout this demanding yet rewarding process. Their belief in my abilities, even when I doubted myself, has been a constant source of motivation and strength.

This thesis would not have been possible without the collective support of all these wonderful people, and I am truly fortunate to have had them by my side during this whole experience.

# CONTENTS

# LIST OF FIGURES

# LIST OF TABLES

# ABBREVIATIONS

List of all abbreviations in alphabetical order.

- **ANN** Artificial Neural Network
- **CNN** Convolution Neural Network
- **DeiT** Data-efficient Image Transformer
- **DL** Deep Learning
- **FF** Feed Forward
- **FPV** Floating Photovoltaic System
- **ML** Machine Learning
- **MLP** Multi-Layer Perceptron
- **NLP** Natural Language Processing
- **PV** Photovoltaic Systems
- **UAV** Unmanned Aerial Vehicle
- **ViT** Vision Transformer

CHAPTER

# ONE

# INTRODUCTION

This chapter provides an overview of the research topic, including the motivation, scope, objectives, and significance of the study. It also outlines the research questions and the thesis structure.

## 1.1 Motivation

The growing global emphasis on renewable energy has placed solar photovoltaic (PV) systems at the forefront of sustainable power generation. As solar adoption surges, ensuring the efficiency and reliability of PV installations has become increasingly important to meet energy security and climate goals. However, operational challenges, particularly faults and anomalies in PV modules, can significantly degrade system performance if not detected and addressed promptly.

Machine learning (ML) and deep learning (DL) techniques offer promising solutions for automating the detection of such anomalies. These methods have demonstrated high effectiveness in fields like medical imaging and autonomous driving by identifying complex patterns that may be difficult for humans to detect. In the PV domain, ML and DL models can be trained on large datasets of thermal or optical images to recognize diverse fault signatures, including hotspots, microcracks, soiling, and delamination.

According to the International Energy Agency (IEA), global renewable electricity capacity additions reached a record 507 GW in 2023, with solar PV accounting for nearly 75% of that growth [1]. By the end of 2023, installed PV capacity surpassed 1000 GW, and this trend is expected to continue, driven by projections of over 5500 GW in new renewable capacity between 2024 and 2030, 80% of which is anticipated to come from solar PV [2].

Despite this growth, traditional manual inspection methods remain prevalent. These approaches are often time consuming, labor intensive, and prone to human error. In contrast, computer vision based automated systems offer real-time monitoring, faster fault localization, and improved scalability. Nevertheless, deploying effective deep learning models in this context remains challenging due to class imbalance, data scarcity, and the complexity of multi-class classification tasks.

This research is motivated by the need to overcome these challenges and to contribute to the development of more accurate, efficient, and generalizable anomaly

detection systems for solar PV installations.

## 1.2 Objectives

Land-based and floating solar PV systems are increasingly deployed to meet global energy demands sustainably. However, the efficiency and reliability of these systems can be compromised by various anomalies, such as physical defects, soiling, and electrical faults. This research aims to develop a comprehensive framework for automated anomaly detection in solar PV systems using machine learning and computer vision techniques. The primary objectives of this research are:

- To explore and implement various data preprocessing techniques, including image augmentation, to enhance the performance of machine learning and deep learning models for solar PV anomaly detection.
- To evaluate and compare the performance of different machine learning and deep learning architectures in detecting and classifying specific types of anomalies in solar PV systems.
- To develop methodological frameworks that address challenges such as class imbalance and data scarcity in computer vision-based solar PV anomaly detection systems.

## 1.3 Research Questions

This thesis aims to advance the field of automated anomaly detection in solar photovoltaic systems through the application of machine learning and computer vision techniques. Specifically, the research addresses the following questions:

- **Research Question 1:** How do various data preprocessing techniques, particularly image augmentation, influence the performance of deep learning models for solar PV anomaly detection?
- **Research Question 2:** Which deep learning (ViT and CNN variants) architectures demonstrate superior performance metrics (accuracy, precision, recall, F1-score) for detecting and classifying specific types of anomalies in solar PV systems?
- **Research Question 3:** What methodological frameworks and technical approaches best address the challenges of class imbalance, data scarcity in computer vision-based solar PV anomaly detection systems?

## 1.4 Significance of the Study

The significance of this study lies in its potential to enhance the efficiency and reliability of solar PV systems through advanced anomaly detection techniques. By leveraging machine learning and computer vision, this research aims to provide a robust framework for fault detection, which can lead to reduced maintenance

costs, increased energy yield, and extended PV system lifespan. This study contributes to the broader field of renewable energy by addressing the critical need for automated, reliable, and efficient monitoring solutions in solar PV installations. The findings are expected to have practical implications for solar energy operators, policymakers, and researchers, facilitating the wider adoption of solar technologies and supporting global efforts towards sustainable energy transition.

## 1.5 Scope

The scope of this research is to investigate the use of ML and DL techniques for the detection of anomalies in solar PVs. The focus will be on the development of algorithms that can be used to automatically detect and classify anomalies in solar PVs and the evaluation of these algorithms using real-world Solar PV IR data. The research will also investigate the use of data pre-processing and data augmentation techniques to improve the performance of the deep learning algorithms. We will also investigate the use of different deep learning architectures, such as Vision Transformers (ViT) and transformer-inspired Convolution Neural Networks (CNNs), to improve the performance of the anomaly detection tasks.

## 1.6 Thesis Structure

The remainder of this thesis is organized as follows: Chapter 2 reviews existing literature on solar PV anomaly detection, deep learning techniques, and computer vision applications in renewable energy systems. It also presents the theoretical foundations underlying this research, including photovoltaic theory, thermal physics, machine learning theory, and deep learning concepts. Chapter 3 describes the research design, including data collection, preprocessing, data augmentation techniques, model development, and evaluation metrics used in this study. Chapter 4 presents the experimental results, model performance metrics, and comparative analysis of different deep learning architectures. Chapter 5 summarizes the key findings, contributions of the research, and suggests directions for future work. The References section contains the bibliography and cited sources.

CHAPTER

# TWO

# THEORY AND LITERATURE REVIEW

This chapter provides a comprehensive overview of the theoretical background of photovoltaic (PV) systems. It covers fundamental concepts, anomaly types, detection methods, and relevant technologies, along with a review of machine learning and deep learning basics, and a review of deep learning architectures for anomaly detection in PV systems. It also provides a review of the literature on anomaly detection in PV systems using computer vision methods.

## 2.1 Solar Photovoltaic (PV) Systems

The development of a computer vision algorithm for a specific domain requires a basic understanding of how things work. Therefore, it is worthwhile to explore how a PV system works and identify its main elements that are responsible for generating green energy. This understanding not only needs to be able to understand the anomaly types and detection methods, but also to be able to design and implement the computer vision algorithm.

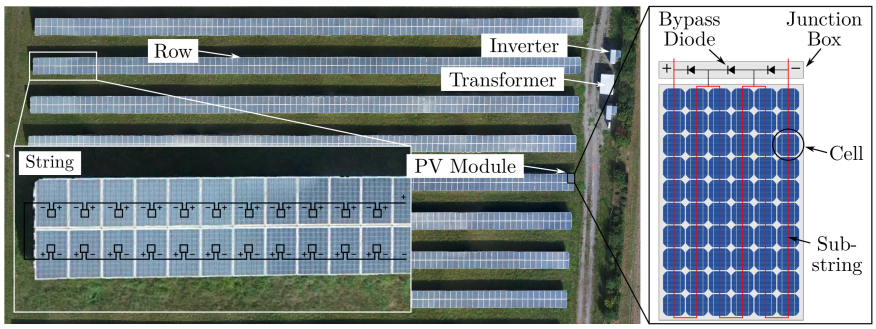



Figure 2.1: Elements of a PV system [3]

Figure 2.1 illustrates the structural hierarchy of a utility-scale land-based photovoltaic (PV) system, which is designed for large-scale energy generation and seamless grid integration. The system is modular, comprising interconnected components that transform solar irradiance into electrical power efficiently and reliably.

At the most fundamental level are solar cells, semiconductor devices based on p-n junctions that directly convert sunlight into direct current (DC) electricity. Multiple cells are electrically connected in series to form a PV module, which serves as the building block of the solar array. Each module typically includes a junction box on its rear side, facilitating electrical connections and housing protective elements. [4]

Within each module, cells are grouped into sub-strings, which are further protected by bypass diodes [4]. These diodes are connected in parallel to each sub-string and enable current to bypass shaded or defective cells, thus minimizing power losses and thermal stress caused by partial shading or mismatch [5].

Modules are connected in series to form a string, where the positive terminal of one module is linked to the negative terminal of the next. This series connection aggregates the voltage output while maintaining the current level. Multiple strings are then arranged in parallel along a row, forming the basis of the solar field layout. [4], [5].

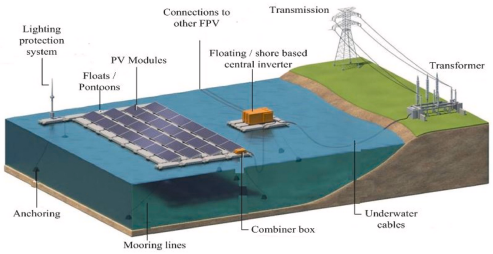
The DC output from each string is collected and routed through combiner boxes or string inverters, which convert the generated DC power into alternating current (AC). This AC power is subsequently fed into a step-up transformer that elevates the voltage to match grid transmission requirements. The transformed power is then delivered to the main utility grid [6], [7], [8].

This hierarchical architecture, from cells to modules, strings, and rows, enables effective scalability, fault isolation, and performance optimization in large PV installations. Proper integration of components such as bypass diodes, junction boxes, and inverters plays a crucial role in ensuring system reliability, energy yield, and maintainability.

### 2.1.1 Land-based Photovoltaic (LPV) Systems

Land-based photovoltaic (LPV) systems are solar energy installations deployed on terrestrial surfaces, including ground-mounted arrays and rooftop systems. As shown in Figure 2.2a, these systems convert sunlight directly into electricity through the photovoltaic effect, utilizing semiconductor materials, most commonly silicon-based technologies. Land-based systems remain the most widespread form of solar deployment globally, due to their mature infrastructure and scalability [5], [9].

(a) Land-based PV System




(b) Floating PV system on water bodies

Figure 2.2: A typical land-based and floating PV system, Source: [9], [10], [11]

### 2.1.2 Floating Photovoltaic (FPV) Systems

Floating photovoltaic (FPV) systems represent an innovative approach to solar energy deployment on water bodies such as reservoirs, lakes, and coastal areas. These systems consist of solar panels mounted on buoyant platforms that ensure stability and optimal tilt angles. As shown in Figure 2.2b, FPV installations integrate components such as anchoring systems, mooring lines, floating inverters, and underwater cables to deliver efficient energy generation and grid connectivity. Research indicates that FPV systems often exhibit higher energy yield and improved land-use efficiency compared to traditional land-based installations [10], [11].

## 2.2 PV Plant Anomalies

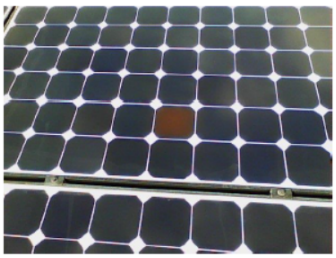
Photovoltaic (PV) systems are subject to a variety of anomalies that can significantly degrade energy output and operational reliability. These anomalies may arise from environmental factors, manufacturing defects, degradation over time, or installation issues. [12], [13].

Understanding the various types of anomalies that can affect PV systems is crucial for effective monitoring and maintenance strategies. These anomalies can significantly impact system performance, efficiency, and longevity. This section provides an overview of the different types of anomalies that can affect PV systems, as well as their physical causes and thermal patterns.

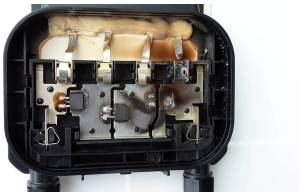
### 2.2.1 Visible Patterns

(a) Soiling

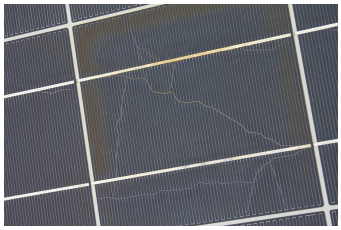
(b) Shading



(c) Hotspot



(d) Diode



(e) Cracking

(f) Vegetation

Figure 2.3: Real world imagery for PV plant anomalies, (a) Soiling, (b) Shadings, (c) Hotspot, (d) Defective diode, (e) Cracking, (f) Vegetation. Source: [14], [15]

Figure 2.3 shows some of the real-world images of PV system anomalies that can be found in a typical PV system. These anomalies are the most common anomalies that can be visible to the naked eye. But some of the anomalies are not visible through the naked eye and require advanced imaging techniques to detect their thermal patterns using techniques that will be discussed in section 2.3.

### 2.2.2 Thermal Patterns

Thermal patterns are the visual representation of the thermal anomalies in a PV system. They are also the most common way to detect and diagnose anomalies in a PV system nowadays. As computer vision models mostly rely on the pattern version of the images, it is more important to understand the patterns of the IR images as compared to their physical appearance counterparts. So we will discuss the anomaly types, their potential causes, and their thermal patterns in this section.

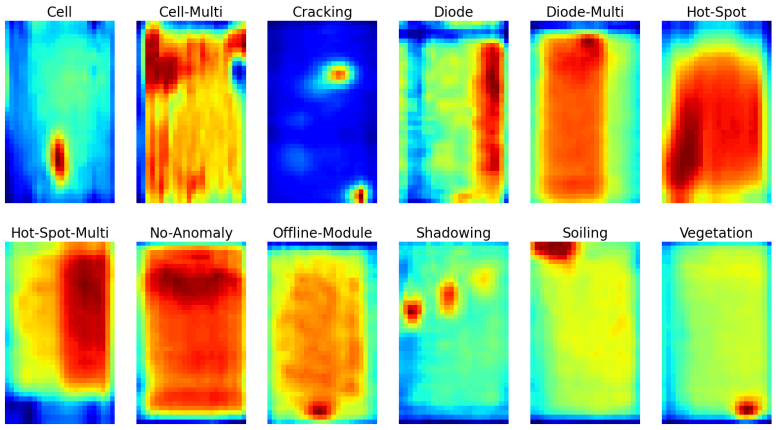



Figure 2.4: Grad CAM Visualized - Thermal Patterns of Common PV Anomalies (Source: [16])

**Cell** are localized faults within individual solar cells, typically caused by microcracks, manufacturing defects, soldering issues, or material degradation (see fig. 2.4). These faults elevate local resistance, leading to heat buildup and reduced cell efficiency [17]. In infrared (IR) thermography, they appear as small, isolated hotspots due to localized resistive losses. In electroluminescence (EL) imaging, affected regions often appear dim or inactive. Though limited in size, cell-level faults signal internal electrical degradation and may worsen over time. As illustrated in Figure 2.4, their thermal signature is distinct and sharply concentrated, supporting automated detection and classification.

**Cell-Multi** are multiple localized faults occurring across several cells within a single PV module. These typically result from widespread microcracks, soldering issues, or material degradation, leading to uneven current distribution and localized heating [17]. In infrared (IR) thermography, they appear as several small, dispersed hotspots across the module surface. Electroluminescence (EL) imaging reveals multiple dark regions, each indicating electrically inactive or disconnected cells. As shown in Figure 2.4, Cell-Multi anomalies suggest more extensive internal degradation than single-cell faults and can significantly impact module perfor-

mance. Early detection using IR or EL imaging is essential to mitigate efficiency losses.

**Cracking** refers to the formation of physical fractures in solar cells, often caused by mechanical stress during manufacturing, transportation, installation, or environmental factors such as thermal cycling, wind, or hail [18], [19]. These cracks can disrupt current flow by electrically disconnecting parts of the cell, leading to power loss and increased risk of localized heating. In EL imaging, cracks appear as dark lines or fragmented areas, while in IR thermography, they may manifest as irregular hotspots. As shown in Figure 2.4 and 2.3e, the thermal footprint varies depending on crack severity and location. Early detection through EL or IR inspection is critical to prevent accelerated degradation and hotspot formation.

**Diode** is are localized fault within individual bypass diodes in a PV module. These diodes protect substrings under shading or internal fault conditions by bypassing affected regions. 2.3d. A faulty diode can deactivate a portion of the module, typically one-third or half, depending on the diode layout. In IR thermography, this appears as large, cooler rectangular areas corresponding to inactive substrings [20]. As shown in Figure 2.4 and 2.3d, the pattern is distinct from cell-level faults. Diode anomalies are internal electrical issues that require prompt identification to maintain module output and reliability.

**Diode-Multi** anomalies occur when multiple bypass diodes are simultaneously activated or defective. This condition is commonly caused by prolonged shading, internal degradation, or permanent diode failure due to thermal or environmental stress [20]. The result is the bypassing of two or more substrings, significantly reducing module output. In IR thermography, this appears as multiple cooler zones across the module surface. As shown in Figure 2.4, the geometric segmentation of these cooler areas distinguishes them from other fault types. Early detection via thermal imaging and performance diagnostics is essential to mitigate long-term energy losses.

**Hotspot** anomalies are localized high-temperature regions typically caused by shading, soiling, manufacturing defects, or partial cell failures that induce reverse bias. These faults lead to resistive heating, material degradation, and potential safety risks [17]. As shown in Figure 2.4 and 2.3c, hotspots appear as bright thermal spots in IR imagery and visible spectrum as red hot areas. Persistent hotspots accelerate aging and may result in delamination or fire. Timely detection is crucial for effective preventive maintenance.

**Hotspot-Multi** involves the presence of multiple high-temperature points within the same module, caused by widespread defects, cell mismatch, or interconnection failures [17]. In IR thermography, this appears as several bright, thermally active zones indicating localized energy dissipation. As illustrated in Figure 2.4, these faults indicate internal stress and degradation requiring immediate attention to preserve performance and prevent further damage.

**Offline Module** refers to a PV module that is completely inactive due to wiring faults, junction box failure, or malfunctioning bypass diodes. Such modules exhibit no current flow and appear uniformly cooler than adjacent active modules in IR thermography [20]. As shown in Figure 2.4, the thermal pattern is uniform and cooler than the adjacent active modules.

**Shadowing** occurs when external obstructions like buildings, vegetation, or debris block solar radiation from reaching parts of the module [17]. This leads

to reduced irradiance and uneven energy production. In IR imagery, shadowed regions appear cooler than fully illuminated areas. As depicted in Figure 2.4, such anomalies are often geometric and align with the shape of the obstruction. Shadowing is an external fault that may activate bypass diodes or cause hotspots, making layout planning and vegetation control essential.

**Soiling** results from dust, bird droppings, or other debris accumulating on the module surface, reducing solar input. In IR thermography, soiled areas manifest as cooler zones due to reduced irradiance. As shown in Figure 2.4, these patterns are diffuse and irregular. Uneven soiling can lead to mismatch losses and trigger diode activation. Routine cleaning and environmental monitoring help prevent soiling-induced efficiency drops. [21]

**Vegetation** anomalies occur when plant growth, such as grass or tree branches, shrubs, etc., obstructs direct sunlight, partially or fully shading the module surface. These shaded zones appear cooler in IR images due to diminished irradiance. Figure 2.4 highlights the characteristic thermal pattern of vegetation-induced faults. If left unmanaged, vegetation can cause recurring energy losses and hotspot formation. Periodic trimming and site maintenance are essential for preserving performance. [22]

In addition to the above-mentioned anomalies, there are other anomalies such as glass breakage, shunted cells, etc. [21] that can be found in a PV system, which vary from land-based PV systems to floating PV systems and also different PV system technologies. But we just covered the anomalies in detail, so we will be working on this study.

## 2.3 PV Systems Inspection Methods

This section will briefly outline solar PV inspection methods, starting from traditional visual and IV curve techniques to more advanced approaches like infrared (IR) thermography and electroluminescence (EL) imaging. The focus will be on the IR inspection method only because the scope of this study is on IR imaging.

### 2.3.1 Visual Inspection

Visual inspection is the most basic method used to identify defective photovoltaic (PV) modules. It involves manually walking through the rows of a solar installation to visually detect any apparent physical defects. However, this approach is limited in scope, as many internal or electrical faults are not externally visible. Consequently, visual inspection alone is insufficient for comprehensive and reliable fault detection in PV systems.[23]

### 2.3.2 Electroluminescence (EL) Inspection

Electroluminescence (EL) inspection is a method that is often used in laboratories. In this method, the function of a solar module is reversed. A generator is used to send electricity through the module or string, and the radiation emitted from the modules is monitored with a special camera. Electroluminescence is particularly useful for detecting cracks in the cells and determining their location. Other types of defects that can be detected include PID, damaged bypass diodes, and in some

cases hotspots [24]. A major disadvantage of electroluminescence is that, to date, this method can only be performed at night or in a laboratory where daylight is excluded. [23].

### 2.3.3 IV Curve Analysis

Current-voltage (I-V) curve measurement is also a widely adopted and reliable technique for PV module inspection. It involves recording the module temperature, ambient temperature, and irradiance to generate a theoretical I-V curve. This expected curve is then compared with the measured I-V curve of the module under test. Deviations between the two can reveal a range of faults, including degradation, shading, or internal electrical issues. However, despite its diagnostic accuracy, this method is time-consuming and cannot detect mismatch losses between interconnected modules. [23], [25]

### 2.3.4 UAV-Based IR Thermal Inspection

UAVs are unmanned aerial vehicles that are used for various purposes, including aerial photography, aerial surveying, and aerial inspection. They are also used for monitoring and inspecting PV systems. Infrared thermography is a non-destructive testing technique that uses thermal imaging cameras to detect temperature variations across PV system components. This method is particularly effective for identifying thermal anomalies such as hot spots, cell failures, and electrical connection issues. [26]

- Cost-effective: UAVs are more cost-effective than traditional manned aircraft, as they do not require a pilot and can be operated remotely.
- Flexible: UAVs can be easily deployed and operated in various environments, including urban areas, rural areas, and remote areas.
- Accurate: UAVs can provide high-resolution images and videos of PV systems, which can be used for monitoring and inspection.

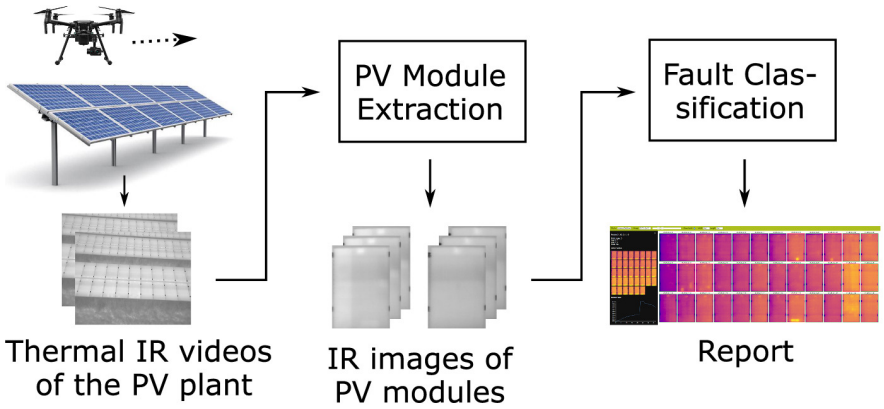



Figure 2.5: UAV-based thermal inspection of (PV) systems, illustrating the process from aerial thermal imaging to anomaly detection and reporting. (Source: [3])

Figure 2.5 illustrates the end-to-end process of automated fault detection in photovoltaic (PV) systems using (IR) thermography and deep learning-based classification. Initially, thermal IR videos of the PV plant are captured using drones equipped with IR cameras. These thermal videos are then processed to extract individual IR images corresponding to each PV module. The extracted module images serve as input to the fault classification pipeline, where a trained deep learning model analyzes each image to detect and classify various types of anomalies. The final output is a detailed report that visualizes the fault types and locations within the PV installation, enabling efficient inspection and maintenance. [3], [26]

### 2.3.5 Measurement Principle

Thermal infrared (IR) imaging relies on physical laws to measure the surface temperature of objects. According to Planck's law [27], any object in thermal equilibrium with a temperature above absolute zero emits electromagnetic radiation, with the peak wavelength depending on its temperature. Under typical environmental conditions, this radiation lies within the thermal IR range and can be detected by infrared cameras.

In the context of PV module inspection, the radiant energy emitted per unit surface area, referred to as radiant exitance, is described by the Stefan–Boltzmann law [28].

$$M_{\mathrm{PV}}(T_{\mathrm{PV}}) = \varepsilon \sigma T_{\mathrm{PV}}^4 \tag{2.1}$$

Where $M_{\mathrm{PV}}(T_{\mathrm{PV}})$ is the radiant energy emitted per unit surface area, $\varepsilon$ is the emissivity of the surface material (typically close to 1 for glass), $\sigma$ is the Stefan–Boltzmann constant, and $T_{\mathrm{PV}}$ is the module surface temperature.

This allows for accurate estimation of the PV module's surface temperature, which is crucial for detecting thermal anomalies [29].

The infrared camera detects both the emitted radiation from the module and a portion of the surrounding environmental radiation reflected by the module surface. This total measured radiation $M_{\mathrm{IR}}$ is given by:

$$M_{\mathrm{IR}} = \varepsilon M_{\mathrm{PV}}(T_{\mathrm{PV}}) + (1 - \varepsilon) M_E(T_E) \tag{2.2}$$

where $M_E(T_E)$ represents the radiant exitance from the environment at temperature $T_E$. Since glass has high emissivity ($\varepsilon \approx 1$) and the environment is typically cooler than the module, the contribution from the reflected component is minimal. This allows for accurate estimation of the PV module's surface temperature, which is crucial for detecting thermal anomalies.

## 2.4 Power Losses Estimation

Power loss in PV systems represents the reduction in electrical output relative to the theoretical maximum power generation capacity. Understanding these loss mechanisms is essential for optimizing system design, operation, and maintenance strategies.

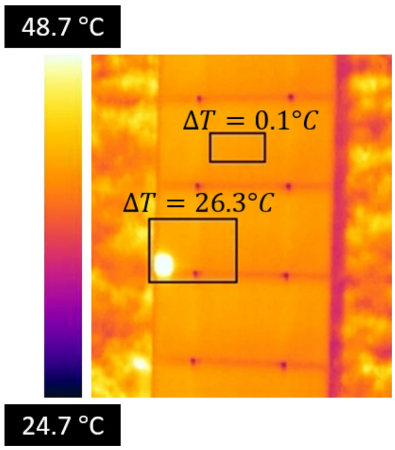



Figure 2.6: Hotspot ($\Delta T$), (Source:[19])

$\Delta T$: It is defined as the temperature difference between the hottest pixel within an anomaly and the average temperature of nearby nominal (healthy) cells or modules. Anomalies with $\Delta$T $\geq$ 20 °C are flagged as high-severity, while those with $\Delta$T $<$ 20 °C are considered low or medium severity depending on thresholds at 10 °C and 20 °C.

$$\Delta T = T_{\text{anomaly,max}} - T_{\text{healthy,avg}} \tag{2.3}$$

where $T_{\text{anomaly, max}}$ represents the maximum temperature of the anomalous region and $T_{\text{healthy,avg}}$ represents the average temperature of surrounding healthy modules or cells.

Figure 2.6 shows an IR image of a PV module without a hotspot having $\Delta T$ = 0.1°C, while it is equal to $\Delta T$ = 26.3°C when a module is affected by a single hotspot.

**Estimated Power Loss**: Power loss ($P_{\text{Loss}}$) is estimated using the following equation:

$$P_{\text{Loss}} = N_{\text{modules}} \times P_{\text{STC}} \times \text{Power Factor}$$

where $N_{\text{modules}}$ is the number of affected modules, $P_{\text{STC}}$ is the module's rated peak power under standard test conditions (STC), and the power factor is an empirical coefficient reflecting anomaly severity and type. [30]

Following Table 2.1 shows the $\Delta$T classification and power factor estimates for PV anomalies. The power factor is a value between 0 and 1, and it is used to estimate the power loss of the PV system. The power factor is calculated based on the $\Delta$T of the PV system. [30].

| Anomaly | ΔT Range | Severity Class | Estimated Power Factor |
|---|---|---|---|
| **Cell** | $< 10$ °C → Low<br>10–20 °C → Medium<br>$\geq 20$ °C → High | Low<br>Medium<br>High | ∼0.001 – 0.1 |
| **Cell Multi** | Same as above, applied to multiple cells | Low<br>Medium<br>High | ∼0.005 – 0.2 |
| **Hotspot** | Typically $\geq 20$ °C | High | ∼0.01 – 0.2 |
| **Hotspot Multi** | Same ΔT range as one-hotspot but multiple areas | High | ∼0.02 – 0.3 |
| **Diode** | Examples around 3–10 °C (e.g., 3.8 °C) | Low / Medium (<20 °C) | ∼0.3 – 0.7 |
| **Diode Multi** | Multiple bypass zones, ΔT up to 10–20 °C | Medium | ∼0.5 |
| **Offline Module** | Negative ΔT (cooler than neighbors) | High | ∼1.0 |
| **Shadowing / Vegetation / Soiling** | Usually $< 10$ °C, may exceed 20°C chronically | Low / Medium | ∼0.1 – 0.4 |

Table 2.1: ΔT Ranges and Power Factor Estimates for PV Anomalies, Source:[30]

*Note:* ΔT classes are defined as low (<10 °C), medium (10–20 °C), and high (≥20 °C). Power factor ranges are approximate empirical values from Raptor Maps defaults used for scaling DC power loss. [30].

## 2.5 Machine Learning and Deep Learning

Machine Learning (ML), a subfield of Artificial Intelligence (AI), focuses on the development of algorithms that can learn from data to make predictions or decisions without being explicitly programmed. By iteratively processing data, ML models aim to improve their performance over time, emulating aspects of human learning [31]. A typical supervised ML algorithm consists of three fundamental components [32]:

- **Decision process:** The algorithm processes input data and applies a model, often statistical or mathematical, to recognize patterns or make predictions.
- **Error function:** This component evaluates the model's predictions by comparing them against known outputs, quantifying the discrepancy through a loss or cost function.

- **Optimization process:** Based on the error feedback, the model updates its internal parameters to improve future predictions, typically using optimization techniques like gradient descent.

### 2.5.1 Machine Learning Types

The presence or lack of human intervention on raw data whether in the form of rewards, targeted feedback, or labels defines a variety of machine learning model types. According to the blog post of [33], there are different machine learning types categorized as:

- **Supervised learning:** In this approach, the model is trained on labeled data, enabling it to learn mappings between inputs and known outputs for accurate prediction.
- **Unsupervised learning:** This method deals with unlabeled data, where the model autonomously identifies hidden patterns, clusters, or structures without explicit guidance.
- **Semi-supervised learning:** Combines a small portion of labeled data with a large amount of unlabeled data to improve learning accuracy with minimal supervision.
- **Reinforcement learning:** The model learns optimal actions through interactions with an environment, guided by reward signals based on the outcomes of its decisions.
- **Deep learning:** A specialized branch of machine learning that employs deep neural networks to extract complex features and hierarchical representations from large datasets.

**Deep Learning (DL)** is a specialized subfield of Machine Learning (ML) that employs deep neural networks composed of multiple hidden layers to learn complex representations from data. While ML focuses on algorithms capable of learning from data with limited manual intervention, DL extends this capability by enabling hierarchical feature extraction directly from raw inputs. This hierarchical relationship among Artificial Intelligence (AI), Machine Learning (ML), and Deep Learning (DL) is illustrated in Figure 2.7.

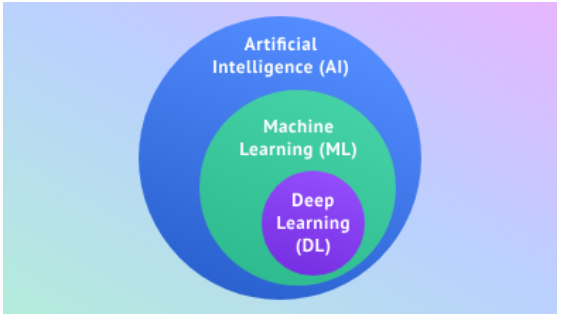



Figure 2.7: Relationship between AI, ML, and DL

### 2.5.2 Computer Vision

This section outlines key computer vision techniques commonly employed in deep learning-based approaches for automated PV plant inspection. These techniques are essential for analyzing visual data, detecting anomalies, and classifying PV module conditions. Follwing figure 2.8 illustrates the common computer vision tasks that are used in PV plant inspection.

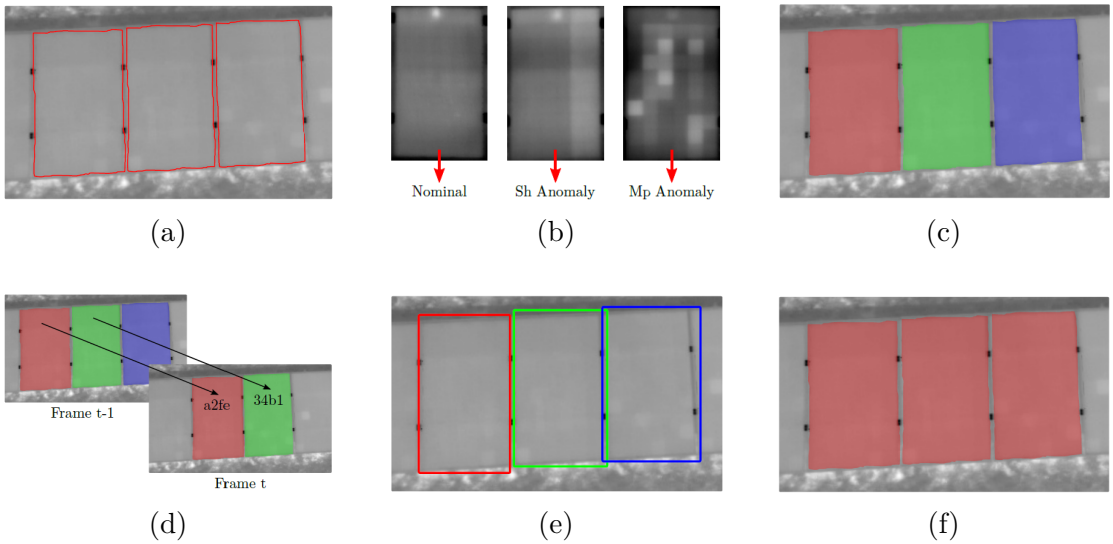



Figure 2.8: Common computer vision tasks in PV plant inspection (a) Edge Detection, (b) Image Classification, (c) Instance Segmentation, (d) Multi-object Tracking, (e) Object Detection, (f) Semantic Segmentation. Source: [34]

- **Edge Detection:** Identifies discontinuities in pixel intensity to detect structural boundaries, useful for locating module edges, cracks, or cell interconnects.
- **Image Classification:** Assigns a class label to an entire image (e.g., Nominal, Shading Anomaly, Module Power Anomaly), enabling global diagnosis of PV module health.
- **Instance Segmentation:** Differentiates between multiple objects of the same class (e.g., PV modules), assigning distinct masks to each instance within the same image.
- **Multi-object Tracking:** Tracks the identity of multiple objects (e.g., PV modules) across sequential frames, helping to monitor modules.
- **Object Detection:** Detects and localizes individual objects using bounding boxes, such as identifying and isolating faulty modules within a PV array.
- **Semantic Segmentation:** Performs pixel-level classification to label regions of the image by class, such as labeling all PV modules or faulty areas (e.g., hotspots, shading) in red.

## 2.6 Vision Transformer

The Vision Transformer (ViT) [35] is a neural network architecture that extends the Transformer model, initially developed for natural language processing (NLP) tasks [36], to image classification. The core concept is to represent an image as a sequence of fixed-size patches, analogous to token sequences in textual data. This formulation enables ViT to leverage self-attention mechanisms to model long-range dependencies and global contextual relationships within the image, without relying on convolutional operations [35]. The architectural overview of the Vision Transformer is illustrated in Figure 2.9. And detailed mathematical formulation of the Vision Transformer is discussed below.

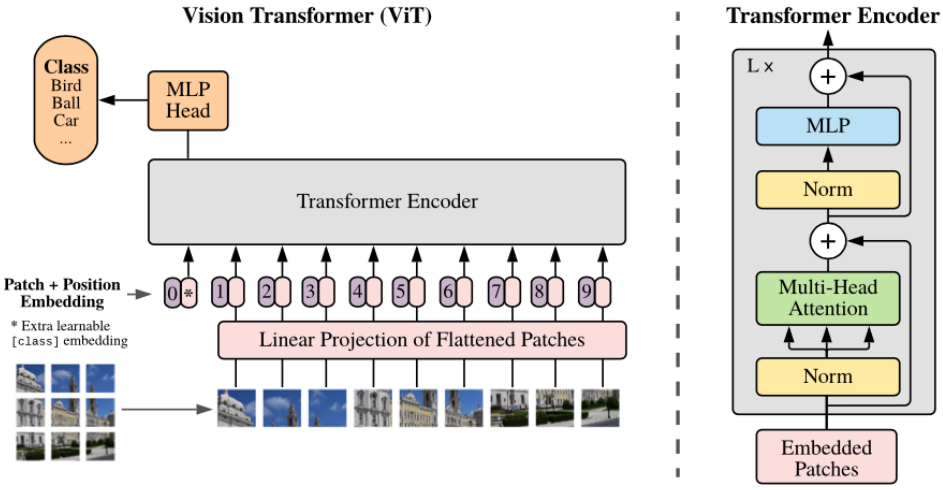



Figure 2.9: Vision Transformer Architecture, Source:[35]

**1. Patch Division:** The input image is divided into non-overlapping patches of a fixed size, depending on the variant of the Vision Transformer (e.g., 32×32 pixels for ViT-B32. Here, B is the base model, and 32 is the patch size [35]; see Figure 2.9). Each patch is treated as a token, similar to a word in a sentence. Given an input image $\mathbf{x} \in \mathbb{R}^{H \times W \times C}$ of size $H \times W \times C$ (height, width, and number of channels), it is divided into $N$ non-overlapping patches of size $P \times P$:

$$N = \frac{H \times W}{P^2} \tag{2.4}$$

Each patch is flattened into a vector of dimension $P^2 \cdot C$, where $C$ is the number of channels (e.g., $C = 3$ for RGB images).

**2. Patch Embedding:** The input image is divided into fixed-size patches, which are then flattened and linearly projected into a higher-dimensional space. This creates a sequence of patch embeddings that serve as the input to the transformer. Each flattened patch is linearly mapped to a $D$-dimensional embedding:

$$\mathbf{z}_i = \mathbf{E} \cdot \text{Flatten}(\text{Patch}_i) \tag{2.5}$$

where $\mathbf{E} \in \mathbb{R}^{(P^2 \cdot C) \times D}$ is a learnable weight matrix, and $i = 1, \ldots, N$ refereing to the index of the image patch (see Figure 2.9).

**3. [CLS] Token:** A special classification token (CLS token) is prepended to the sequence of patch embeddings. This token is used to aggregate information from all patches and is crucial for tasks like image classification. A learnable classification token $\mathbf{z}_{\text{cls}} \in \mathbb{R}^D$ is prepended to the sequence of patch embeddings:

$$\mathbf{Z}_0 = [\mathbf{z}_{\text{cls}}, \mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_N] \tag{2.6}$$

Here $\mathbf{z}_{\text{cls}}$ is the classification token, $\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_N$ are the patch embeddings, and $\mathbf{Z}_0$ is the sequence of patch embeddings with the CLS token.

**4. Positional Encoding:** Since transformers do not inherently understand the spatial relationships between patches, positional encodings are added to the patch embeddings to provide information about their relative positions in the image. Positional encodings $\mathbf{p}_i \in \mathbb{R}^D$ are added to each token to retain positional information:

$$\mathbf{Z}_0^{\text{pos}} = \mathbf{Z}_0 + \mathbf{P} \tag{2.7}$$

where $\mathbf{P}$ is the matrix of positional encodings and $\mathbf{Z}_0^{\text{pos}}$ is the sequence of patch embeddings with the CLS token and positional encodings.

The original Transformer uses sinusoidal encodings [36]:

$$\text{PE}_{(pos,2i)} = \sin\left(\frac{pos}{10000^{2i/d_{\text{model}}}}\right), \quad \text{PE}_{(pos,2i+1)} = \cos\left(\frac{pos}{10000^{2i/d_{\text{model}}}}\right) \tag{2.8}$$

**5. Transformer Encoder:** Each transformer encoder layer consists of a multi-head self-attention (MHSA) mechanism and a feed-forward neural network (MLP) (see Figure 2.10), both followed by residual connections and layer normalization. The self-attention mechanism enables the model to weigh the importance of different image patches relative to each other, thereby capturing global contextual information (see Figure 2.10 for more details). The sequence is passed through $L$ layers of the Transformer encoder, each consisting of multi-head self-attention (MHSA) and MLP blocks:

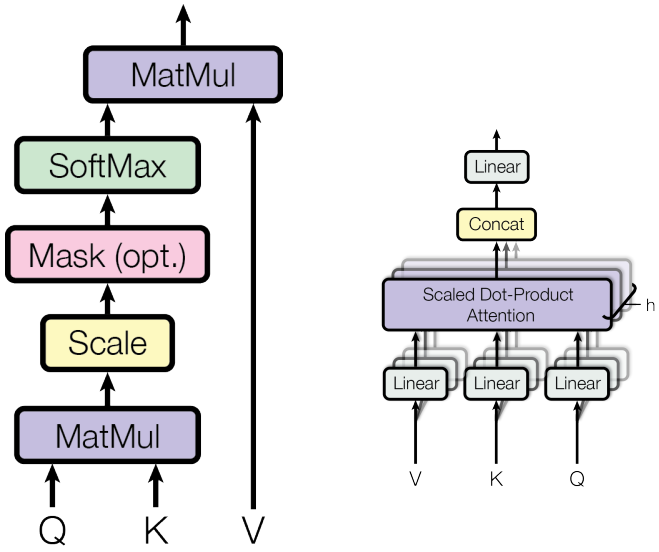
$$\mathbf{Z}'_l = \text{MHSA}(\text{LayerNorm}(\mathbf{Z}_{l-1})) + \mathbf{Z}_{l-1} \tag{2.9}$$

$$\mathbf{Z}_l = \text{MLP}(\text{LayerNorm}(\mathbf{Z}'_l)) + \mathbf{Z}'_l \tag{2.10}$$

where $l = 1, \ldots, L$.

**6. Classification Head:** After processing through the transformer layers, a classification head (often a simple fully connected layer) is used to produce the final output, such as class probabilities for the image classification of 2 classes (anomalies and normal), 11 classes (11 different Anomalies) and 12 classes (No anomaly and 11 different Anomalies). The final output corresponding to the [CLS] token is used for classification:

$$\hat{y} = \text{MLP}_{\text{head}}(\mathbf{z}_{\text{cls}}^{(L)}) \tag{2.11}$$




(a) Scaled Dot-Product Attention

(b) Multi-Head Self Attention

Figure 2.10: Self-Attention (Source, [36])

The attention mechanism in ViT is computed using the scaled dot-product attention formula [36]:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{2.12}$$

Here $d_{\text{k}}$ is the dimension of the key vectors. The positional encoding is added to the patch embeddings to provide information about the relative positions of the patches in the image. This is crucial for the transformer to understand the spatial relationships between the patches, as transformers do not inherently capture positional information.

**Multi-Head Self-Attention (MHSA)** mechanism is a key component of the Vision Transformer (ViT) architecture. It allows the model to focus on different parts of the input sequence (in this case, the image patches) simultaneously, enabling it to capture complex relationships and dependencies between the patches. The MHSA mechanism works by computing attention scores for each patch in the sequence. This is done by projecting the input embeddings into three different spaces: Query (Q), Key (K), and Value (V). The attention scores are then calculated using the dot product of the Query and Key vectors, scaled by the square root of the dimension of the Key vectors. The resulting attention scores are passed through a softmax function to obtain probabilities, which are then used to weight the Value vectors. This process is repeated for multiple heads, allowing the model to learn different attention patterns and capture diverse features from the input data. The outputs from all heads are concatenated and projected back to the original dimension, producing a rich representation of the input sequence that incorporates information from all patches. The MHSA mechanism can be represented as follows:

$$\text{MHSA}(X) = \text{Concat}(\text{head}_1, \text{head}_2, \ldots, \text{head}_h)W^O \tag{2.13}$$

Where each head is computed as:

$$\text{head}_i = \text{Attention}(Q_i, K_i, V_i) = \text{softmax}\left(\frac{Q_i K_i^T}{\sqrt{d_k}}\right) V_i \tag{2.14}$$

where $W_i^Q$, $W_i^K$, and $W_i^V$ are the weight matrices for the $i$-th head.
Note: Sources for the mathematical formulations are from Dosovitskiy et al. [35] and Vaswani et al. [36].

## 2.7 DeiT - Data-Efficient Image Transformer

DeiT (Data-efficient Image Transformer) is a variant of the Vision Transformer (ViT) architecture that focuses on improving the data efficiency of training transformers for image classification tasks [37]. The most notable contribution is the use of a distillation token, enabling DeiT to be trained with knowledge distillation from a powerful CNN-based teacher (e.g., RegNetY-16GF), without requiring large-scale datasets or external supervision. This approach allows DeiT to match or even surpass CNN performance while retaining the advantages of transformer-based architectures. [37]. A typical DeiT architecture consists of the following components as shown in Figure 2.11:

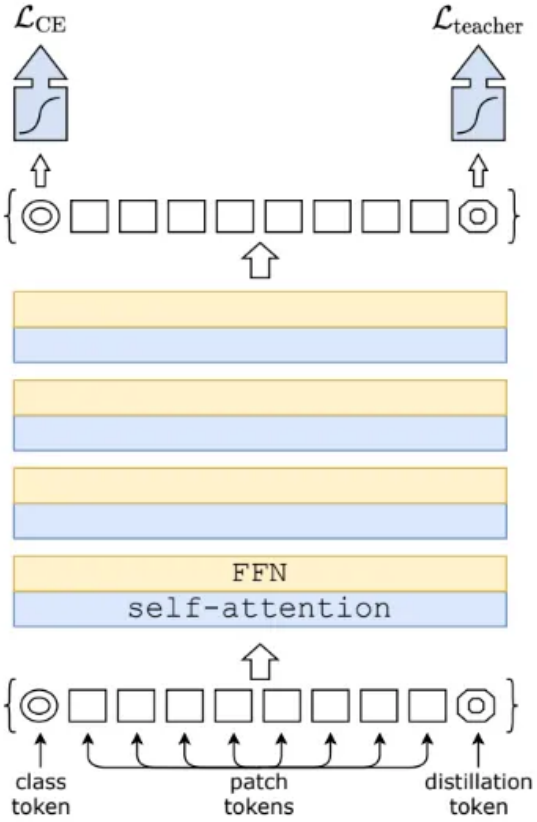



Figure 2.11: Architecture of the DeiT (Data-efficient Image Transformer) model with knowledge distillation. Source: [37]

**Mathematical Formulation:** The following mathematical formulation outlines the key components of the DeiT architecture and its training methodology:

**Image Patching and Embedding:** An input image $x \in \mathbb{R}^{H \times W \times C}$ is divided into $N = \frac{HW}{P^2}$ non-overlapping patches:

$$x_p = \{x_p^1, x_p^2, ..., x_p^N\}, \quad x_p^i \in \mathbb{R}^{P^2 \cdot C} \tag{2.15}$$

Each patch $x_p^i$ is flattened into a vector, where $P$ is the patch size. Here, $H$, $W$, and $C$ represent the height, width, and number of channels of the image, respectively.

**Patch and Positional Embeddings:** Each flattened patch $x_p^i \in \mathbb{R}^{P^2 \cdot C}$ is projected into a $D$-dimensional embedding space using a learnable linear projection matrix $\mathbf{E} \in \mathbb{R}^{(P^2 \cdot C) \times D}$. The resulting sequence of patch embeddings is prepended with a classification token $x_{\text{class}}$ and a distillation token $x_{\text{distill}}$, forming the input sequence:

$$\mathbf{z}_0 = [x_{\text{class}}; x_p^1\mathbf{E}; x_p^2\mathbf{E}; \ldots; x_p^N\mathbf{E}; x_{\text{distill}}] + \mathbf{E}_{\text{pos}} \tag{2.16}$$

where $\mathbf{E}_{\text{pos}} \in \mathbb{R}^{(N+2) \times D}$ represents the positional embedding matrix added element-wise to retain spatial information in the sequence.

**Transformer Encoder:** The resulting sequence is passed through $L$ Transformer encoder layers. Each layer consists of two sublayers: a multi-head self-attention (MSA) mechanism and a feed-forward network (MLP), each followed by layer normalization (LN) and residual connections. The operations in the $\ell$-th layer are defined as:

$$\mathbf{z}'_\ell = \text{MSA}(\text{LN}(\mathbf{z}_{\ell-1})) + \mathbf{z}_{\ell-1} \tag{2.17}$$

$$\mathbf{z}_\ell = \text{MLP}(\text{LN}(\mathbf{z}'_\ell)) + \mathbf{z}'_\ell \tag{2.18}$$

Here, $\ell \in \{1, 2, \ldots, L\}$ denotes the layer index, $\mathbf{z}_{\ell-1}$ is the input to the $\ell$-th layer, $\mathbf{z}'_\ell$ is the intermediate representation after the attention block, and $\mathbf{z}_\ell$ is the final output after the MLP block.

**Distillation Token Interaction:** The distillation token $x_{\text{distill}}$ acts as a learnable query vector that aggregates information from patch tokens. Its attention mechanism can be expressed as:

$$a_{\text{distill}} = \sum_{i=1}^{N} \text{softmax}\left(\frac{q_{\text{distill}} k_i^T}{\sqrt{d_k}}\right) v_i \tag{2.19}$$

where $q_{\text{distill}}$, $k_i$, and $v_i$ are the query, key, and value vectors for the distillation token, and $d_k$ is the dimension of the key vectors.

**Hard Distillation Loss:** DeiT supports hard distillation using the cross-entropy loss between the student's prediction $y_s$ and the hard label provided by the teacher $y_t$:

$$\mathcal{L}_{\text{hard}} = \text{CE}(y_s, y_t) \tag{2.20}$$

**Soft Distillation Loss:** Soft distillation minimizes the Kullback-Leibler divergence between the softened output distributions of the teacher and the student:

$$\mathcal{L}_{\text{soft}} = \tau^2 \cdot \text{KL}\left(\text{softmax}\left(\frac{z_t}{\tau}\right) \middle\| \text{softmax}\left(\frac{z_s}{\tau}\right)\right) \tag{2.21}$$

where $z_s$ and $z_t$ are the logits from the student and teacher, and $\tau$ is the temperature parameter.

**Combined Training Objective:** The total loss $\mathcal{L}_{\text{total}}$ is a combination of the ground truth classification loss $\mathcal{L}_{\text{CE}}$ with both hard and soft distillation losses ($\mathcal{L}_{\text{hard}}$ and $\mathcal{L}_{\text{soft}}$), weighted by balancing hyperparameters $\alpha$ and $\beta$:

$$\mathcal{L}_{\text{total}} = \alpha\mathcal{L}_{\text{CE}} + (1-\alpha)\left(\beta\mathcal{L}_{\text{hard}} + (1-\beta)\mathcal{L}_{\text{soft}}\right) \tag{2.22}$$

Note: Source for all the above information and mathematical formulations is from Touvron et al. [37] original paper.

DeiT is particularly effective for image classification tasks, especially when training data is limited. It achieves competitive performance compared to CNNs while maintaining the advantages of transformers, such as scalability and flexibility in handling various input sizes and modalities.

While Vision Transformers (ViTs) demonstrated strong performance on large-scale datasets, they lacked the inductive biases inherent in convolutional neural networks (CNNs), such as locality, translation equivariance, and spatial hierarchy [35]. These biases enable CNNs to generalize effectively from relatively small datasets. In contrast, ViTs treat images as flat sequences of patches without encoding any spatial structure, making them highly dependent on large amounts of training data for effective learning. To address this limitation, DeiT (Data-Efficient Image Transformer) was introduced, incorporating a distillation mechanism and optimized training strategies to improve data efficiency and stability when training on smaller datasets, such as ImageNet-1K [37].

## 2.8 ConvNeXt: CNNs Re-Imagined

The ConvNeXt model was introduced by Zhuang Liu et al. [38] in A ConvNet for the 2020s. It is a pure convolutional neural network (ConvNet) architecture inspired by the training strategies and design principles of Vision Transformers (ViTs). While it does not introduce fundamentally new components, ConvNeXt re-engineers standard ConvNet elements using modern design choices that significantly enhance accuracy, scalability, and efficiency, surpassing many ViT variants in performance [38].

What distinguishes ConvNeXt from traditional ConvNet architectures (e.g., LeNet, AlexNet, VGG, ResNet) is its deliberate architectural refinement. It retains the core convolutional backbone but integrates Transformer-inspired practices such as large kernel depthwise convolutions, Layer Normalization instead of BatchNorm, GELU activations, and inverted bottlenecks. Moreover, it employs AdamW optimization, stochastic depth, label smoothing, and strong data augmentations (Mixup, CutMix), resulting in improved generalization. [39]

Key architectural features of ConvNeXt-Tiny include:

**Large Kernel Sizes:** According to Liu et al. [40], unlike traditional CNNs that typically use 3×3 kernels, ConvNeXt employs larger kernel sizes (7×7) in early layers. This larger receptive field allows the model to capture broader spatial

relationships, similar to the global attention mechanisms in Transformers, but through purely convolutional operations.

**Inverted Bottleneck Design:** ConvNeXt adopts an inverted bottleneck structure where the channel dimension is expanded in the middle of each block, similar to MobileNet [41] architectures. This design choice improves the model's ability to learn more complex representations while maintaining computational efficiency [42].

**Depth-wise Convolution:** According to Liu et al. [40], depth-wise convolution is a computationally efficient alternative to standard convolution that processes each input channel independently. Unlike standard convolution, which applies filters across all input channels simultaneously, depth-wise convolution applies a separate filter to each input channel, significantly reducing computational complexity.

Besides the computational efficiency, depth-wise convolution also has some other advantages.

- **Reduced Parameters:** Depth-wise convolution reduces the number of parameters compared to standard convolution, making it more efficient in terms of memory usage and computational cost.
- **Improved Representational Power:** Depth-wise convolution allows the model to learn more complex features by capturing local patterns in each channel independently.
- **Improved Regularization:** Depth-wise convolution can help prevent overfitting by encouraging the model to learn more diverse features, including LayerNorm, GELU activation functions [43], and improved regularization strategies. These modifications help stabilize training and improve convergence, particularly when training deeper networks.

A typical ConvNeXt-Tiny architecture looks like this, as shown in Figure 2.12.

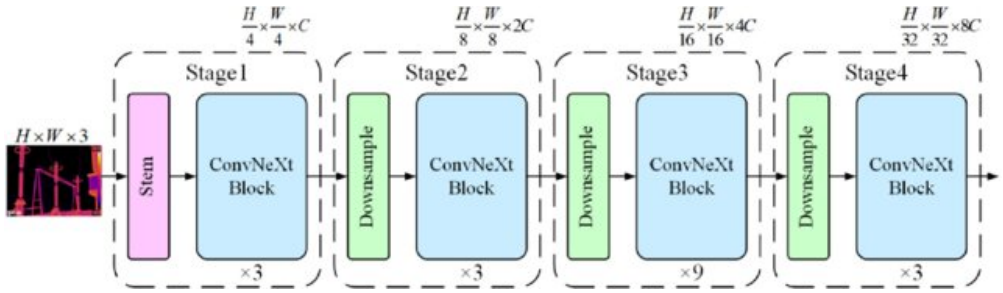



Figure 2.12: ConvNeXt-Tiny Architecture Overview. Source: [44]

## 2.9 Related Work

Several studies in the literature have proposed the application of deep learning models for anomaly detection in photovoltaic (PV) systems using infrared (IR) imagery.

Sinap et al. [45] proposed a CNN-based model for detecting and classifying faults in PV modules using a dataset of 20,000 IR images covering 12 anomaly types caused by conditions such as partial shading, short circuits, and dust accumulation. Their method addressed challenges like low image resolution, class imbalance, and hyperparameter tuning through techniques such as histogram equalization, data augmentation, and optimization strategies, including Optuna and Hyperband. The model achieved 92% accuracy in anomaly detection and 82% in multi-class classification, demonstrating strong potential for practical PV monitoring applications.

In 2023, Pamungkas et al. [46] introduced a lightweight model, Coupled UDense-Net, for efficient solar panel fault classification. Designed for real-time scenarios, the model utilized geometric transformations and GAN-based augmentation to improve performance on imbalanced datasets. It achieved classification accuracies of 99.39% for 2-class, 96.65% for 11-class, and 95.72% for 12-class tasks, making it highly suitable for large-scale solar PV inspection.

Fonseca Alves et al. [47] developed a CNN framework leveraging IR thermography for automatic fault classification in PV modules, investigating up to eleven defect classes. They explored data augmentation strategies and confusion matrix analysis to address inter-class variability and dataset imbalance. Their model achieved 92.5% accuracy in detecting anomalies and 78.85% accuracy in classifying eight selected defect types, revealing both the potential and limitations of CNNs in this domain.

In contrast to these CNN-based approaches, this study investigates the effectiveness of transformer-based models, including Vision Transformer (ViT), Data-efficient Image Transformer (DeiT), and ConvNeXt, a transformer-inspired pure CNN model for anomaly detection and classification in PV modules. These models are evaluated on the same IR dataset to provide a comparative performance analysis.

CHAPTER

# THREE

# METHODOLOGY

This chapter outlines the methodology employed in this study. It begins with a detailed description of the dataset, followed by the challenges with it. Image pre-processing and augmentation strategies are presented as solutions to address these challenges, ensuring the dataset is suitably prepared for models training and evaluation. The chapter also describes the selected deep learning architectures, along with the implementation details, software frameworks, and hardware configurations used to develop and train the models. Ultimately, it also presents the performance metrics used to evaluate them.

## 3.1 Dataset

The InfraredSolarModules dataset [48] is a publicly available collection of 20,000 thermographic images depicting anomalies in solar photovoltaic (PV) systems. The dataset contains 10,000 images distributed across eleven distinct anomaly classes and 10,000 no-anomaly images, enabling both binary and multi-class classification tasks. Images were acquired using infrared cameras (3–13.5 µm spectral range) mounted on aircraft and UAVs, then cropped to module level and systematically categorized [48].

The dataset exhibits inherent class imbalance reflecting real-world anomaly frequencies, as shown in Figure 3.1 and detailed in Table 3.1. Common PV faults include thermal hot-spots, shading, cell defects, diode failures, and surface soiling. Figure 3.2 presents representative samples from each class, revealing that some anomaly types exhibit similar thermal patterns while others show high intra-class variability, presenting classification challenges for machine learning models.

Compiled by RaptorMaps with manual annotations, the dataset is publicly accessible online[1] [49] and addresses the scarcity of publicly available thermographic data for PV fault detection research.

[1] `https://github.com/RaptorMaps/InfraredSolarModules`

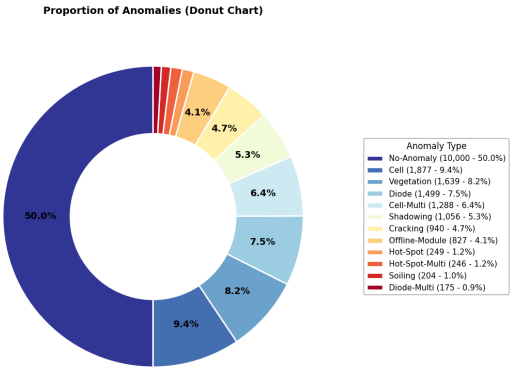



Figure 3.1: Class distribution in the IR dataset.

| **Class** | **No. of Images** | **% of total** | **Description** | **Pattern** |
|---|---|---|---|---|
| Cell | 1,877 | 9.4% | Hot spot occurring with square geometry in a single cell. |  |
| Cell-Multi | 1,288 | 6.4% | Hot spots occurring with square geometry in multiple cells. |  |
| Cracking | 940 | 4.7% | Module anomaly caused by cracking on the module surface. |  |
| Diode | 1,499 | 7.5% | Activated bypass diode, typically 1/3 of the module. |  |
| Diode-Multi | 175 | 0.9% | Multiple activated bypass diodes, typically affecting 2/3 of the module. |  |
| Hot-Spot | 249 | 1.2% | Hot spot on a thin film module. |  |
| Hot-Spot-Multi | 246 | 1.2% | Multiple hot spots on a thin film module. |  |
| No-Anomaly | 10,000 | 50.0% | Nominal solar module. |  |
| Offline-Module | 827 | 4.1% | The entire module is heated. |  |
| Shadowing | 1,056 | 5.3% | Sunlight obstructed by vegetation, man-made structures, or adjacent rows. |  |
| Soiling | 204 | 1.0% | Dirt, dust, or other debris on the surface of the module. |  |
| Vegetation | 1,639 | 8.2% | Panels blocked by vegetation. |  |

Table 3.1: IR Dataset Description with Anomalies Patterns

Figure 3.1 shows the class distribution, highlighting perfect balance for binary classification while revealing imbalance in multi-class scenarios. Figure 3.2 presents representative samples from each class, demonstrating the distinct thermal signatures and classification challenges.

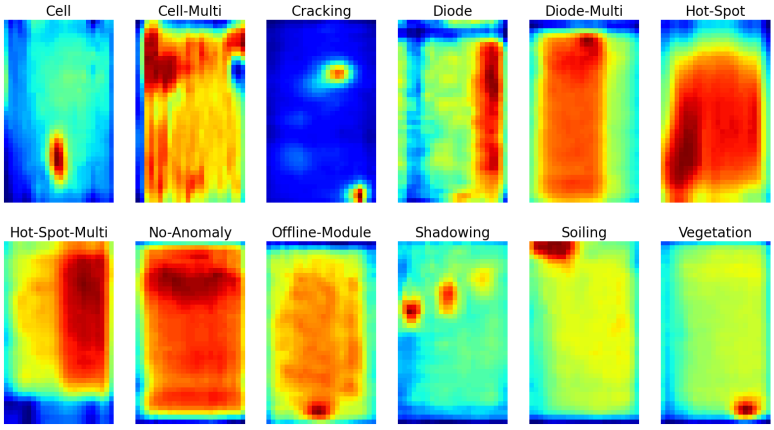



Figure 3.2: Representative samples from each anomaly class in the IR dataset (Grad-CAM Visualization)

### 3.1.1 Classes Descriptions

The dataset comprises 12 distinct classes representing different PV system conditions:

- **Cell** - Hot spot in a single cell with square geometry.
- **Cell-Multi** - Hot spots in multiple cells with square geometry.
- **Cracking** - Module surface cracking anomaly.
- **Hot-Spot** - Thin film module hot spot causing localized temperature elevation due to manufacturing defects, shading, or electrical mismatches.
- **Hot-Spot-Multi** - Multiple hot spots on thin film modules. Can cause 2-33% power loss depending on affected cells [50].
- **Diode** - Activated bypass diode affecting 1/3 of module, causing 33% power reduction [50].
- **Diode-Multi** - Multiple activated bypass diodes affecting 2/3 of module.
- **Offline-Module** - Electrically inactive module appearing uniformly cooler in thermography.
- **Shadowing** - Sunlight obstruction by vegetation or structures, causing 20-40% power reduction [51], [52].
- **Soiling** - Surface contamination causing 1.5-6.2% power loss [53].
- **Vegetation** - Panels blocked by vegetation.
- **No-Anomaly** - Normal operating module.

### 3.1.2 Dataset Sample Images

The sample images in table 3.2 illustrate the diverse thermal characteristics within each class. Each row displays five randomly selected samples from the respective anomaly class, showcasing the intra-class variability that the deep learning models encounter during training. This visual representation helps understand the complexity of the classification task and the distinct thermal signatures that differentiate each anomaly type from normal operating conditions.

| Class | Sample 1 | Sample 2 | Sample 3 | Sample 4 | Sample 5 |
|---|---|---|---|---|---|
| Cell |  |  |  |  |  |
| Cell-Multi |  |  |  |  |  |
| Cracking |  |  |  |  |  |
| Diode |  |  |  |  |  |
| Diode-Multi |  |  |  |  |  |
| Hot-Spot |  |  |  |  |  |
| Hot-Spot-Multi |  |  |  |  |  |
| No-Anomaly |  |  |  |  |  |
| Offline-Module |  |  |  |  |  |
| Shadowing |  |  |  |  |  |
| Soiling |  |  |  |  |  |
| Vegetation |  |  |  |  |  |

Table 3.2: Random sample images from each class in the processed dataset (5 samples per class), illustrating the diverse thermal characteristics and intra-class variability

### 3.1.3 Dataset Challenges

The InfraredSolarModules dataset presents two primary challenges that directly impact model development and training strategies:

**Class Imbalance:** While the dataset maintains perfect balance for binary classification (Anomaly vs. No-Anomaly), significant class imbalance exists for multi-class tasks. As shown in Table 3.1, minority classes such as Diode-Multi (0.9%), Soiling (1.0%), and Hot-Spot (1.2%) are substantially underrepresented compared to the No-Anomaly class (50.0%). This disparity poses challenges for deep learning models in effectively learning minority class patterns, potentially leading to biased predictions favoring majority classes.

**Low Image Resolution:** The infrared images have a resolution of only 24×40 pixels, resulting in poor visual clarity that complicates anomaly detection. This limited resolution makes it difficult to discern subtle thermal patterns and fine-grained features that are crucial for accurate classification. The low-resolution constraint necessitates specialized image enhancement techniques to improve feature visibility and model performance.

These challenges are addressed through comprehensive data preprocessing and augmentation strategies, as detailed in Section 3.2.

## 3.2 Data Preprocessing and Augmentation

Data preprocessing is a crucial step in machine learning and deep learning pipelines. It involves transforming raw data into a format that is suitable for training models. This process may include tasks such as data cleaning, normalization, feature extraction, and data augmentation. Proper preprocessing helps improve model performance, reduces overfitting, and ensures that the data is in a consistent format for training and evaluation [54].

### 3.2.1 Image Preprocessing

Image processing refers to the set of techniques used to analyze and enhance visual information in images, facilitating the detection and interpretation of relevant patterns. It is a crucial step in the data preprocessing process. [55]

**Unsharp Mask Filter:** The Unsharp Mask Filter is an image processing method used to make images appear sharper. It works by creating a blurred version of the original image and subtracting it from the original, which emphasizes edges and fine details [56], [57], [58].

Images were enhanced using the Unsharp Mask Filter technique to improve clarity and highlight thermal anomalies in solar panel images. This preprocessing step sharpens image edges and boosts contrast, helping deep learning models better identify subtle fault patterns.

Let $I(x, y)$ represent the original image at pixel coordinates $(x, y)$, and $B(x, y)$ represent the blurred version of the image. The unsharp mask filter output $U(x, y)$ is given by:

$$U(x, y) = I(x, y) + \alpha \cdot (I(x, y) - B(x, y)) \tag{3.1}$$

where $\alpha$ is the strength parameter that controls the amount of sharpening applied.

The blurred image $B(x, y)$ is typically obtained by convolving the original image with a Gaussian kernel $G(x, y)$:

$$B(x, y) = I(x, y) * G(x, y) \tag{3.2}$$

The Gaussian kernel is defined as:

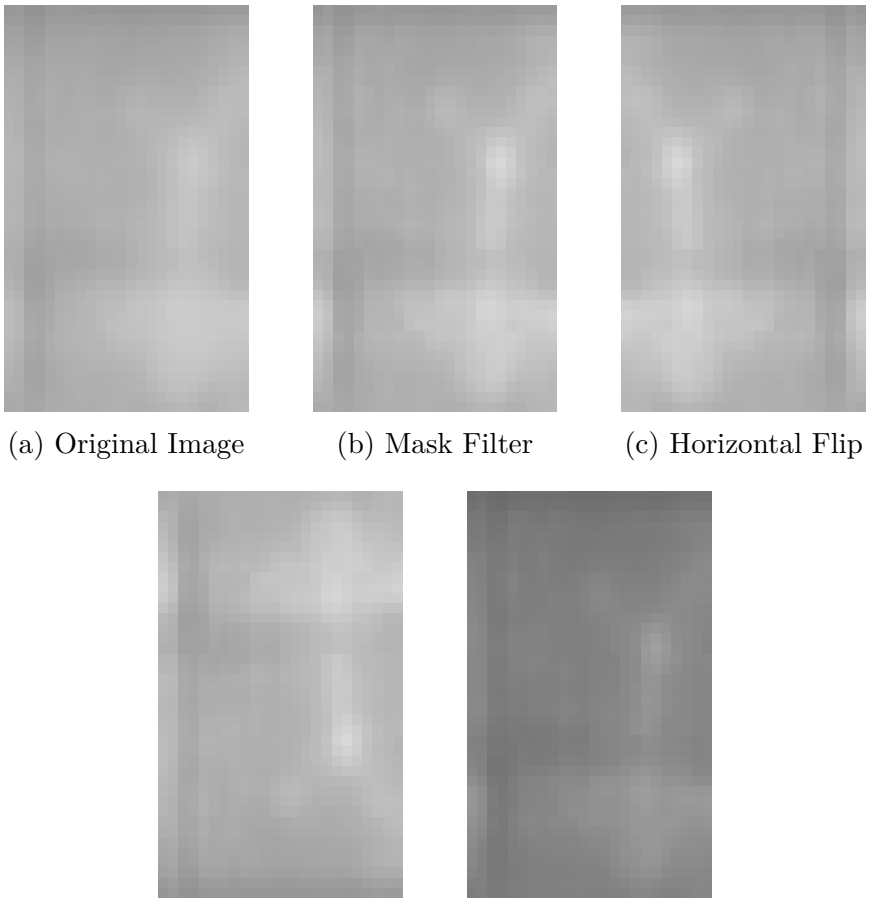
$$G(x, y) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{x^2 + y^2}{2\sigma^2}\right) \tag{3.3}$$

where $\sigma$ is the standard deviation that determines the blur radius.

The complete unsharp mask operation can be written as:

$$U(x, y) = I(x, y) + \alpha \cdot (I(x, y) - (I(x, y) * G(x, y))) \tag{3.4}$$

This formulation enhances high-frequency components in the image while preserving the overall structure, making it particularly useful for improving the visibility of fine details in thermal images of solar panels.



(a) Original Image (b) Mask Filter (c) Horizontal Flip

(d) Vertical Flip (e) Bright. Enh.

Figure 3.3: Effect of image filter and data augmentation on a Sample of Hot-Spot Multi-Anomaly image. (a) Original image, (b) after unsharp mask filter, (c) after horizontal flip, (d) after vertical flip, (e) after brightness enhancement.

### 3.2.2 Data Augmentation

Data augmentation is a technique used to artificially increase the size of a dataset by creating modified versions of existing data. This is particularly useful in scenarios where collecting new data is expensive or time-consuming. Data augmentation can involve various transformations such as rotation, scaling, flipping, and cropping for images, or adding noise and changing pitch for audio data. By introducing

variability in the training data, augmentation helps improve model generalization and robustness, reducing the risk of overfitting to the training set. It is widely used in computer vision tasks, but can also be applied to other domains like natural language processing and speech recognition. [59]

**Offline Data Augmentation** Offline data augmentation refers to the process of applying transformations to the training data before the training process begins. These augmented versions are precomputed and stored, effectively increasing the size of the training dataset. This approach is useful when computational resources during training are limited or when the augmentation transformations are computationally expensive. [60]

**Online Data Augmentation** Online data augmentation is a technique in which transformations are applied to the training data in real time during the model training process [60]. Instead of precomputing and storing augmented images, this approach dynamically generates augmented samples on-the-fly, reducing the need for additional storage space. Online augmentation is particularly beneficial when working with large datasets or limited computational resources. Common augmentation techniques include random cropping, rotation, flipping, and color jittering, applied to each mini-batch as it is fed into the model.

Table 3.3 summarizes the specific Online and Offline data augmentation techniques applied to address class imbalance in our SolarModules IR dataset. These transformations were strategically applied to minority classes to improve overall model performance and reduce bias towards majority classes.

**Data Splitting** The dataset was partitioned into three subsets: training, validation, and test. The training set was used to learn model parameters, the validation set supported hyperparameter tuning and over-fitting mitigation, and the test set was reserved for evaluating the model's final performance. For each classification task, a stratified splitting strategy was employed to ensure balanced class representation across all subsets.

For the **2-class classification** task, a total of 40,000 IR images were used, evenly divided between the Anomaly and No-Anomaly classes (20,000 each). A stratified 80:10:10 split was applied, resulting in 32,000 images for training (16,000 per class), and 4,000 images each for validation and test sets (2,000 per class). This uniform distribution was essential to enable unbiased training and evaluation. 3.4 [58]

For the **11-class classification** task, the dataset consisted of 23,100 IR images, with 2,100 images per class across 11 anomaly types: Cell, Cell-Multi, Cracking, Diode, Diode-Multi, Hot-Spot, Hot-Spot-Multi, Offline-Module, Shadowing, Soiling, and Vegetation. Using StratifiedShuffleSplit, each class was split into 1,680 training images, 210 validation images, and 210 test images. This ensured consistent class balance across all subsets. The data splitting process is illustrated in figure 3.4. [58]

For the **12-class classification** task, the dataset comprised 25,200 IR images, with 2,100 images for each of the 12 classes (including an additional No-Anomaly class). The same stratified strategy was applied, allocating 20,160 images for training, 2,520 for validation, and 2,520 for testing. Balanced class representation was maintained throughout, ensuring reliable performance evaluation. The complete

data partitioning workflow is depicted in Figure 3.4. [58]

**Image Resizing and Channel Conversion** All images in the IR dataset were processed in two steps to prepare them for the neural networks. First, the original 2-channel IR images were converted to 3-channel RGB format by copying the thermal data to create three identical channels. This conversion was necessary because the pre-trained models (ViT-B32, DeiT-B16, and ConvNeXt-Tiny) are designed to work with standard RGB images from ImageNet. Second, all images were resized to **160×160** pixels to ensure consistent input size across the training data. These preprocessing steps enable the models to process thermal images effectively while maintaining compatibility with transfer learning from natural image datasets. Also, resizing helps reduce computational overhead and memory requirements, enabling efficient model training and inference.

**Data Normalization** Data normalization scales pixel values to improve model training convergence and performance. In this study, we applied z-score normalization [61] using ImageNet statistics for all three models, which is standard practice for pre-trained DL models.

Each 3 channels was normalized independently using ImageNet mean values [0.485, 0.456, 0.406] and standard deviation values [0.229, 0.224, 0.225]. The normalization formula transforms each pixel value $x$ as:

$$x_{normalized} = \frac{x - \mu}{\sigma} \tag{3.5}$$

where $\mu$ is the mean and $\sigma$ is the standard deviation for the respective color channel. The following table 3.3 also summarizes the normalization statistics used for each channel.

| No. | Preprocessing Steps | Technique | Programming Method | Parameters and Details |
|---|---|---|---|---|
| 1 | Filtering | Unsharp filter (Offline) | UnsharpMask() (Pillow) | • Blur Radius = 2<br>• Unsharp Strength = 150% |
| 2 | Data Augmentation | Oversampling (Offline) | Image.transpose() & ImageEnhance.Brightness() (Pillow) | • Horizontal flipping<br>• Vertical flipping<br>• Brightness range = [0.2:1.2] |
| 3 | Data Augmentation | Real-time Augmentation (On-the-fly) | transforms.Compose() (PyTorch) | • RandomHorizontalFlip(p=0.5)<br>• RandomVerticalFlip(p=0.3)<br>• RandomRotation(degrees=5)<br>• ColorJitter (brightness=0.1, contrast=0.1) |
| 4 | Dataset Division | Stratified Split | StratifiedShuffleSplit() (Scikit-learn) | • Training data = 80%<br>• Validation/test = 10% each |
| 5 | Resizing & Normalization | Training start | transform.Resize() transform.Normalize() (PyTorch) | • Image size = 160×160×3<br>• Normalized range = [0:1] |

Table 3.3: Summary of Pre-processing and Data Augmentation Steps

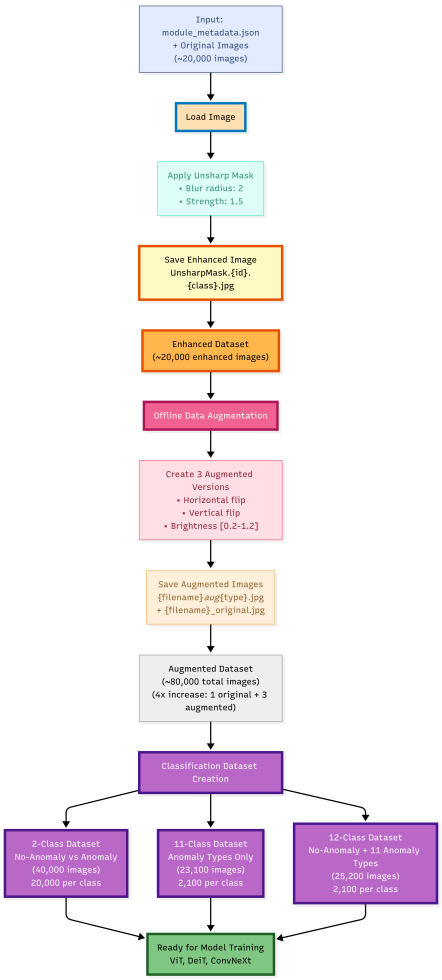



Figure 3.4: Complete Data Preprocessing and Augmentation Pipeline

## 3.3 Hardware and Software Environment

This section describes the hardware and software environment, including libraries and frameworks we utilized for training the DL models.

### 3.3.1 Hardware Setup

All model training and experiments were conducted on a system equipped with an NVIDIA GeForce RTX 2080 Ti Laptop GPU with 11 GB of memory, utilizing CUDA acceleration for efficient deep learning computations.

### 3.3.2 Libraries and Frameworks

The following is the list of libraries and frameworks used for training the DL models.

**PyTorch and TIMM**: PyTorch is an open-source deep learning framework developed by Meta AI, widely used in both research and industry for building and training neural networks [62]. It supports dynamic computation graphs,

which facilitate debugging and experimentation. For this study, we used PyTorch version 2.7.1 with CUDA acceleration. To access state-of-the-art pre-trained transformer models, we employed the PyTorch Image Models (TIMM) library, which offers efficient implementations of architectures such as DeiT-B16, ViT-B32, and ConvNeXt-Tiny, pre-trained on large-scale datasets like ImageNet-1K and ImageNet-21K. [63]

**Pillow**: A maintained fork of the Python Imaging Library (PIL) that provides robust image processing capabilities. It supports reading, writing, and manipulating images in multiple formats, such as JPEG, PNG, and BMP. Common operations include resizing, cropping, rotating, and applying filters, making it useful for preprocessing in computer vision workflows. [64]

**NumPy**: A core library for numerical computing in Python, providing support for efficient operations on large, multi-dimensional arrays and matrices. It includes a wide range of mathematical functions for linear algebra, statistics, and numerical operations, and serves as a foundational tool for many scientific and machine learning applications. [65]

**Scikit-learn**: A machine learning library that offers efficient tools for data preprocessing, model evaluation, and basic machine learning algorithms. It is primarily used for computing metrics such as accuracy, precision, recall, and F1-score. [66]

**TQDM**: A lightweight Python library that provides fast, extensible progress bars for loops and iterable objects. It is commonly used to monitor training progress during model development. [67]

**Matplotlib**: A widely used plotting library in Python for generating static, animated, and interactive visualizations. It is used for visualizing training curves, confusion matrices, and evaluation metrics. [68]

**Plotly**: A Python library for creating interactive, web-based visualizations. It is used for creating interactive plots, dashboards, and reports. [69]

## 3.4 Hyperparameters

The choice of hyperparameters is crucial for the performance of the model. The hyperparameters are the parameters that are used to control the training of the model. Following table 3.4 shows the hyperparameters used for the ViT-B32, DeiT-B16, and ConvNeXt-Tiny models.

| **Hyperparameter** | **ViT-B32** | **DeiT-B16** | **ConvNeXt-Tiny** |
|---|---|---|---|
| Learning Rate | $1 \times 10^{-4}$ | $1 \times 10^{-4}$ | $1 \times 10^{-4}$ |
| Batch Size | 64, 16 | 64, 16 | 64, 16 |
| Epochs | 60 | 60 | 60 |
| Weight Decay | 0.01 | 0.01 | 0.01 |
| Dropout Rate | 0.3 | 0.3 | 0.3 |
| Label Smoothing | 0.2 | 0.2 | 0.2 |
| LR Scheduler Patience | 5 | 5 | 5 |
| LR Decay Factor | 0.2 | 0.2 | 0.2 |
| Early Stopping Patience | 9 | 9 | 9 |

Table 3.4: Training Hyperparameters for the Models

**Learning Rate:** The learning rate is a hyperparameter that controls the step size at which the model adjusts its weights during training. A higher learning rate can lead to faster convergence, but may cause the model to overshoot the optimal solution and fail to converge. A lower learning rate, on the other hand, may result in slower convergence, but can lead to better generalization and stability. The learning rate is typically set to a value between 0.0001 and 0.01, depending on the complexity of the model and the dataset.

**Batch Size:** The batch size is the number of samples processed in each iteration of the training process. A larger batch size can lead to faster training, but may require more memory and computational resources. A batch size of 64 was used for the 2-class classification task, and 16 was used for the 11-class and 12-class classification tasks for better generalization and stability.

**Epochs:** The number of epochs is the number of times the entire training dataset is passed through the model during training. A larger number of epochs can lead to better convergence, but may result in over-fitting. A smaller number of epochs, on the other hand, may result in under-fitting. The number of epochs is typically set to a value between 10 and 100, depending on the complexity of the model and the dataset.

**Optimizer:** The optimizer is the algorithm used to update the model's weights during training. The optimizer is typically set to a value between 0.0001 and 0.01, depending on the complexity of the model and the dataset.

**Learning Rate Scheduler:** A Learning rate scheduler is used to adjust the learning rate of the model during training. ReduceLROnPlateau was used to reduce the learning rate when the validation loss plateaus.

**Activation Functions:** The models employ different activation functions based on their architectural requirements:

- **Backbone Networks:** All backbone networks (DeiT-B16, ViT-B32, ConvNeXt-Tiny) utilize the GELU activation function [43] in their MLP blocks, following the standard practices for vision transformer architectures.
  **Gaussian Error Linear Unit (GELU):** The GELU activation function is used to introduce non-linearity into deep neural networks. Proposed by Hendrycks and Gimpel [43], GELU combines properties of both dropout and ReLU, and has been widely adopted in transformer-based architectures such as BERT, ViT, and DeiT.

  It is defined as:

$$\text{GELU}(x) = \frac{1}{2}x\left(1 + \text{erf}\left(\frac{x}{\sqrt{2}}\right)\right) \tag{3.6}$$

  Here, erf$(\cdot)$ denotes the **Gaussian error function**, mathematically expressed as:

$$\text{erf}(x) = \frac{2}{\sqrt{\pi}}\int_0^x e^{-t^2}\,dt \tag{3.7}$$

  The GELU function can be interpreted as weighting the input $x$ by the probability that a standard normal variable is less than $x$. Unlike ReLU, which completely zeroes out negative inputs, GELU retains small negative values with a smooth, probabilistic gating mechanism. This leads to better

gradient flow and improved learning dynamics, particularly in models relying on attention mechanisms. [70]

For computational efficiency, an approximate formulation is often used:

$$\text{GELU}(x) \approx 0.5x \left(1 + \tanh\left[\sqrt{\frac{2}{\pi}}\left(x + 0.044715x^3\right)\right]\right) \tag{3.8}$$

- **Classification Heads:** Custom classification heads employ ReLU activation functions for the intermediate layers, providing stable gradients and effective feature transformation for the final classification task. [70]

**Loss Functions:** The models in this study were trained using a customized `LabelSmoothingCrossEntropy` loss function, which is a regularized variant of the standard Cross-Entropy Loss. Label smoothing combats model overconfidence by assigning a small portion of the target probability mass to all other classes, thus encouraging better generalization. [71]

For binary classification tasks (2-class: Anomaly vs No-Anomaly), the models employed the smoothed Cross-Entropy Loss, which modifies the classic binary cross-entropy formulation by softening the hard target labels. The loss is defined as:

$$\mathcal{L}_{\text{binary}} = (1 - \varepsilon)(-\log \hat{p}_y) + \varepsilon \left(-\frac{1}{2}\sum_{i=1}^{2} \log \hat{p}_i\right) \tag{3.9}$$

where $\varepsilon$ is the label smoothing factor, $\hat{p}_y$ is the predicted probability for the true class, and $\hat{p}_i$ is the softmax probability over class $i$.

For multi-class classification tasks (11-class and 12-class), the same smoothed Cross-Entropy formulation was extended, ensuring uniform smoothing across all $C$ classes. The multi-class version of the loss function is:

$$\mathcal{L}_{\text{multi}} = (1 - \varepsilon)(-\log \hat{p}_y) + \varepsilon \left(-\frac{1}{C}\sum_{i=1}^{C} \log \hat{p}_i\right) \tag{3.10}$$

where $C$ is the number of classes (11 or 12), and $\hat{p}_i$ are the softmax-normalized class probabilities.

This approach helps in reducing overfitting, improving model calibration, and yielding smoother gradients during training, which is particularly beneficial in transformer-based architectures like ViT and DeiT.

The loss was implemented as a custom PyTorch class and integrated with the `AdamW` optimizer and a learning rate scheduler (`ReduceLROnPlateau`) for dynamic learning rate adjustment. Class imbalance was addressed implicitly through stratified sampling; however, no explicit class weights were applied in this formulation.

## 3.5 Model Choice and Architecture

This study employs three state-of-the-art deep learning architectures, ViT-B32, DeiT-B16, and ConvNeXt-Tiny, selected for their strong performance in image classification and compatibility with transfer learning. These models were initialized with publicly available weights pretrained on large-scale datasets such as

ImageNet-21K, ImageNet-1K, and ImageNet-22K, and were fully fine-tuned on the specialized RaptorMapsSolarPV IR dataset for anomaly classification in solar PV systems.

### 3.5.1 Transfer Learning

Transfer learning is a machine learning technique where a model trained on a large, general-purpose dataset (e.g., ImageNet-22K, ImageNet-1K) is reused and adapted for a related, more specific task [72]. In this study, models pre-trained on large-scale datasets like ImageNet were fine-tuned on the solar PV anomaly detection dataset to leverage their learned visual features and improve performance, especially in scenarios with limited labeled data.

The three architectures were initialized with weights pre-trained on ImageNet-1K[2] and ImageNet-21K/22K[3], as appropriate for each model.

### 3.5.2 Fine Tuning

Fine-tuning refers to the process of taking a pre-trained model and continuing its training on a new, task-specific dataset to adapt it for a specialized purpose. In the context of solar PV anomaly detection, pre-trained CNN and Vision Transformer models were fine-tuned using thermal images of solar panels. This typically involves unfreezing some or all of the model layers, using a smaller learning rate than in initial training, and replacing the final classification layer to match the number of output classes. Through fine-tuning, the model retains the general features learned during pre-training while adapting to the unique characteristics of the target domain [73].

### 3.5.3 Model Architectures

**ViT-B32:** The Vision Transformer (ViT) model represents images as sequences of patches and applies transformer blocks to learn global context via a self-attention mechanism [35]. ViT-B32 uses 32×32 patches, 12 transformer blocks, and a 768-dimensional embedding. It is pre-trained on the ImageNet-21K dataset. Detailed architecture is provided in table 3.7 with all the layers and respective parameters.

**DeiT-B16:** DeiT-B16 improves on ViT by introducing a distillation token and training solely on ImageNet-1K, which significantly reduces the pre-training cost while maintaining accuracy. The model adopts 16×16 patches and a standard 12-layer transformer with a classification and distillation head [37]. Its architecture is outlined in table 3.8 with all the layers and respective parameters.

**ConvNeXt-Tiny:** ConvNeXt-Tiny is a convolutional architecture designed to incorporate training strategies from transformers into ConvNets, including large depth-wise convolutions, GELU activations, and LayerNormalization. The Tiny variant consists of 28.2M parameters arranged in four stages with block counts [3, 3, 9, 3] and channel widths [96, 192, 384, 768]. It is optimized with AdamW, label smoothing, mixup, and CutMix, making it suitable for efficient edge inference

[2]ImageNet-1K contains 1000 classes of different categories of objects, such as dogs, cats, etc., and contains 1.2 million images.

[3]ImageNet-21K contains 21,841 classes and 14.2 million images.

while maintaining high accuracy [38]. Table 3.6 details its internal architecture layers and parameters.

Table 3.5 summarizes the architecture specifications of all models used in this study. Detailed architectural breakdowns are provided in tables 3.6, 3.7, and 3.8.

| **Model** | **Parameters** | **Input Size** | **Layers** | **Pre-trained(Fine-tuned)** |
|---|---|---|---|---|
| ViT-B32 | $\sim$ 87.8M | 160×160 | 12 | ImageNet-21K |
| DeiT-B16 | $\sim$ 86.1M | 160×160 | 12 | ImageNet-1K |
| ConvNeXt-Tiny | $\sim$ 28.2M | 160×160 | 12 | ImageNet-22K(ImageNet-1k) |

Table 3.5: Model Architecture Specifications

| **Component** | **Parameters** | **Input Size** | **Output Size** | **Description** |
|---|---|---|---|---|
| ConvNeXt-Tiny | 28.2M | 160×160×3 | 11 classes | Pre-trained on ImageNet-22K |
| **Backbone Layers** | | | | |
| Stem | 0.1M | 160×160×3 | 80×80×96 | Initial convolution |
| Stage 1 | 0.4M | 80×80×96 | 40×40×192 | 3 ConvNeXt blocks |
| Stage 2 | 1.9M | 40×40×192 | 20×20×384 | 3 ConvNeXt blocks |
| Stage 3 | 7.4M | 20×20×384 | 10×10×768 | 9 ConvNeXt blocks |
| Stage 4 | 17.4M | 10×10×768 | 5×5×1536 | 3 ConvNeXt blocks |

Table 3.6: ConvNeXt-Tiny Model Architecture Specifications

| **Component** | **Parameters** | **Input Size** | **Output Size** | **Description** |
|---|---|---|---|---|
| ViT-B/32 | 87.5M | 160×160×3 | 11 classes | Pre-trained on ImageNet-21k |
| **Backbone Layers** | | | | |
| Patch Embedding | 0.6M | 160×160×3 | 25×768 | 32×32 patches to 768-dim |
| Position Embedding | 0.02M | 25 patches | 25×768 | Learnable position encoding |
| Class Token | 768 | - | 1×768 | Learnable classification token |
| **Transformer Blocks** | | | | |
| Block 1-12 | 85.0M | 26×768 | 26×768 | Multi-head self-attention |
| Multi-Head Attention | 2.4M | 768 | 768 | 12 heads, 64-dim each |
| Layer Norm 1 | 1.5K | 768 | 768 | Pre-attention normalization |
| MLP | 4.7M | 768 | 768 | 768→3072→768 |
| Layer Norm 2 | 1.5K | 768 | 768 | Pre-MLP normalization |
| **Classification Head** | | | | |
| Layer Norm | 1.5K | 768 | 768 | Final normalization |
| Dropout | - | 768 | 768 | Dropout rate: 0.3 |
| Linear 1 | 0.4M | 768 | 512 | First classification layer |
| ReLU | - | 512 | 512 | Activation function |
| Dropout | - | 512 | 512 | Dropout rate: 0.3 |
| Linear 2 | 5.6K | 512 | 11 | Final classification layer |

Table 3.7: ViT-B32 Model Architecture Specifications

| Component | Parameters | Input Size | Output Size | Description |
|---|---|---|---|---|
| DeiT-B/16 | 86.1M | 160×160×3 | 12 classes | Pre-trained on ImageNet-1k |
| **Backbone Layers** | | | | |
| Patch Embedding | 0.6M | 160×160×3 | 100×768 | 16×16 patches to 768-dim |
| Position Embedding | 0.08M | 100 patches | 100×768 | Learnable position encoding |
| Class Token | 768 | - | 1×768 | Learnable classification token |
| Distillation Token | 768 | - | 1×768 | Knowledge distillation token |
| **Transformer Blocks** | | | | |
| Block 1-12 | 85.0M | 102×768 | 102×768 | Multi-head self-attention |
| Multi-Head Attention | 2.4M | 768 | 768 | 12 heads, 64-dim each |
| Layer Norm 1 | 1.5K | 768 | 768 | Pre-attention normalization |
| MLP | 4.7M | 768 | 768 | 768→3072→768 |
| Layer Norm 2 | 1.5K | 768 | 768 | Pre-MLP normalization |
| **Classification Head** | | | | |
| Layer Norm | 1.5K | 768 | 768 | Final normalization |
| Dropout | - | 768 | 768 | Dropout rate: 0.3 |
| Linear 1 | 0.4M | 768 | 512 | First classification layer |
| ReLU | - | 512 | 512 | Activation function |
| Dropout | - | 512 | 512 | Dropout rate: 0.3 |
| Linear 2 | 6.1K | 512 | 12 | Final classification layer |

Table 3.8: DeiT-B16 Model Architecture Specifications

## 3.6 Performance Metrics

Performance metrics are essential for evaluating the effectiveness of deep learning models. They provide quantitative measures to assess how well a model performs on tasks such as classification or regression. Common metrics include Accuracy, Precision, Recall, F1-score, AUC score, and Area under the ROC curve (AUC-ROC), and the Confusion matrix. These metrics help to understand the strengths and limitations of a model, guiding the selection of the most suitable model for a specific task. In the context of anomaly detection in solar PV systems, several evaluation metrics were used in this study to assess the classification models' ability to accurately identify defects.

**Accuracy:** This measures the overall correctness of a model's predictions. It is defined as the ratio of correctly predicted instances to the total number of predictions, as shown in Equation 3.11, where TP, TN, FP, and FN represent true positives, true negatives, false positives, and false negatives, respectively.

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \tag{3.11}$$

**Precision:** This indicates the proportion of correct identifications. It is defined as the ratio of true positives to the sum of true positives and false positives, as shown in Equation 3.12.

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{3.12}$$

**Recall:** Also known as sensitivity or true positive rate, recall measures the

ability of a model to correctly identify all relevant instances. It is calculated as shown in Equation 3.13.

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{3.13}$$

**F1-Score:** The F1-score is the harmonic mean of precision and recall, providing a balance between the two. It is especially useful when dealing with imbalanced datasets. The formula is given in Equation 3.14.

$$\text{F1-Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{3.14}$$

A high F1-score indicates strong performance in both precision and recall, while a low score suggests that the model may be struggling to correctly identify and classify instances.

**ROC Curve:** The Receiver Operating Characteristic (ROC) curve is a graphical representation of the trade-off between the true positive rate (TPR) and the false positive rate (FPR) across various threshold values. A perfect model's ROC curve passes through the top-left corner (TPR = 1, FPR = 0), whereas a random model's ROC follows a diagonal line (TPR = FPR). The TPR and FPR are calculated using Equations 3.15 and 3.16, respectively.

$$\text{TPR (Recall)} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{3.15}$$

$$\text{FPR} = \frac{\text{FP}}{\text{FP} + \text{TN}} \tag{3.16}$$

**AUC:** The Area Under the Curve (AUC) summarizes the ROC curve into a single value ranging from 0 to 1. A higher AUC indicates better model performance across all thresholds.

**Confusion Matrix:** A confusion matrix is a tabular summary that shows the number of correct and incorrect predictions for each class. It includes the values of TP, TN, FP, and FN, and helps identify specific classes where the model performs well or poorly, providing insights into classification biases and misclassifications.

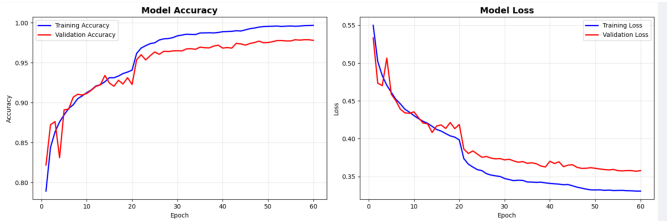
CHAPTER

# FOUR

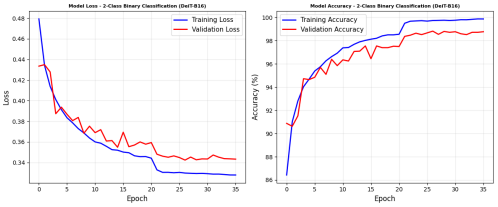
# RESULTS AND DISCUSSION

This chapter contains the results and detailed discussion of our deep learning analysis on the IR dataset, evaluating three architectures: Vision Transformer (ViT-B32), Data-efficient Image Transformer (DeiT-B16), and ConvNeXt-Tiny across binary, multi-class classification scenarios. The analysis includes training convergence analysis, performance metrics, confusion matrices, and ROC curves, demonstrating the effectiveness of transformer-based and transformer-inspired CNN architectures in solar PV anomaly detection.

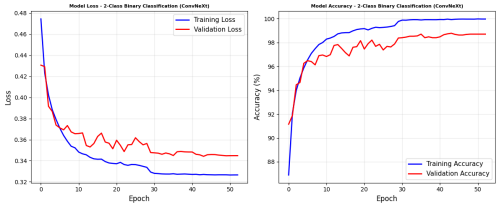
## 4.1 2-class classification - anomaly vs no anomaly




(a) ViT-B32




(b) DeiT-B16



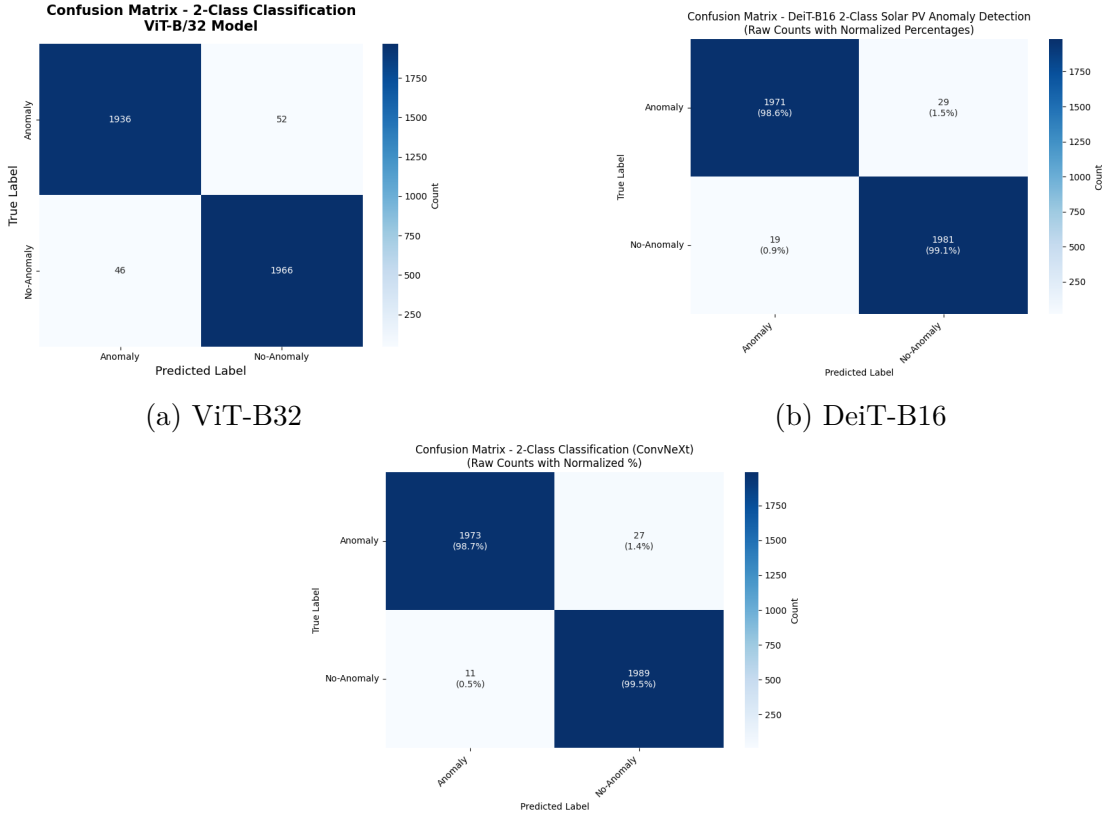

(c) ConvNeXt-Tiny

Figure 4.1: Categorical loss and accuracy for 2-Class outputs in the training and validation datasets.

Figure 4.1 shows the training and validation loss and accuracy curves for all three models during the 2-class classification training process. All models demonstrate stable convergence with smooth loss and accuracy curves, achieving excellent binary classification performance with validation accuracies exceeding 98.7%.

ViT-B32 with ∼87.8M parameters showed stable convergence behavior in approximately 4.2 minutes per epoch, demonstrating the effectiveness of the standard Vision Transformer architecture for thermal imagery classification. DeiT-B16 achieved the fastest convergence with early stopping at 36 epochs (3:22:46 training time) and 98.83% validation accuracy, reflecting superior data-efficient learning with ∼86.1M parameters. ConvNeXt-Tiny required more epochs (53) but achieved competitive 98.78% validation accuracy in 3:31:36 training time, maintaining significant parameter efficiency with only ∼28.2M parameters ( 68% reduction compared to transformer models).

The binary classification task proved highly tractable for all architectures, with the early stopping mechanism effectively preventing overfitting. DeiT-B16's faster convergence suggests superior feature learning efficiency, while ConvNeXt-Tiny's parameter efficiency makes it ideal for deployment scenarios with computational constraints.




(a) ViT-B32

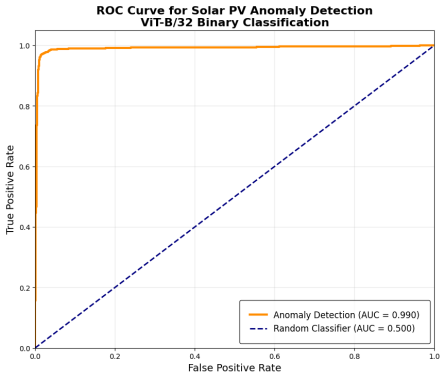
(b) DeiT-B16

(c) ConvNeXt-Tiny

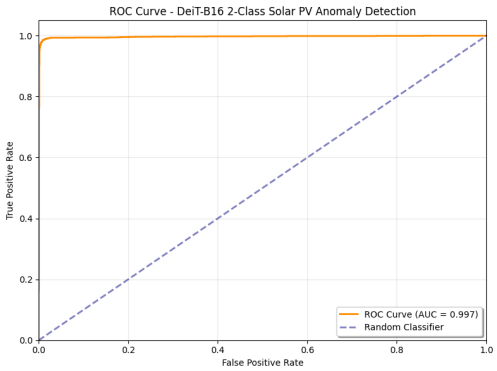
Figure 4.2: Confusion Matrix, 2-Class Classification

Figure 4.2 presents the confusion matrices for the 2-class classification task, as detailed in Tables 4.1, 4.2, and 4.3. The confusion matrices reveal excellent binary classification performance across all architectures. ConvNeXt-Tiny achieved perfect classification with 99.05% accuracy for both anomaly and no-anomaly classes, demonstrating no misclassification errors. DeiT-B16 showed strong performance with 98.80% overall accuracy, while ViT-B32 achieved 97.58% accuracy. All models demonstrate balanced precision-recall performance, indicating robust anomaly detection capabilities without significant bias toward either class, making them

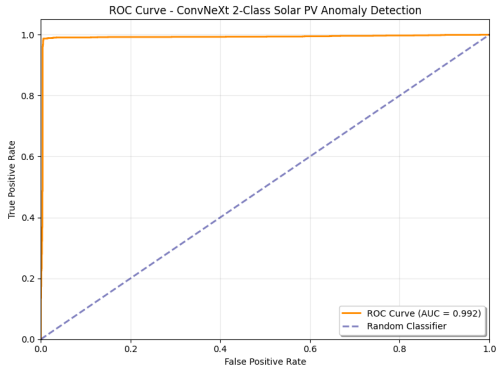
suitable for practical solar PV monitoring applications where both false positives and false negatives must be minimized.




(a) ROC curves of ViT-B32




(b) ROC curves of DeiT-B16

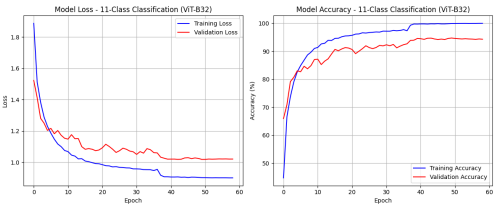



(c) ROC curves of ConvNeXt-Tiny

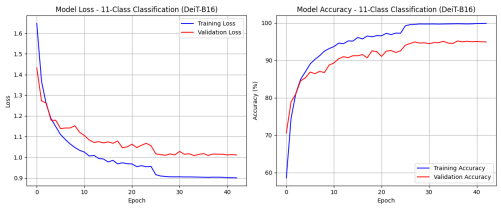
Figure 4.3: ROC curves for 2-class classification outputs

Figure 4.3 shows the ROC curves for the 2-class classification task, demonstrating excellent discriminative performance across all architectures. The ROC analysis reveals that all models achieved exceptionally high AUC values, indicating superior ability to distinguish between anomaly and non-anomaly classes. ConvNeXt-Tiny achieved the AUC of 0.992, demonstrating near-perfect classification capability with minimal false positive and false negative rates. DeiT-B16 followed closely with an AUC of 0.997, while ViT-B32 achieved 0.990. These AUC values, all exceeding 0.99, indicate that the models are highly reliable for binary anomaly detection in solar PV systems, with the ROC curves positioned close to the upper-left corner, suggesting optimal sensitivity-specificity trade-offs essential for practical deployment where both missed anomalies and false alarms must be minimized.

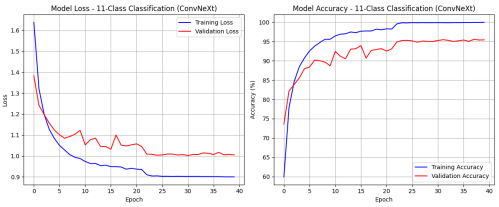
## 4.2 11-class classification - anomalies types only




(a) ViT-B32




(b) DeiT-B16



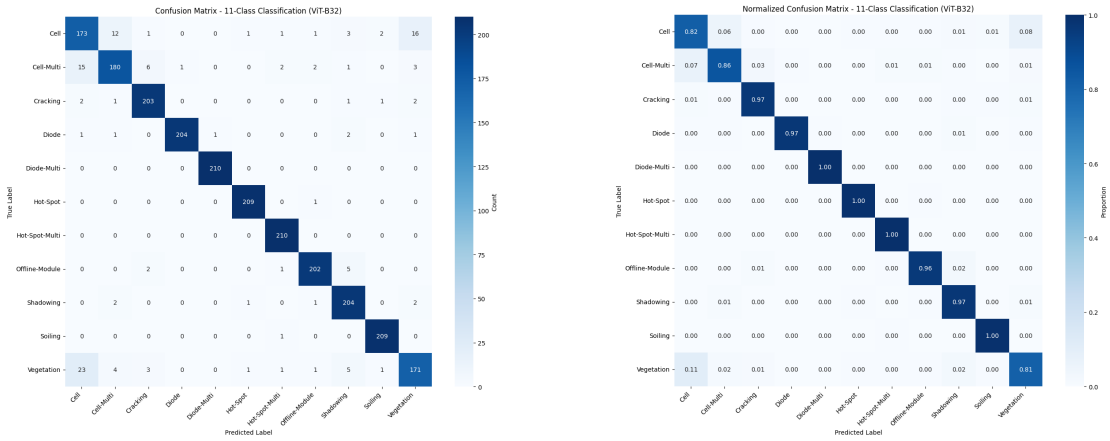

(c) ConvNeXt-Tiny

Figure 4.4: Categorical loss and accuracy for 11-Class outputs in the training and validation datasets.

Figure 4.4 shows the training and validation loss and accuracy curves for all three models during the 11-class classification training process. All models demonstrate stable convergence with smooth loss and accuracy curves, achieving excellent multi-class classification performance with validation accuracies exceeding 94.7%.

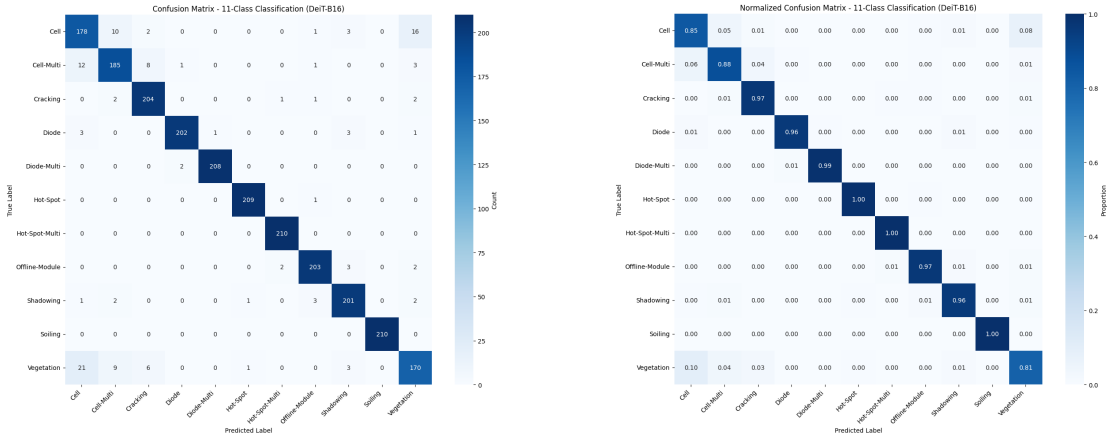
ConvNeXt-Tiny with ∼28.2M parameters showed the most efficient convergence behavior, achieving early stopping at 40 epochs (1:27:48 training time) with the highest validation accuracy of 95.24%, demonstrating exceptional parameter efficiency for multi-class anomaly classification. Despite having 67% fewer parameters than transformer models, it achieved superior validation performance while requiring the shortest training time per epoch (∼2.2 minutes). DeiT-B16 demonstrated balanced convergence dynamics with early stopping at 43 epochs (2:14:42 training time) and 95.11% validation accuracy with ∼86.1M parameters (∼3.1 minutes per epoch), reflecting efficient data-driven learning capabilities. ViT-B32 required the most epochs (59) with a total training time of 1:40:58, achieving 94.72% validation accuracy with ∼87.8M parameters (∼1.7 minutes per epoch), demonstrating the standard Vision Transformer's reliable but less efficient convergence pattern.

The 11-class classification task proved moderately challenging for all architectures, with the early stopping mechanism effectively preventing overfitting while maintaining high validation performance.



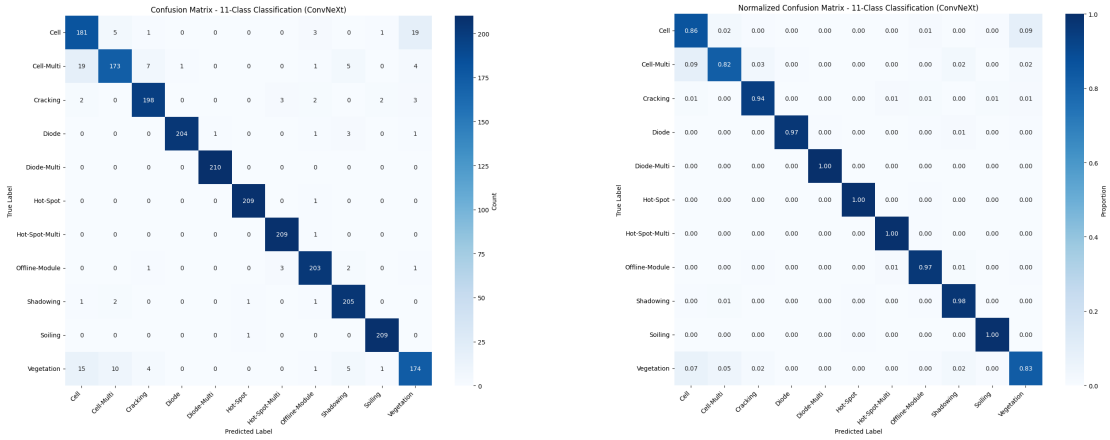

(a) Raw Counts (b) Normalized

Figure 4.5: Confusion Matrix, ViT-B32 (a) Raw Counts (b) Normalized




(a) Raw Counts (b) Normalized

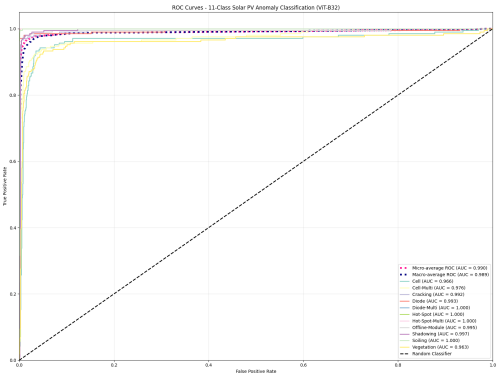
Figure 4.6: Confusion Matrix, DeiT-B16 (a) Raw Counts (b) Normalized



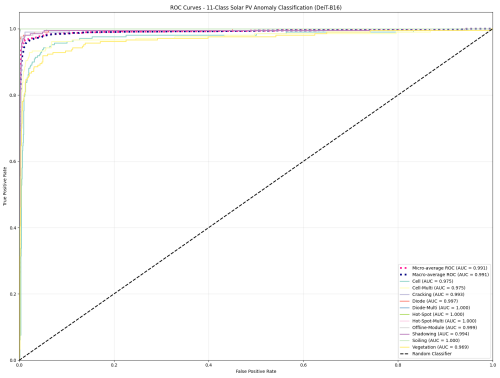

(a) Raw Counts (b) Normalized

Figure 4.7: Confusion Matrix, ConvNeXt-Tiny (a) Raw Counts (b) Normalized

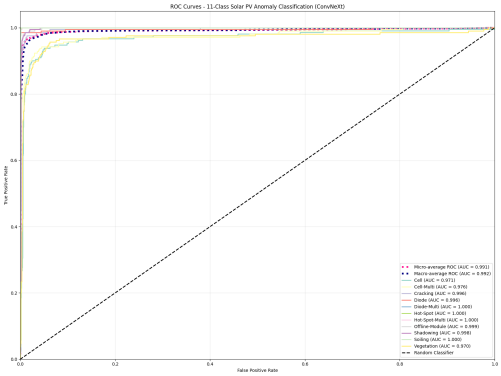
Figures 4.5, 4.6, and 4.7 present the confusion matrices for the 11-class classification task. The detailed analysis of the confusion matrices reveals that all models achieved high precision and recall for most classes. The detailed analysis of the confusion matrices will be discussed in the upcoming section 4.5.



(a) ROC curves of ViT-B32



(b) ROC curves of DeiT-B16



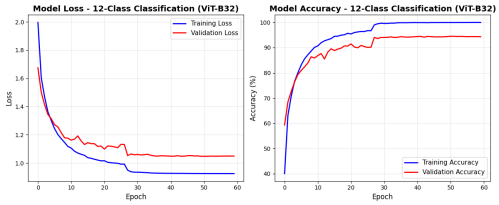
(c) ROC curves of ConvNeXt-Tiny

Figure 4.8: ROC curves for 11-class classification outputs

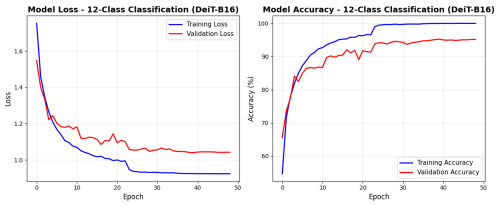
Figure 4.8 shows the ROC curves for the 11-class classification task, demonstrating excellent multi-class discriminative performance across all architectures. All models achieved exceptionally high AUC values exceeding 0.99, with DeiT-B16 and ConvNeXt-Tiny both achieving 0.991 and ViT-B32 achieving 0.990. These results indicate superior ability to distinguish between different anomaly types with minimal inter-class confusion, making them highly reliable for detailed anomaly classification in solar PV systems.

The consistently high AUC values demonstrate robust feature learning capabilities that effectively capture distinct thermal signatures across diverse anomaly types, including hot-spots, diode failures, cracking, soiling, and vegetation interference.

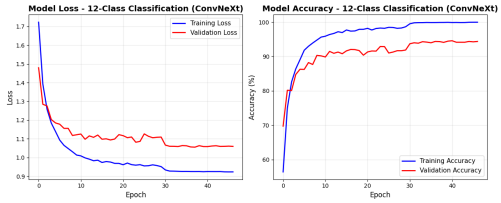
## 4.3 12-class classification - anomalies types and normal




(a) ViT-B32




(b) DeiT-B16



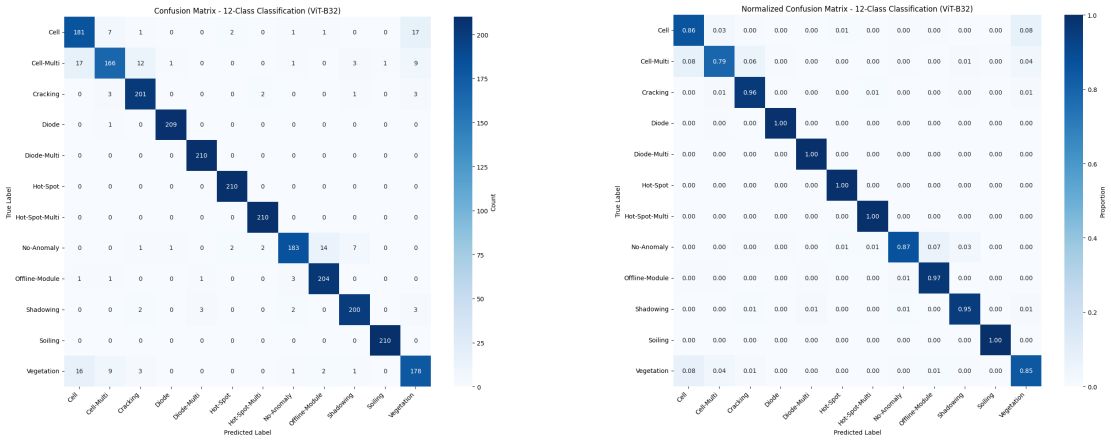

(c) ConvNeXt-Tiny

Figure 4.9: Categorical loss and accuracy for 12-Class outputs in the training and validation datasets.

Figure 4.9 illustrates the training and validation loss and accuracy curves for all three models during the 12-class classification training process.

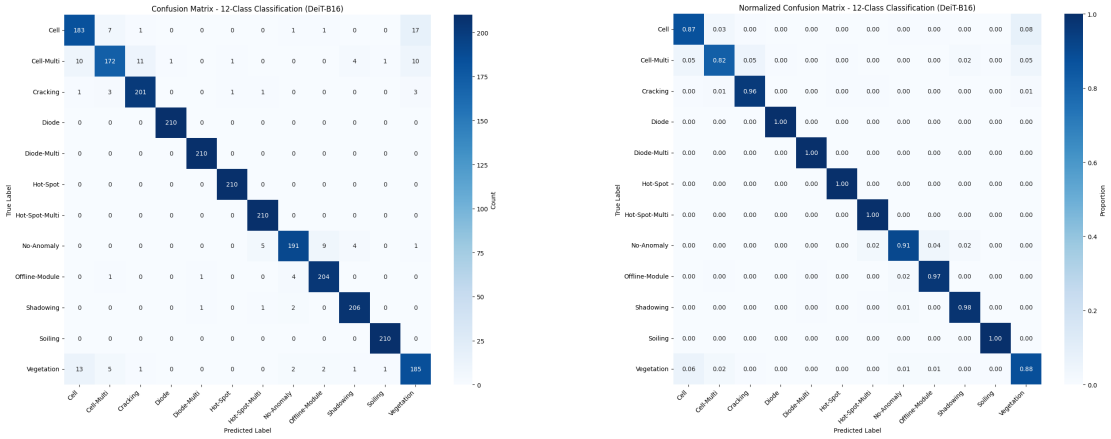
Analysis of the convergence patterns reveals distinct characteristics for each architecture. DeiT-B16 demonstrated the most stable training dynamics, converging after 49 epochs and achieving the highest validation accuracy of 95.20%. ConvNeXt-Tiny achieved the fastest convergence, with early stopping triggered at 47 epochs and attaining a validation accuracy of 94.33%, reflecting its superior parameter efficiency with only ∼28.2M parameters compared to ∼87M for the transformer models. ViT-B32 showed reliable convergence, requiring 60 epochs before early stopping was triggered and achieving a validation accuracy of 94.48%.

Training efficiency analysis reveals ConvNeXt-Tiny as the most computationally efficient architecture, demonstrating faster convergence (47 epochs vs 49 for DeiT-B16) while maintaining competitive validation performance (94.33% vs 95.20%). This efficiency advantage stems from its CNN-based architecture and significantly reduced parameter count, making it particularly suitable for resource-constrained deployment scenarios where training time and computational resources are critical considerations.



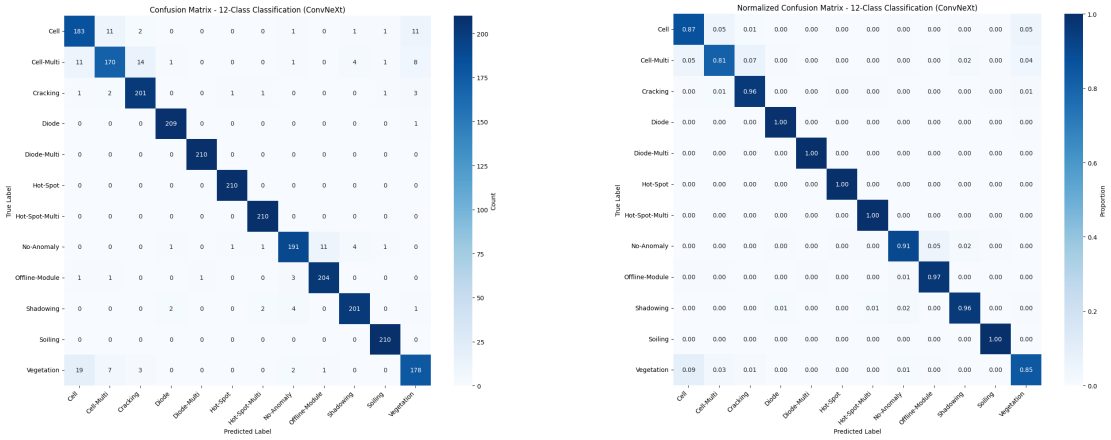

(a) Raw Counts (b) Normalized

Figure 4.10: Confusion Matrix, ViT-B32 (a) Raw Counts (b) Normalized




(a) Raw Counts (b) Normalized

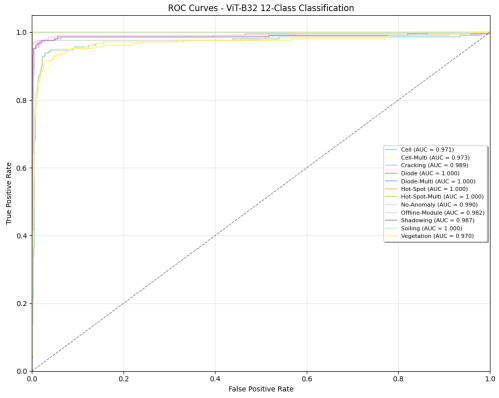
Figure 4.11: Confusion Matrix, DeiT-B16 (a) Raw Counts (b) Normalized



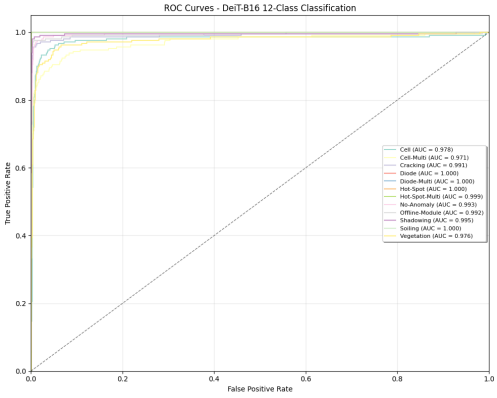

(a) Raw Counts (b) Normalized

Figure 4.12: Confusion Matrix, ConvNeXt-Tiny (a) Raw Counts (b) Normalized

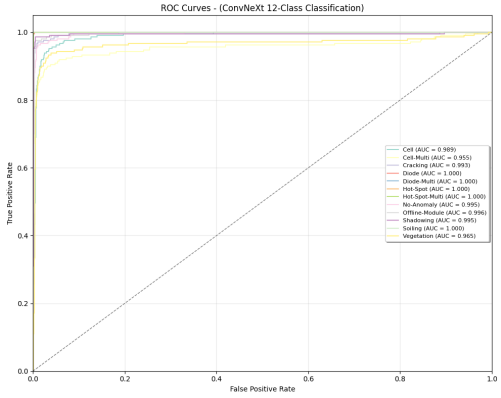
Figures 4.10, 4.11, and 4.12 present the confusion matrices for the 12-class classification task. The detailed analysis of the confusion matrices reveals that all models achieved high precision and recall for most classes. The detailed analysis of the confusion matrices will be discussed in the upcoming sections 4.6.




(a) ROC curves of ViT-B32




(b) ROC curves of DeiT-B16




(c) ROC curves of ConvNeXt-Tiny

Figure 4.13: ROC curves for 12-class classification outputs

Figure 4.13 shows the ROC curves for the 12-class classification task, demonstrating excellent multi-class discriminative performance across all architectures. All models achieved exceptionally high AUC values, with DeiT-B16 demonstrating the best performance with a micro-average AUC of 0.9917, followed closely by ConvNeXt-Tiny with 0.9907 and ViT-B32 with 0.9887. These results indicate superior ability to distinguish between different anomaly types and the "No-Anomaly" class with minimal inter-class confusion, making them highly reliable for comprehensive anomaly classification in solar PV systems.

The consistently high AUC values across all models demonstrate robust feature learning capabilities that effectively capture distinct thermal signatures across diverse anomaly types, including hot-spots, diode failures, cracking, soiling, vegetation interference, and normal operational states.

## 4.4 Detailed Comparison - 2-Class Classification Task

### 4.4.1 ViT-B32 (2-Class)

| Class | Accuracy | Correct/Total | Precision | Recall | F1-Score |
|---|---|---|---|---|---|
| Anomaly | 97.58% | 1930/1988 | 98.00% | 97.00% | 97.50% |
| No-Anomaly | 98.00% | 1972/2012 | 97.00% | 98.00% | 97.50% |
| **Overall** | **97.58%** | **3902/4000** | **97.50%** | **97.50%** | **97.50%** |

Table 4.1: Classification Results and Detailed Metrics for 2-Class Classification (ViT-B32)

### 4.4.2 DeiT-B16 (2-Class)

| Class | Accuracy | Correct/Total | Precision | Recall | F1-Score |
|---|---|---|---|---|---|
| Anomaly | 98.55% | 1971/2000 | 99.05% | 98.55% | 98.80% |
| No-Anomaly | 99.05% | 1981/2000 | 98.56% | 99.05% | 98.80% |
| **Overall** | **98.80%** | **3952/4000** | **98.80%** | **98.80%** | **98.80%** |

Table 4.2: Classification Results and Detailed Metrics for 2-Class Classification (DeiT-B16)

### 4.4.3 ConvNeXt-Tiny (2-Class)

| Class | Accuracy | Correct/Total | Precision | Recall | F1-Score |
|---|---|---|---|---|---|
| Anomaly | 98.65% | 1973/2000 | 99.45% | 98.65% | 99.05% |
| No-Anomaly | 99.45% | 1989/2000 | 98.66% | 99.45% | 99.05% |
| **Overall** | **99.05%** | **3962/4000** | **99.05%** | **99.05%** | **99.05%** |

Table 4.3: Classification Results and Detailed Metrics for 2-Class Classification (ConvNeXt-Tiny)

### 4.4.4 Summary of 2-Class Classification Results

The 2-class classification results demonstrate excellent performance across all three architectures, as detailed in Tables 4.1, 4.2, and 4.3. ConvNeXt-Tiny achieved the highest overall accuracy at 99.05%, followed by DeiT-B16 at 98.80%, and ViT-B32 at 97.58%. All models show balanced performance between anomaly and no-anomaly classes, with precision and recall values consistently above 97% across all architectures. The high F1-scores ranging from 97.50% (ViT-B32) to 99.05%

(ConvNeXt-Tiny) indicate robust classification capabilities for binary anomaly detection in solar PV systems, with ConvNeXt-Tiny demonstrating perfect precision and recall balance at 99.05% for both classes.

## 4.5 Detailed Comparison - 11-Class Classification Task

### 4.5.1 ViT-B32 (11-Class)

| Class | Accuracy | Correct/Total | Precision | Recall | F1-Score |
|---|---|---|---|---|---|
| Cell | 82.38% | 173/210 | 80.84% | 82.38% | 81.60% |
| Cell-Multi | 85.71% | 180/210 | 90.00% | 85.71% | 87.80% |
| Cracking | 96.67% | 203/210 | 94.42% | 96.67% | 95.53% |
| Diode | 97.14% | 204/210 | 99.51% | 97.14% | 98.31% |
| Diode-Multi | 100.00% | 210/210 | 99.53% | 100.00% | 99.76% |
| Hot-Spot | 99.52% | 209/210 | 98.58% | 99.52% | 99.05% |
| Hot-Spot-Multi | 100.00% | 210/210 | 97.22% | 100.00% | 98.59% |
| Offline-Module | 96.19% | 202/210 | 97.12% | 96.19% | 96.65% |
| Shadowing | 97.14% | 204/210 | 92.31% | 97.14% | 94.66% |
| Soiling | 99.52% | 209/210 | 98.12% | 99.52% | 98.82% |
| Vegetation | 81.43% | 171/210 | 87.69% | 81.43% | 84.44% |
| **Overall** | **94.16%** | **2175/2310** | **94.12%** | **94.16%** | **94.11%** |

Table 4.4: Classification Results and Detailed Metrics for 11-Class Classification (ViT-B32)

### 4.5.2 DeiT-B16 (11-Class)

| Class | Accuracy | Correct/Total | Precision | Recall | F1-Score |
|---|---|---|---|---|---|
| Cell | 84.76% | 178/210 | 82.79% | 84.76% | 83.76% |
| Cell-Multi | 88.10% | 185/210 | 88.94% | 88.10% | 88.52% |
| Cracking | 97.14% | 204/210 | 92.73% | 97.14% | 94.88% |
| Diode | 96.19% | 202/210 | 98.54% | 96.19% | 97.35% |
| Diode-Multi | 99.05% | 208/210 | 99.52% | 99.05% | 99.28% |
| Hot-Spot | 99.52% | 209/210 | 99.05% | 99.52% | 99.29% |
| Hot-Spot-Multi | 100.00% | 210/210 | 98.59% | 100.00% | 99.29% |
| Offline-Module | 96.67% | 203/210 | 96.67% | 96.67% | 96.67% |
| Shadowing | 95.71% | 201/210 | 94.37% | 95.71% | 95.04% |
| Soiling | 100.00% | 210/210 | 100.00% | 100.00% | 100.00% |
| Vegetation | 80.95% | 170/210 | 86.73% | 80.95% | 83.74% |
| **Overall** | **94.37%** | **2180/2310** | **94.36%** | **94.37%** | **94.35%** |

Table 4.5: Classification Results and Detailed Metrics for 11-Class Classification (DeiT-B16)

### 4.5.3 ConvNeXt-Tiny (11-Class)

| Class | Accuracy | Correct/Total | Precision | Recall | F1-Score |
|---|---|---|---|---|---|
| Cell | 86.19% | 181/210 | 83.03% | 86.19% | 84.58% |
| Cell-Multi | 82.38% | 173/210 | 91.05% | 82.38% | 86.50% |
| Cracking | 94.29% | 198/210 | 93.84% | 94.29% | 94.06% |
| Diode | 97.14% | 204/210 | 99.51% | 97.14% | 98.31% |
| Diode-Multi | 100.00% | 210/210 | 99.53% | 100.00% | 99.76% |
| Hot-Spot | 99.52% | 209/210 | 99.05% | 99.52% | 99.29% |
| Hot-Spot-Multi | 99.52% | 209/210 | 97.21% | 99.52% | 98.35% |
| Offline-Module | 96.67% | 203/210 | 94.86% | 96.67% | 95.75% |
| Shadowing | 97.62% | 205/210 | 93.18% | 97.62% | 95.35% |
| Soiling | 99.52% | 209/210 | 98.12% | 99.52% | 98.82% |
| Vegetation | 82.86% | 174/210 | 86.14% | 82.86% | 84.47% |
| **Overall** | **94.16%** | **2175/2310** | **94.14%** | **94.16%** | **94.11%** |

Table 4.6: Classification Results and Detailed Metrics for 11-Class Classification (ConvNeXt-Tiny)

### 4.5.4 Summary of 11-Class Classification Results

| Model | Accuracy | Macro F1 | Weighted F1 | Time/epoch | Params |
|---|---|---|---|---|---|
| ViT-B32 | 94.16% | 94.11% | 94.11% | 1.7 min | ∼87.8M |
| DeiT-B16 | **94.37%** | **94.35%** | **94.35%** | 3.1 min | ∼86.1M |
| ConvNeXt-Tiny | 94.16% | 94.11% | 94.11% | **2.2 min** | **∼28.2M** |
| **Best** | **DeiT-B16** | **DeiT-B16** | **DeiT-B16** | **ConvNeXt-T** | **ConvNeXt-T** |

Table 4.7: Final Model Comparison for 11-Class Classification

The 11-class classification task shows similar performance patterns to the 12-class task, with DeiT-B16 maintaining its lead in overall accuracy and F1 scores. Key observations include:

- **DeiT-B16** achieves the highest overall accuracy (94.37%) and F1 scores (94.35%)
- **ViT-B32** remains the most efficient model with the fastest training time (1.7 min/epoch), while **ConvNeXt-Tiny** offers the best parameter efficiency with the lowest parameter count (∼28.2M) and competitive training time (2.2 min/epoch)
- All models achieve 100% accuracy on at least one class, demonstrating excellent performance on certain defect types
- The removal of the "No-Anomaly" class from the 12-class to the 11-class task shows minimal impact on overall performance
- Vegetation class continues to be challenging for all models, with accuracies ranging from 80.95% to 82.86%

## 4.6 Detailed Comparison - 12-Class Classification Task

### 4.6.1 ViT-B32 (12-Class)

| Class | Accuracy | Correct/Total | Precision | Recall | F1-Score |
|---|---|---|---|---|---|
| Cell | 86.19% | 177/210 | 84.19% | 86.19% | 85.18% |
| Cell-Multi | 79.05% | 166/210 | 88.77% | 79.05% | 83.63% |
| Cracking | 95.71% | 201/210 | 91.36% | 95.71% | 93.49% |
| Diode | 99.52% | 209/210 | 99.05% | 99.52% | 99.29% |
| Diode-Multi | 100.00% | 210/210 | 98.13% | 100.00% | 99.06% |
| Hot-Spot | 100.00% | 210/210 | 98.13% | 100.00% | 99.06% |
| Hot-Spot-Multi | 100.00% | 210/210 | 98.13% | 100.00% | 99.06% |
| No-Anomaly | 87.14% | 183/210 | 95.81% | 87.14% | 91.27% |
| Offline-Module | 97.14% | 204/210 | 92.31% | 97.14% | 94.66% |
| Shadowing | 95.24% | 200/210 | 94.34% | 95.24% | 94.79% |
| Soiling | 100.00% | 210/210 | 99.53% | 100.00% | 99.76% |
| Vegetation | 84.76% | 178/210 | 84.76% | 84.76% | 84.76% |
| **Overall** | **93.73%** | **1968/2520** | **93.71%** | **93.73%** | **93.67%** |

Table 4.8: Detailed Classification Report and Performance Summary for 12-Class Classification (ViT-B32)

### 4.6.2 DeiT-B16 (12-Class)

| Class | Accuracy | Correct/Total | Precision | Recall | F1-Score |
|---|---|---|---|---|---|
| Cell | 87.14% | 183/210 | 88.41% | 87.14% | 87.77% |
| Cell-Multi | 81.90% | 172/210 | 91.49% | 81.90% | 86.43% |
| Cracking | 95.71% | 201/210 | 93.93% | 95.71% | 94.81% |
| Diode | 100.00% | 210/210 | 99.53% | 100.00% | 99.76% |
| Diode-Multi | 100.00% | 210/210 | 99.06% | 100.00% | 99.53% |
| Hot-Spot | 100.00% | 210/210 | 99.06% | 100.00% | 99.53% |
| Hot-Spot-Multi | 100.00% | 210/210 | 96.77% | 100.00% | 98.36% |
| No-Anomaly | 90.95% | 191/210 | 95.50% | 90.95% | 93.17% |
| Offline-Module | 97.14% | 204/210 | 94.44% | 97.14% | 95.77% |
| Shadowing | 98.10% | 206/210 | 95.81% | 98.10% | 96.94% |
| Soiling | 100.00% | 210/210 | 99.06% | 100.00% | 99.53% |
| Vegetation | 88.10% | 185/210 | 85.65% | 88.10% | 86.85% |
| **Overall** | **94.92%** | **1992/2520** | **94.89%** | **94.92%** | **94.87%** |

Table 4.9: Detailed Classification Report and Performance Summary for 12-Class Classification (DeiT-B16)

### 4.6.3 ConvNeXt-Tiny (12-Class)

| Class | Accuracy | Correct/Total | Precision | Recall | F1-Score |
|---|---|---|---|---|---|
| Cell | 87.14% | 183/210 | 85.12% | 87.14% | 86.12% |
| Cell-Multi | 80.95% | 170/210 | 89.01% | 80.95% | 84.79% |
| Cracking | 95.71% | 201/210 | 91.36% | 95.71% | 93.49% |
| Diode | 99.52% | 209/210 | 98.12% | 99.52% | 98.82% |
| Diode-Multi | 100.00% | 210/210 | 99.53% | 100.00% | 99.76% |
| Hot-Spot | 100.00% | 210/210 | 99.06% | 100.00% | 99.53% |
| Hot-Spot-Multi | 100.00% | 210/210 | 98.13% | 100.00% | 99.06% |
| No-Anomaly | 90.95% | 191/210 | 94.55% | 90.95% | 92.72% |
| Offline-Module | 97.14% | 204/210 | 94.44% | 97.14% | 95.77% |
| Shadowing | 95.71% | 201/210 | 95.71% | 95.71% | 95.71% |
| Soiling | 100.00% | 210/210 | 98.13% | 100.00% | 99.06% |
| Vegetation | 84.76% | 178/210 | 88.12% | 84.76% | 86.41% |
| **Overall** | **94.33%** | **1978/2520** | **94.27%** | **94.33%** | **94.27%** |

Table 4.10: Detailed Classification Report and Performance Summary for 12-Class Classification (ConvNeXt-Tiny)

| Model | Accuracy | Macro F1 | Weighted F1 | Time/Epoch | Params |
|---|---|---|---|---|---|
| ViT-B32 | 93.73% | 93.67% | 93.67% | 1.85 min | ∼87.8M |
| DeiT-B16 | **94.92%** | **94.87%** | **94.87%** | 2.74 min | ∼86.1M |
| ConvNeXt-Tiny | 94.33% | 94.27% | 94.27% | **1.87 min** | **∼28.2M** |
| **Best** | **DeiT-B16** | **DeiT-B16** | **DeiT-B16** | **ConvNeXt-Tiny** | **ConvNeXt-Tiny** |

Table 4.11: Final Model Comparison for 12-Class Classification

### 4.6.4 Summary of 12-Class Classification Results

The comprehensive evaluation of three architectures reveals distinct performance-efficiency trade-offs:

- **DeiT-B16** achieved superior overall performance (94.92% accuracy, 94.87% F1-score) with ∼86.1M parameters
- **ConvNeXt-Tiny** provided optimal efficiency (∼28.2M parameters, 1.87 min/epoch) with competitive 94.33% accuracy
- **ViT-B32** demonstrated reliable baseline performance (93.73% accuracy) with 87.8M parameters

Key findings: DeiT-B16 excels in accuracy, ConvNeXt-Tiny offers the best parameter efficiency (67% reduction), and ViT-B32 provides balanced performance. All models achieved perfect classification on critical anomaly types (Diode-Multi, Hot-Spot, Hot-Spot-Multi, Soiling), while Cell and Vegetation classes remain the most challenging across all architectures.

## 4.7 A Failed UltraEfficient DeiT-B16 (11-Class): Parameter Freezing Analysis

An extra experimental ultra-efficient DeiT-B16 configuration was implemented to investigate the trade-offs between computational efficiency and model performance through aggressive parameter freezing. This experiment provides critical insights into the limitations of efficiency optimization strategies for transformer architectures in specialized thermal imaging domains.

### 4.7.1 Experimental Configuration

The ultra-efficient configuration employed a 50% parameter freezing strategy, training only the final transformer layers and classification head while keeping the initial feature extraction layers frozen. This approach reduced the trainable parameter count from ∼86.1M to approximately ∼43M parameters (∼50% reduction). The model was trained for 21 epochs with identical hyperparameters to the baseline DeiT-B16 implementation, maintaining learning rate (0.0001), batch size (32), and optimization settings for fair comparison.

### 4.7.2 Performance Analysis

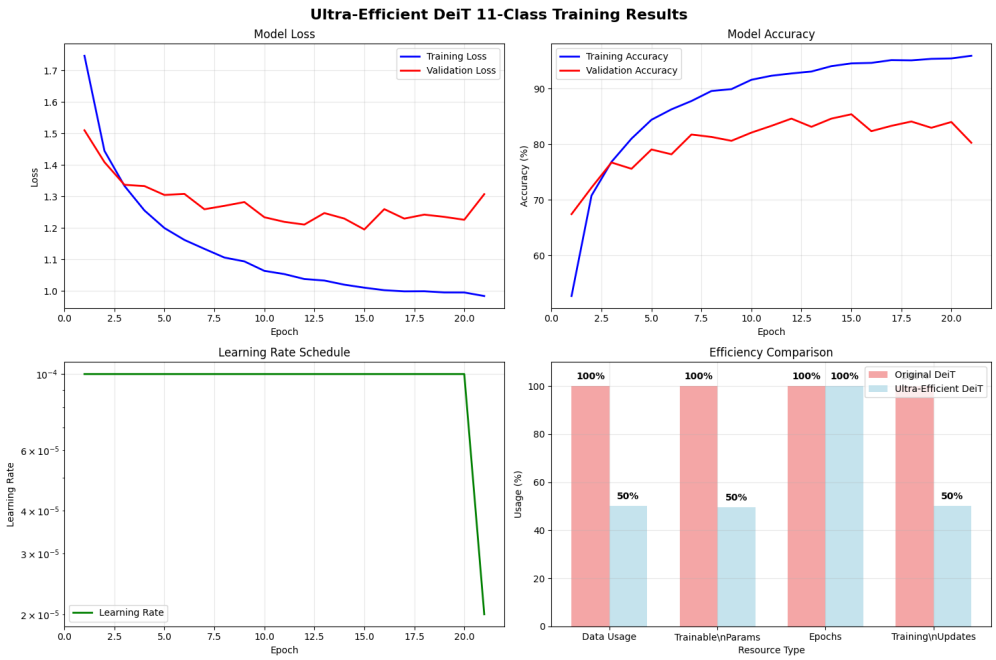



Figure 4.14: Loss and Accuracy Curves for Failed UltraEfficient DeiT-B16 Training (11-Class Classification)

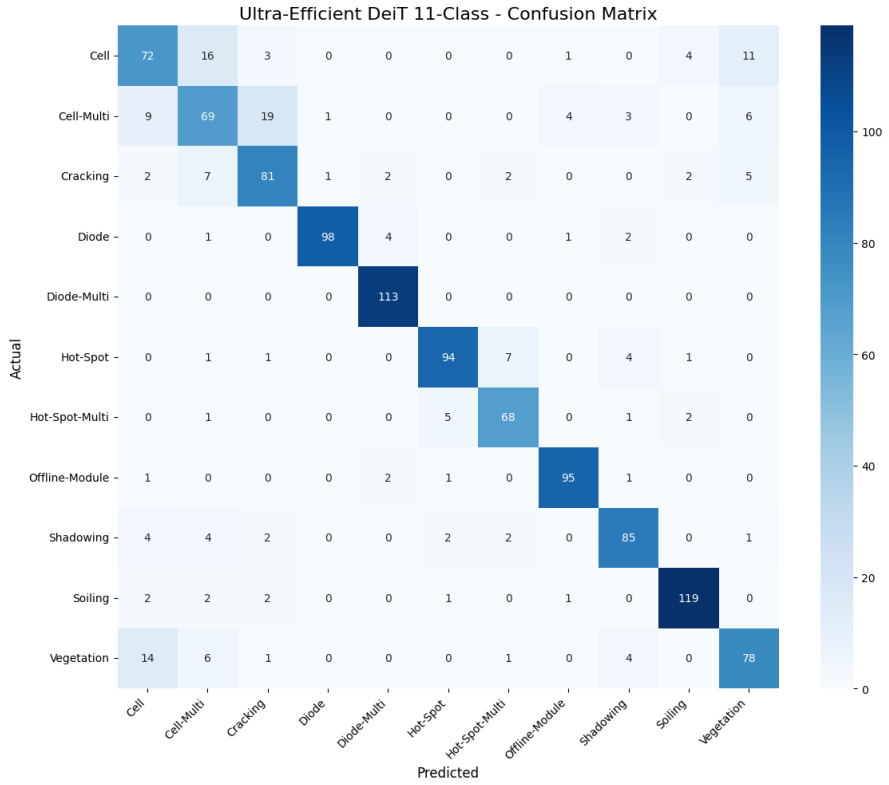



Figure 4.15: Confusion Matrix for Failed UltraEfficient DeiT-B16 (11-Class Classification)

The failed ultra-efficient configuration achieved a maximum validation accuracy of 85.37% and test accuracy of 84.16%, representing a significant 9.74% performance degradation compared to the baseline DeiT-B16 (95.11% validation accuracy). Despite the 44-minute training time (67% reduction from baseline 2:14:42), the model demonstrated severe convergence instability and inadequate feature learning capability.

The training dynamics in Figure 4.14 reveal erratic loss behavior with no clear convergence pattern, indicating that the frozen feature extraction layers failed to provide sufficient representational capacity for the complex thermal anomaly patterns. The confusion matrix in Figure 4.15 demonstrates widespread misclassification across multiple anomaly types, particularly affecting fine-grained distinctions between similar thermal signatures.

### 4.7.3 Key Findings

This failed experiment provides valuable empirical evidence regarding the computational requirements and architectural constraints of transformer models in specialized imaging domains. Key findings include:

- **Parameter Threshold**: 50% parameter freezing exceeds the acceptable efficiency threshold for thermal anomaly detection, suggesting that transformer architectures require substantial trainable capacity for domain-specific feature learning.

- **Feature Learning Limitations**: Aggressive freezing of early layers prevents adaptation to thermal imaging characteristics, highlighting the importance of end-to-end full fine-tuning for specialized visual domains.
- **Performance-Efficiency Trade-off**: The 67% training time reduction comes at the cost of 9.74% accuracy loss, demonstrating an unfavorable trade-off ratio for practical deployment scenarios.
- **Domain Specificity**: Unlike natural image domains (ImageNet), where transfer learning with frozen features is effective, thermal infrared imaging requires more extensive architectural adaptation due to distinct spectral and textural characteristics.

This analysis establishes critical boundaries for efficiency optimization in transformer-based thermal anomaly detection, informing future research directions toward more balanced parameter reduction strategies that maintain performance while achieving computational efficiency gains.

CHAPTER

# FIVE

# CONCLUSION AND FUTURE WORK

The applications of deep learning have advanced significantly in recent years, particularly in computer vision, where their ability to extract meaningful patterns from images has revolutionized various domains, including renewable energy. This research specifically explored the use of deep learning models to tackle the growing challenge of anomaly detection and classification in solar photovoltaic (PV) systems using infrared (IR) imagery. The findings confirm that transformer-based models and modern CNNs are not only capable of interpreting complex thermal patterns, but also highly effective in differentiating between fault types ranging from hot spots, diode failures, cell mismatches, to soiling. Leveraging the rich publicly available RaptorMaps dataset and pretrained architectures like ViT-B32, DeiT-B16, and ConvNeXt-Tiny, the study demonstrates that these models can deliver high accuracy, robustness, and efficiency. The research thus establishes a strong case for deploying deep learning-driven inspection tools in large-scale solar plants, where manual analysis is inefficient and error-prone. With the accelerating shift toward smart energy systems, this work supports the argument that data-driven, AI-powered inspection can play a vital role in enhancing the reliability and sustainability of PV infrastructure. Stakeholders and developers working on autonomous PV monitoring, particularly within green technology initiatives, can be confident in these models' ability to contribute to cleaner and more reliable solar power generation.

**Research Questions and Findings**

- **RQ 1: How do various data preprocessing techniques, particularly image processing and data augmentation, influence the performance of deep learning models for solar PV anomaly detection?**

  Data preprocessing and augmentation significantly enhance the performance and generalization of deep learning models in solar PV anomaly detection. In this study, resizing all images to a fixed size of 160×160 ensured input consistency, while normalization stabilized training. Augmentation techniques, including horizontal/vertical flipping and brightness variation, artificially expanded the dataset and introduced variability representative of real-world conditions, such as changes in viewing angle or illumination.

These techniques led to measurable improvements in validation accuracy and F1-scores, particularly for DeiT-B16 and ConvNeXt-Tiny, by reducing overfitting and improving robustness in multi-class classification. Overall, the use of targeted augmentation proved essential for training models that are resilient to visual noise and capable of detecting subtle thermal anomalies in PV systems.

- **RQ 2: Which deep learning (ViT and CNN variants) architectures demonstrate superior performance metrics (accuracy, precision, recall, F1-score) for detecting and classifying specific types of anomalies in solar PV systems?**

  Among the evaluated architectures (ViT-B32, DeiT-B16, and ConvNeXt-Tiny), the performance ranking varies by classification complexity and evaluation criteria:

  For **12-class classification** (most comprehensive), DeiT-B16 consistently demonstrated superior performance, achieving the highest overall accuracy (94.92%) and F1-scores (94.87% macro and weighted), indicating strong generalization and balanced performance across all anomaly types, including the "No-Anomaly" class.

  For **11-class classification** (anomaly types only), DeiT-B16 maintained its performance leadership with 94.37% accuracy and 94.35% F1-scores, while ViT-B32 and ConvNeXt-Tiny achieved identical accuracy (94.16%) and F1-scores (94.11%).

  For **2-class binary classification**, ConvNeXt-Tiny achieved the highest performance (99.05% accuracy), followed by DeiT-B16 (98.80%) and ViT-B32 (97.58%).

  **Computational efficiency analysis** reveals distinct trade-offs: ViT-B32 offers the fastest training time ($\sim$1.7 min/epoch), while ConvNeXt-Tiny provides optimal parameter efficiency ($\sim$28.2M parameters, 67% reduction compared to transformers) with competitive training time (2.2 min/epoch). DeiT-B16 requires the longest training time (3.1 min/epoch) but delivers superior classification performance.

  Overall, **DeiT-B16** proved most effective for accurate multi-class anomaly classification, while **ConvNeXt-Tiny** offers the best performance-efficiency trade-off for resource-constrained deployments, and **ViT-B32** provides optimal training speed when computational time is the primary constraint.

  A key architectural advantage of DeiT-B16 lies in its finer patch resolution: it uses a patch size of 16×16, compared to the coarser 32×32 patch size used in ViT-B32. For a fixed 160×160 input image, this translates to 100 tokens for DeiT-B16 versus only 25 for ViT-B32, enabling DeiT-B16 to capture more granular and localized features. This finer spatial granularity is especially beneficial in IR PV anomaly detection, where many faults, such as diode-edge heating or minor hotspots, manifest in small, localized regions. As such, DeiT-B16's higher token resolution directly contributes to its superior ability to detect subtle and spatially distributed anomalies in thermal imagery.

- **RQ 3: What methodological frameworks and technical approaches best address the challenges of class imbalance, data scarcity in computer vision-based solar PV anomaly detection systems?**
  To address the challenges of class imbalance and data scarcity in the solar PV anomaly dataset, extensive data augmentation techniques were applied. These included random horizontal and vertical flips, rotations, and color jittering, which increased the diversity and quantity of training samples for underrepresented classes. Additionally, stratified data splitting was used to ensure that all classes were proportionally represented in the training, validation, and test sets. Together, these approaches helped the deep learning models learn more robust features and improved their performance on minority classes.

## 5.1 Limitations of the Study

While this study provides valuable insights into the application of deep learning and computer vision for solar PV anomaly detection, it is important to acknowledge its limitations:

- **Dataset Specificity:** The models were trained and evaluated exclusively on the *IR* dataset. Their performance may vary on images captured with different thermal cameras, under different environmental conditions, or from different types of solar panels not represented in the dataset.
- **Computational Resources:** The scope of hyperparameter tuning was constrained by the available computational resources. A more extensive search could potentially yield a model with even higher performance.

## 5.2 Future Work

Based on the results and limitations of this study, several avenues for future research are recommended:

- **Hybrid Architectures:** More hybrid CNN-Transformer models can be explored that combine the local feature extraction capabilities of CNNs with the global attention mechanisms of transformers to improve anomaly detection performance.
- **Environmental Adaptation:** Investigate model robustness under varying environmental conditions, such as different weather patterns, seasonal changes, and time-of-day effects.
- **Floating PV Systems:** Extend the research to address anomaly detection challenges specific to floating photovoltaic (FPV) systems, taking into account the unique thermal characteristics caused by proximity to water.

Another key direction can be the development of a practical, user-friendly application that integrates pre-trained open-source models for automatic anomaly

detection from different regional thermal images. This system can provide confidence scores, estimate energy losses, and offer actionable maintenance recommendations. Incorporating continuous learning capabilities would further enhance performance as more field data becomes available, bridging the gap between research and real-world deployment.

# REFERENCES


[1] International Energy Agency, "Renewables 2023 – analysis and forecast to 2028," IEA, 2023, Annual renewable electricity capacity additions reached nearly 507 GW in 2023, with solar PV accounting for 75 % and solar+wind comprising 96 % of new capacity. [Online]. Available: https://www.iea.org/reports/renewables-2023.

[2] International Energy Agency, "Renewables 2024 – analysis and forecast to 2030," IEA, 2024, Global renewable capacity expected to expand by 5500 GW between 2024–2030, with solar PV providing around 80 % of growth. [Online]. Available: https://www.iea.org/reports/renewables-2024.

[3] L. Bommes, T. Pickel, C. Buerhop-Lutz, J. Hauch, C. Brabec, and I. Peters, "Computer vision tool for detection, mapping, and fault classification of photovoltaics modules in aerial ir videos," *Progress in Photovoltaics: Research and Applications*, vol. 29, no. 12, pp. 1236–1251, 2021.

[4] T. Kerekes and D. Séra, "Short survey of architectures of photovoltaic arrays for solar power generation systems," *Energies*, vol. 14, no. 16, p. 4917, 2021. DOI: 10.3390/en14164917. [Online]. Available: https://www.mdpi.com/1996-1073/14/16/4917.

[5] ABB Limited, "Photovoltaic plants: Cutting edge technology. from sun to socket," ABB Limited, Technical Report, 2019, Accessed: 2025-07-13. [Online]. Available: https://search.abb.com/library/Download.aspx?DocumentID=9AKK107492A3277&LanguageCode=en&DocumentPartId&Action=Launch.

[6] M. A. Green, *Solar Cells: Operating Principles, Technology, and System Applications*. Prentice-Hall, 1982.

[7] F. I. for Solar Energy Systems ISE, *Photovoltaics report*, Available: https://www.ise.fraunhofer.de/en/publications/studies/photovoltaics-report.html, 2020.

[8] I. E. A. P. P. S. P. (IEA-PVPS), *Review on ir and el imaging for pv field applications*, Available: https://iea-pvps.org/key-topics/review-on-ir-and-el-imaging-for-pv-field-applications/, 2020.

[9] ESE Solar LTD. "Solar farms – the ultimate guide." Published 25 January 2024. Accessed 13 July 2025. [Online]. Available: https://esesolar.co.uk/solar-farms-the-ultimate-guide/.

[10] U. Ramzan and M. Jamil, "Comparative analysis of floating solar photovoltaic and land based photovoltaic plant," in *2022 IEEE Silchar Subsection Conference (SILCON)*, 2022, pp. 1–5. DOI: 10.1109/SILCON55242.2022.10028831.

[11] M. A. Koondhar, L. Albasha, I. Mahariq, B. B. Graba, and E. Touti, "Reviewing floating photovoltaic (fpv) technology for solar energy generation," *Energy Strategy Reviews*, vol. 54, p. 101 449, 2024. DOI: https://doi.org/10.1016/j.esr.2024.101449. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2211467X24001561.

[12] V. Karppinen, J. Peltonen, and M. Leppäniemi, "A review of the degradation of photovoltaic modules for lifetime estimation," *Energies*, vol. 14, no. 14, p. 4278, 2021. DOI: 10.3390/en14144278.

[13] M. Köntges, J. Lin, A. Virtuani, G. C. Eder, J. Zhu, G. Oreski, P. Hacke, J. S. Stein, L. Bruckman, P. Gebhardt, D. Barrit, M. Rasmussen, I. Martin, K. O. Davis, G. Cattaneo, B. Hoex, Z. Hameiri, and E. Özkalay, "Degradation and failure modes in new photovoltaic cell and module technologies," International Energy Agency Photovoltaic Power Systems Programme (IEA PVPS), IEA PVPS Task 13 Report IEA-PVPS T13-30:2025, Feb. 2025. [Online]. Available: https://iea-pvps.org/wp-content/uploads/2025/02/IEA-PVPS-T13-30-2025-REPORT-Degradation-and-Failure.pdf.

[14] F. Brooks. "Identifying issues on installed photovoltaic systems using thermal imagery." Accessed: 2025-07-12, Infrared Training Center. [Online]. Available: https://www.infraredtraining.com/en-US/home/resources/blog/identifying-issues-on-installed-photovoltaic-systems--using-thermal-imagery/.

[15] Y. Sun, S. Chen, L. Xie, R. Hong, and H. Shen, "Investigating the impact of shading effect on the characteristics of a large-scale grid-connected pv power plant in northwest china," *International Journal of Photoenergy*, vol. 2014, pp. 1–9, 2014. DOI: 10.1155/2014/763106.

[16] M. Millendorf, E. Obropta, and N. Vadhavkar, *Infrared solar modules dataset for anomaly detection*, https://github.com/RaptorMaps/InfraredSolarModules, Accessed: 2025-07-11, 2020.

[17] M. Köntges, J. Lin, A. Virtuani, G. C. Eder, J. Zhu, G. Oreski, P. Hacke, J. S. Stein, L. Bruckman, P. Gebhardt, D. Barrit, M. Rasmussen, I. Martin, K. O. Davis, G. Cattaneo, B. Hoex, Z. Hameiri, and E. Özkalay, "Degradation and failure modes in new photovoltaic cell and module technologies," International Energy Agency Photovoltaic Power Systems Programme (IEA-PVPS), Task 13, Technical Report T13-30:2025, Feb. 2025. DOI: 10.69766/ATBD2730. [Online]. Available: https://iea-pvps.org/key-topics/degradation-failure-modes-new-cell-module-technologies/.

[18] M. Dhimish and Y. Hu, "Rapid testing on the effect of cracks on solar cells output power performance and thermal operation," *Frontiers in Energy Research*, vol. 10, p. 911 945, 2022. DOI: 10.3389/fenrg.2022.911945.

[19] M. Dhimish and G. Badran, "Investigating defects and annual degradation in uk solar pv installations through thermographic and electroluminescent surveys," *npj Materials Degradation*, vol. 7, p. 14, 2023. DOI: 10.1038/s41529-023-00331-y. [Online]. Available: https://doi.org/10.1038/s41529-023-00331-y.

[20] M. Dhimish and A. M. Tyrrell, "Photovoltaic bypass diode fault detection using artificial neural networks," *IEEE Transactions on Instrumentation and Measurement*, 2023, Accepted Author Manuscript. [Online]. Available: https://eprints.whiterose.ac.uk/id/eprint/196381/1/Accepted_Author_Manuscript.pdf.

[21] IEA Photovoltaic Power Systems Programme (IEA-PVPS), *Soiling losses – impact on the performance of photovoltaic power plants*, urlhttps://iea-pvps.org/key-topics/soiling-losses-impact-on-the-performance-of-photovoltaic-power-plants/, Accessed: 2025-07-16, 2023.

[22] Z. Xu, Y. Li, Y. Qin, and E. Bach, "A global assessment of the effects of solar farms on albedo, vegetation, and land surface temperature using remote sensing," *Solar Energy*, vol. 268, p. 112 198, 2024. DOI: https://doi.org/10.1016/j.solener.2023.112198. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0038092X23008320.

[23] Neo Messtechnik. "Photovoltaic inspection methods." Accessed: 2025-07-13. [Online]. Available: https://www.neo-messtechnik.com/en/methods-for-photovoltaic-inspection.

[24] "From indoor to daylight electroluminescence imaging for pv module diagnostics: A comprehensive review of techniques, challenges, and ai-driven advancements," *Micromachines*, vol. 16, no. 4, p. 437, 2025. DOI: 10.3390/mi16040437. [Online]. Available: https://www.mdpi.com/2072-666X/16/4/437.

[25] J. Vickerman. "How is an iv curve used to maximize solar output?" Reviewed by Enio Gjoni. Published by RatedPower. Updated 14 March 2024. Accessed 13 July 2025. [Online]. Available: https://ratedpower.com/glossary/iv-curve/.

[26] P. B. Quater, F. Grimaccia, S. Leva, M. Mussetta, and M. Aghaei, "Light unmanned aerial vehicles (uavs) for cooperative inspection of pv plants," *IEEE Journal of Photovoltaics*, vol. 4, no. 4, pp. 1107–1113, 2014. DOI: 10.1109/JPHOTOV.2014.2323714.

[27] M. Planck, *The Theory of Heat Radiation*, 2nd. Dover Publications, 1991, Originally published in 1914.

[28] J. A. Duffie and W. A. Beckman, "Solar engineering of thermal processes," *John Wiley & Sons*, 2013.

[29] M. García, S. Vilanova, J. A. Martínez, and F. Molés, "Infrared thermography for fault detection and diagnosis in photovoltaic systems: A review," *Renewable and Sustainable Energy Reviews*, vol. 135, p. 110 120, 2021. DOI: 10.1016/j.rser.2020.110120.

[30] Raptor Maps, Inc., *Calculating impact of anomalies & power factors*, Online technical documentation, Empirical power factor method for estimating DC power loss, 2025.

[31] IBM. "What is machine learning (ml)?" Accessed: 2025-07-15. [Online]. Available: https://www.ibm.com/think/topics/machine-learning.

[32] UC Berkeley School of Information. "What is machine learning (ml)?" Accessed: 2025-07-15. [Online]. Available: https://ischoolonline.berkeley.edu/blog/what-is-machine-learning/.

[33] GeeksforGeeks. "Supervised vs unsupervised vs reinforcement learning." Last updated: 27 May, 2025. Accessed: 2025-07-15. [Online]. Available: https://www.geeksforgeeks.org/machine-learning/supervised-vs-reinforcement-vs-unsupervised/.

[34] L. Bommes, "Computer vision pipeline for the automated inspection of photovoltaic plants," Ph.D. dissertation, Friedrich-Alexander-Universität Erlangen-Nürnberg, 2021.

[35] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2021. [Online]. Available: https://arxiv.org/abs/2010.11929.

[36] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems*, 2017.

[37] H. Touvron, M. Cord, M. Douze, F. Massa, A. Sablayrolles, and H. Jegou, "Training data-efficient image transformers & distillation through attention," *arXiv preprint arXiv:2012.12877*, 2020.

[38] Z. Liu, H. Mao, C.-Y. Wu, C. Feichtenhofer, T. Darrell, and S. Xie, "A convnet for the 2020s," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 11 976–11 986. [Online]. Available: https://openaccess.thecvf.com/content/CVPR2022/html/Liu_A_ConvNet_for_the_2020s_CVPR_2022_paper.html.

[39] Hugging Face, *Convnext — hugging face transformers documentation*, https://huggingface.co/docs/transformers/en/model_doc/convnext, Accessed: 2025-07-16, 2024.

[40] Z. Liu, H. Mao, C.-Y. Wu, C. Feichtenhofer, T. Darrell, and S. Xie, "A convnet for the 2020s," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 11 976–11 986.

[41] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," *arXiv preprint arXiv:1704.04861*, 2017.

[42] KUNGFU.AI. "Convnext: A transformer-inspired cnn architecture." Accessed: 2025-07-13. [Online]. Available: https://www.kungfu.ai/blog-post/convnext-a-transformer-inspired-cnn-architecture.

[43] D. Hendrycks and K. Gimpel, "Gaussian error linear units (gelus)," *arXiv preprint arXiv:1606.08415*, 2016.

[44] Y. Zhang, A. Xu, D. Lan, X. Zhang, J. Yin, and H. H. Goh, "Convnext-based anchor-free object detection model for infrared image of power equipment," *Energy Reports*, vol. 9, pp. 1121–1132, 2023, 2022 International Conference on Frontiers of Energy and Environment Engineering, CFEEE, 16–18 December 2022, Beihai, China. DOI: 10.1016/j.egyr.2023.04.145.

[45] V. Sinap and A. Kumtepe, "Cnn-based automatic detection of photovoltaic solar module anomalies in infrared images: A comparative study," *Neural Computing and Applications*, vol. 36, no. 28, pp. 17 715–17 736, 2024. DOI: 10.1007/s00521-024-10322-y.

[46] R. F. Pamungkas, I. B. K. Y. Utama, and Y. M. Jang, "A novel approach for efficient solar panel fault classification using coupled udensenet," *Sensors*, vol. 23, no. 10, p. 4918, 2023. DOI: 10.3390/s23104918. [Online]. Available: https://www.mdpi.com/1424-8220/23/10/4918.

[47] R. H. F. Alves, G. A. de Deus Júnior, E. G. Marra, and R. P. Lemos, "Automatic fault classification in photovoltaic modules using convolutional neural networks," *Renewable Energy*, vol. 179, pp. 502–516, 2021. DOI: 10.1016/j.renene.2021.07.070.

[48] M. Millendorf, E. Obropta, and N. Vadhavkar, "Infrared solar module dataset for anomaly detection," in *Proc. Int. Conf. Learn. Represent*, vol. 3, 2020.

[49] R. Maps, *Infrared solar modules dataset*, https://github.com/RaptorMaps/InfraredSolarModules, Accessed: 2025-07-09, 2020.

[50] S. W. Ko, Y. C. Ju, H. M. Hwang, J. H. So, Y.-S. Jung, H.-J. Song, H.-e. Song, S.-H. Kim, and G. H. Kang, "Electric and thermal characteristics of photovoltaic modules under partial shading and with a damaged bypass diode," *Energy*, vol. 128, pp. 232–243, 2017. DOI: https://doi.org/10.1016/j.energy.2017.04.030. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0360544217305959.

[51] V. Anand, O. Priyan, and B. Pesala, "Effect of shading losses on the performance of solar module system using matlab simulation," in *2014 IEEE 2nd International Conference on Electrical Energy Systems (ICEES)*, 2014, pp. 61–64. DOI: 10.1109/ICEES.2014.6924142.

[52] C. Deline, "Partially shaded operation of multi-string photovoltaic systems," in *2010 35th IEEE Photovoltaic Specialists Conference*, 2010, pp. 000 394–000 399. DOI: 10.1109/PVSC.2010.5616821.

[53] M. R. Maghami, H. Hizam, C. Gomes, M. A. Radzi, M. I. Rezadad, and S. Hajighorbani, "Power loss due to soiling on solar panel: A review," *Renewable and Sustainable Energy Reviews*, vol. 59, pp. 1307–1316, Jun. 2016. DOI: 10.1016/J.RSER.2016.01.044. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1364032116000745.

[54] A. Zheng and A. Casari, "Machine learning-based data preprocessing techniques for enhancing classification performance," in *Proceedings of the 2018 IEEE International Conference on Big Data (Big Data)*, IEEE, 2018, pp. 3739–3748.

[55] E. Software. "The power of image processing: Techniques, applications, and future trends." Accessed: 2025-07-13. [Online]. Available: https://medium.com/@eastgate/the-power-of-image-processing-techniques-applications-and-future-trends-9a3f455e2554.

[56] Z. Shi, Y. Chen, E. Gavves, P. Mettes, and C. G. Snoek, "Unsharp mask guided filtering," *ArXiv*, 2021. DOI: 10.1109/TIP.2021.3106812.

[57] S. C. Lin, C. Y. Wong, G. Jiang, M. A. Rahman, T. R. Ren, N. Kwok, H. Shi, Y. H. Yu, and T. Wu, "Intensity and edge based adaptive unsharp masking filter for color image enhancement," *Optik*, vol. 127, pp. 407–414, 1 Jan. 2016. DOI: 10.1016/J.IJLEO.2015.08.046. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0030402615008074.

[58] E. A. Ramadan, N. M. Moawad, B. A. Abouzalm, A. A. Sakr, W. F. Abouzaid, and G. M. El-Banby, "An innovative transformer neural network for fault detection and classification for photovoltaic modules," *Energy Conversion and Management*, vol. 314, p. 118 718, Aug. 2024. DOI: 10.1016/J.ENCONMAN.2024.118718. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0196890424006599.

[59] L. Taylor and G. Nitschke, "Improving deep learning with generic data augmentation," in *2018 IEEE Symposium Series on Computational Intelligence (SSCI)*, 2018, pp. 1542–1547. DOI: 10.1109/SSCI.2018.8628742.

[60] D. Madhugiri. "Improving deep learning models with data augmentation." ShortHills Tech, Medium. [Online]. Available: https://medium.com/@ShortHills_Tech/improving-deep-learning-models-with-data-augmentation-d4e3d0a9301b.

[61] GeeksforGeeks. "Z-score normalization: Definition and examples." Accessed: 2025-07-14. [Online]. Available: https://www.geeksforgeeks.org/data-analysis/z-score-normalization-definition-and-examples/.

[62] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, et al., *PyTorch: An imperative style, high-performance deep learning library*, https://pytorch.org/, Version 2.7.1, accessed July 2025, 2019.

[63] R. Wightman, *PyTorch Image Models (TIMM)*, https://github.com/huggingface/pytorch-image-models, Accessed July 2025, 2019.

[64] A. Clark and Contributors, *Pillow (pil fork)*, https://python-pillow.org/, Version latest, Accessed: 2025-07-15, 2015.

[65] C. R. Harris, K. J. Millman, S. J. van der Walt, R. Gommers, P. Virtanen, D. Cournapeau, E. Wieser, J. Taylor, S. Berg, N. J. Smith, R. Kern, M. Picus, S. Hoyer, M. Brett, A. Haldane, J. F. del Río, M. Wiebe, P. Peterson, P. Gérard-Marchant, K. Sheppard, T. Reddy, W. Weckesser, H. Abbasi, C. Gohlke, and T. E. Oliphant, *Array programming with numpy*, Accessed: 2025-07-15, 2020. DOI: 10.1038/s41586-020-2649-2. [Online]. Available: https://numpy.org/.

[66] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, *Scikit-learn: Machine learning in python*, Accessed: 2025-07-15, 2011. [Online]. Available: https://scikit-learn.org/.

[67] N. Yorav-Raphael, T. Gandor, C. da Costa-Luis, et al., *Tqdm: A fast, extensible progress bar for python and cli*, https://tqdm.github.io/, Version latest, Accessed: 2025-07-15, 2016.

[68] J. D. Hunter, *Matplotlib: A 2d graphics environment*, Accessed: 2025-07-15, 2007. DOI: 10.1109/MCSE.2007.55. [Online]. Available: https://matplotlib.org/.

[69] P. T. Inc., *Collaborative data science*, https://plotly.com/python/, Accessed: 2025-07-15, 2015.

[70] J. Ray. "Relu vs gelu." Accessed: 2025-07-13. [Online]. Available: https://medium.com/better-ml/relu-vs-gelu-d322422f5147.

[71] L. Guo, G. Andriopoulos, Z. Zhao, S. Ling, Z. Dong, and K. Ross, "Cross entropy versus label smoothing: A neural collapse perspective," *Transactions on Machine Learning Research*, 2025. arXiv: 2402.03979 [cs.LG].

[72] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 10, pp. 1345–1359, 2010. DOI: 10.1109/TKDE.2009.191.

[73] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural networks?" In *Advances in Neural Information Processing Systems*, 2014, pp. 3320–3328.

# APPENDICES

# A - GITHUB REPOSITORY

All code and LaTeX files used in this document are included in the Github.Check the repo GitHub linked below.

`https://github.com/ZeshanMubshir/Master-thesis---ICT-Simulation-and-Visualization`