



UNDERGRADUATE PROJECT REPORT

Project Title:	IABiLSTM-NET on Solar Cell Detection using Thermal Imaging
Surname:	Gao
First Name:	Rui
Student Number:	202118010407
Supervisor Name:	Happy Nkanta Monday
Module Code:	CHC 6096
Module Name:	Project
Date Submitted:	May 6, 2025

Chengdu University of Technology Oxford Brookes College

Chengdu University of Technology

BSc (Single Honors) Degree Project

Program Name: Projects

Module No.: CHC 6096

Surname: Gao

First Name: Rui

Project Title: IABiLSTM-NET on Solar Cell Detection using Thermal Imaging

Student No.: 202118010407

Supervisor: Happy Nkanta Monday

submitted: 6rd May 2025

*A report submitted as part of the requirements for the degree of BSc (Hons) in Computer
Science*

At

Chengdu University of Technology Oxford Brookes College

Declaration

Student Conduct Regulations:

Please ensure you are familiar with the regulations in relation to Academic Integrity. The University takes this issue very seriously and students have been expelled or had their degrees withheld for cheating in assessment. It is important that students having difficulties with their work should seek help from their tutors rather than be tempted to use unfair means to gain marks. Students should not risk losing their degree and undermining all the work they have done towards it. You are expected to have familiarized yourself with these regulations.

<https://www.brookes.ac.uk/regulations/current/appeals-complaints-and-conduct/c1-1/>

Guidance on the correct use of references can be found on www.brookes.ac.uk/services/library, and also in a handout in the library.

The full regulations may be accessed online at

<https://www.brookes.ac.uk/students/sirt/student-conduct/>

If you do not understand what any of these terms mean, you should ask your Project Supervisor to clarify them for you.

I declare that I have read and understood Regulations C1.1.4 of the Regulations governing Academic Misconduct, and that the work I submit is fully in accordance with them.

Signature 

Date 6 May 2025

REGULATIONS GOVERNING THE DEPOSIT AND USE OF OXFORD BROOKES
UNIVERSITY MODULAR PROGRAMME PROJECTS AND DISSERTATIONS

Copies of projects/dissertations, submitted in fulfillment of Modular Programme requirements and achieving marks of 60% or above, shall normally be kept by the Oxford Brookes University Library.

I agree that this dissertation may be available for reading and photocopying in accordance with the Regulations governing the use of the Oxford Brookes University Library.

Signature 

Date 6 May 2025

Acknowledgment

I would like to express my sincere gratitude to Dr. Happy Nkanta Monday, my dedicated supervisor, and Dr. Joojo Walker, my Module Leader, for their invaluable support and guidance throughout my project on the application of deep learning in solar cell detection. Dr. Happy Nkanta Monday provided insightful advice and rigorous supervision, which significantly enriched my research journey. Dr. Joojo Walker, in addition to delivering comprehensive course content, offered substantial assistance that greatly facilitated my comprehension and application of complex concepts. Their combined efforts were instrumental in the successful completion of my project. I am deeply thankful for their unwavering dedication and support.

Table of Contents

Declaration	i
Acknowledgment	iii
Table of Contents	iv
List of Figures	vii
List of Tables	x
Abstract	xi
Abbreviations	xii
Glossary	xiv
Chapter 1 Introduction	1
1.1 Background	1
1.1.1 Solar Cell Structure	1
1.1.2 Solar Cell Applications	2
1.1.3 Solar Cell Defects	3
1.1.4 Detection Methods Overview	4
1.1.5 Convolution Neural Network Overview	5
1.2 Aim	11
1.3 Objectives	11
1.4 Project Overview	12
1.4.1 Scope	12
1.4.2 Audience	12
Chapter 2 Background Review	14
Chapter 3 Methodology	17
3.1 Approach	17
3.1.1 Inception Module	17

3.1.2	Attention Block	18
3.1.3	BiLSTM Module	19
3.1.4	Proposed Model	20
3.1.5	Focal Loss Function	21
3.2	Dataset	22
3.2.1	Dataset Introduction	22
3.2.2	Data Separation	23
3.2.3	Data Resize	26
3.2.4	Data Augmentation	26
3.3	Technology	27
3.4	Data Testing and Evaluation Plan	27
3.4.1	Data Testing	27
3.4.2	Model Performance Evaluation Criteria	28
Chapter 4 Implementation and Results	30
4.1	Experiment Phases	30
4.2	Three different split ratios on the proposed model	31
4.2.1	Split 80:10:10 dataset on the proposed model	31
4.2.2	Split 70:20:10 dataset on the proposed model	34
4.2.3	Split 60:20:20 on the proposed model	36
4.3	Comparison of three split radio	40
4.3.1	Accuracy Comparison of three split radio	40
4.3.2	Loss Comparison of three split radio	40
4.3.3	AUC Comparison of three split radio	42
4.3.4	F1 Score Comparison of three split radio	42
4.3.5	Confusion Matrix Comparison of three split radio	44
4.3.6	Overall Comparison of three split radio	46
4.4	Comparison of Four Models	47

4.4.1	Accuracy comparison of four models	47
4.4.2	Loss comparison of four models	48
4.4.3	AUC comparison of four models	49
4.4.4	F1 Score comparison of four models	50
4.4.5	Confusion Matrix comparison of four models	51
4.4.6	Overall Comparison of four models	54
4.5	Model Visualization	55
4.5.1	Grad-CAM	55
4.6	Model Deployment	55
Chapter 5 Professional Issues	60
5.1	Project Management	60
5.1.1	Activities	60
5.1.2	Schedule	61
5.1.3	Project Data Management	62
5.1.4	Project Data Management	63
5.1.5	Project Deliverables	63
5.2	Risk Analysis	63
5.3	Professional Issues	65
Chapter 6 Conclusion	66
References	67
Appendices	70

List of Figures

Figure 1. The Basic Structure of a Solar Cell	2
Figure 2. Examples of Different Solar Cell Applications	3
Figure 3. An Example of CNN Architecture for Image Classification	6
Figure 4. The Primary Calculations Executed at Each Step of Convolutional Layer	7
Figure 5. Diagram of ReLU Function	8
Figure 6. Diagram of Sigmoid Function	9
Figure 7. Diagram of Tanh Function	9
Figure 8. An Example of Three Pooling Operations	10
Figure 9. Fully Connected Layer	11
Figure 10. The Inception Module of the Proposed Model	18
Figure 11. The Attention Block in the Proposed Model	19
Figure 12. The BiLSTM Module in the Proposed Model	20
Figure 13. The Structure of the Proposed Module	21
Figure 14. Examples of Solar Cell Thermal Images	22
Figure 15. The Structure of the Dataset Split 80:10:10 Ratio	23
Figure 16. The Class Distribution of Split 80:10:10 Ratio	24
Figure 17. The Structure of the Dataset Split 70:20:10 Ratio	24
Figure 18. The Class Distribution of Split 70:20:10 Ratio	25
Figure 19. The Structure of the Dataset Split 60:20:20 Ratio	25
Figure 20. The Class Distribution of Split 60:20:20 Ratio	26
Figure 21. The Augmentation of the Images	27
Figure 22. Accuracy Curve of Split 80:10:10	31

Figure 23. Loss Curve of Split 80:10:10	32
Figure 24. Precision and Recall Curve of Split 80:10:10	33
Figure 25. AUC Curve of Split 80:10:10	33
Figure 26. Accuracy Curve of Split 70:20:10	34
Figure 27. Loss Curve of Split 70:20:10	35
Figure 28. Precision and Recall Curve of Split 70:20:10	35
Figure 29. AUC Curve of Split 70:20:10	36
Figure 30. Accuracy Curve of Split 60:20:20	37
Figure 31. Loss of Split 60:20:20	38
Figure 32. Precision and Recall Curve of Split 60:20:20	38
Figure 33. AUC Curve of Split 60:20:20	39
Figure 34. Accuracy Comparison of three split radio	40
Figure 35. Loss Comparison of three split radio	41
Figure 36. AUC Comparison of three split radio	42
Figure 37. F1 Score Comparison of three split radio	43
Figure 38. Confusion Matrix of split 80:10:10	44
Figure 39. Confusion Matrix of split 70:20:10	45
Figure 40. Confusion Matrix of split 60:20:20	46
Figure 41. Accuracy comparison of four models	47
Figure 42. Loss comparison of four models	48
Figure 43. AUC comparison of four models	49
Figure 44. F1 Score comparison of four models	50
Figure 45. Confusion matrix of proposed model	51
Figure 46. Confusion matrix of Inception model	52

Figure 47. Confusion matrix of Inception + Attention model	52
Figure 48. Confusion matrix of CNN model	53
Figure 49. Grad-CAM image example	55
Figure 50. Homepage 1 of the Webpage	56
Figure 51. Homepage 2 of the Webpage	56
Figure 52. Functions Component of the Webpage	57
Figure 53. Detection Component of the Webpage	57
Figure 54. Result of Prediction	58
Figure 55. Dataset Component of the Website	58
Figure 56. Dataset Website of the Project	59
Figure 57. Gantt Chart of the Project	62

List of Tables

Table 1. Details of CNN Layers	6
Table 2. Summary of Existing Researches	16
Table 3. Summary of Relevant Technology involved in this project.....	27
Table 4. Overall Comparison of three split radio	46
Table 5. Overall Comparison of four models	54
Table 6. Activities of the project	61
Table 7. Project Data Management	62
Table 8. Risk Analysis	Error! Bookmark not defined.

Abstract

As global awareness of climate change grows, renewable energy sources like solar power have become increasingly important. Photovoltaic (PV) systems, which convert sunlight into electricity using solar cells, are widely used due to their sustainability, low maintenance costs, and ease of installation. However, PV panels often suffer from defects such as cracks, hot spots, and shading, which can significantly reduce energy conversion efficiency and, if left undetected, lead to system failure. To address this, the project proposes a deep learning classification model that combines Inception modules, Attention mechanisms, and Bi-directional Long Short-Term Memory (BiLSTM) to detect solar panel defects from thermal infrared images. The model is trained and evaluated on a public dataset containing 5,352 thermal images of solar panels, with labels indicating the presence or absence of thermal defects. Three data partitioning strategies (80:10:10, 70:20:10, and 60:20:20) are applied to assess model performance under different validation conditions. Evaluation metrics include Accuracy, Loss, Precision, Recall (Sensitivity), Specificity, F1-Score, and AUC-ROC. The proposed model achieved an accuracy of 91.56%, an AUC of 0.9677, and an F1-Score of 95.33%. Furthermore, a user-friendly graphical interface has been developed to allow users to upload thermal images and receive instant classification results. This system enhances defect detection efficiency, supports preventive maintenance, extends panel lifespan, and reduces long-term operational costs in solar energy systems.

Keywords: CNN, Solar Cell, Deep Learning, Thermal image, Inception, Attention Mechanism, BiLSTM, Explainable AI

Abbreviations

Acc	Accuracy
AUC	Area Under the ROC Curve
Attention	Attention Mechanism
BiLSTM	Bidirectional Long Short-Term Memory
BS	Batch Size
CNN	Convolutional Neural Network
FN	False Negative
FP	False Positive
F1	F1-Score
GAP	Global Average Pooling
GPU	Graphic Processing Unit
HF	Horizontal Flip
HS	Height Shift
IR	Infrared
Inception	Inception Module
L2	L2 Regularization
LR	Learning Rate
MRI	Magnetic Resonance Imaging
PV	Photovoltaic
Prec	Precision
ReLU	Rectified Linear Unit
Rec	Recall
ROC	Receiver Operating Characteristic

RR	Rotation Range
Spec	Specificity
SR	Shear Range
TN	True Negative
TP	True Positive
WS	Width Shift
XAI	Explainable Artificial Intelligence
XML	Extensible Markup Language
ZR	Zoom Range

Glossary

Convolutional Neural Network (CNN): A type of deep neural network commonly used in image processing and computer vision tasks. CNNs are designed to automatically and adaptively learn spatial hierarchies of features from input images by using convolutional layers, pooling layers, and fully connected layers.

Solar Cell: A semiconductor device that converts light energy directly into electrical energy through the photovoltaic effect. Solar cells are the fundamental building blocks of photovoltaic systems, which are used for solar power generation.

Deep Learning: A subset of machine learning that involves training artificial neural networks with many layers (hence “deep”). Deep learning is particularly effective for large-scale data and complex tasks, such as image classification, speech recognition, and natural language processing.

Thermal Image: An image that represents the infrared radiation emitted by objects, typically captured using thermal cameras. These images are used to detect temperature variations, which can be particularly useful in applications like identifying faulty components in photovoltaic systems.

Inception: A deep learning model architecture introduced by Google, known for its use of multiple filter sizes in parallel within the same layer. This helps the model capture both fine and coarse features from input data, improving performance for complex image classification tasks.

Attention Mechanism: A mechanism used in deep learning models, particularly in sequence processing tasks, that allows the model to focus on important parts of the input data while processing it. It assigns different levels of importance (or “attention”) to different parts of the input, which helps improve the model’s performance, especially for tasks like translation and image captioning.

Bidirectional Long Short-Term Memory (BiLSTM): A type of recurrent neural network (RNN) that processes data in both forward and backward directions. BiLSTM improves performance in tasks involving sequential data by learning from past and future context simultaneously, which is useful for time-series analysis and sequence labeling tasks.

Explainable AI (XAI): An area of artificial intelligence focused on developing models and techniques that are interpretable and understandable by humans. XAI aims to make

the decision-making processes of AI systems transparent, ensuring that users can trust and understand how conclusions are drawn, which is especially important in safety-critical applications.

Chapter 1 Introduction

1.1 Background

People now focus more on renewable and clean resource due to the impact of global warming and fossil fuels. Additionally, the increasing demand for electricity due to factors such as a growing population necessitates effort to move away from conventional methods of energy production [1]. There is a kind of technology called solar photovoltaic cells which can generate electricity by absorbing sunlight and converting light energy into an electric current. Solar energy is mainly harvested by photovoltaic plants which have numerous advantages, including extended lifespan, sustainability, low-noise performance, and cleanliness [2]. Solar cells can be installed in series or parallel to form photovoltaic (PV) panels. PV panels are not only environmentally friendly but also relatively easy to install and deploy [3].

1.1.1 Solar Cell Structure

A solar cell is a device that harnesses the photovoltaic effect to directly transform light energy into electrical energy [4]. Here is the descriptions of the solar cell structure and figure 1 shows the basic structure of a solar cell.

1. **Top Contact Layer**, as known as Transparent Conductive Layer, is the topmost layer of the solar cell which is typically made of a transparent conductive material like indium tin oxide (ITO) or doped tin oxide. The conductive material allows the sunlight to pass through to the active layers beneath.
2. **Emitter Layer** is a thin layer located under the top contact layer. This layer facilitates the movement of electrons generated after absorbing sunlight.
3. **Base Layer** is a thicker layer of lightly doped p-type silicon located under the emitter layer. This layer contributes to the formation of an electric field within the cell by supplying positively charged "holes" that attract electrons.
4. **Back Surface Field (BSF) Layer** is added to the rear surface of the solar cell to improve charge carrier (electrons and holes) collection and minimize recombination losses.
5. **Back Contact Layer** is the bottom layer. This layer is often made of a conductive material like aluminum or silver and its main role is to gather the generated electrons and establish an external pathway for the electrical current to exit the cell.

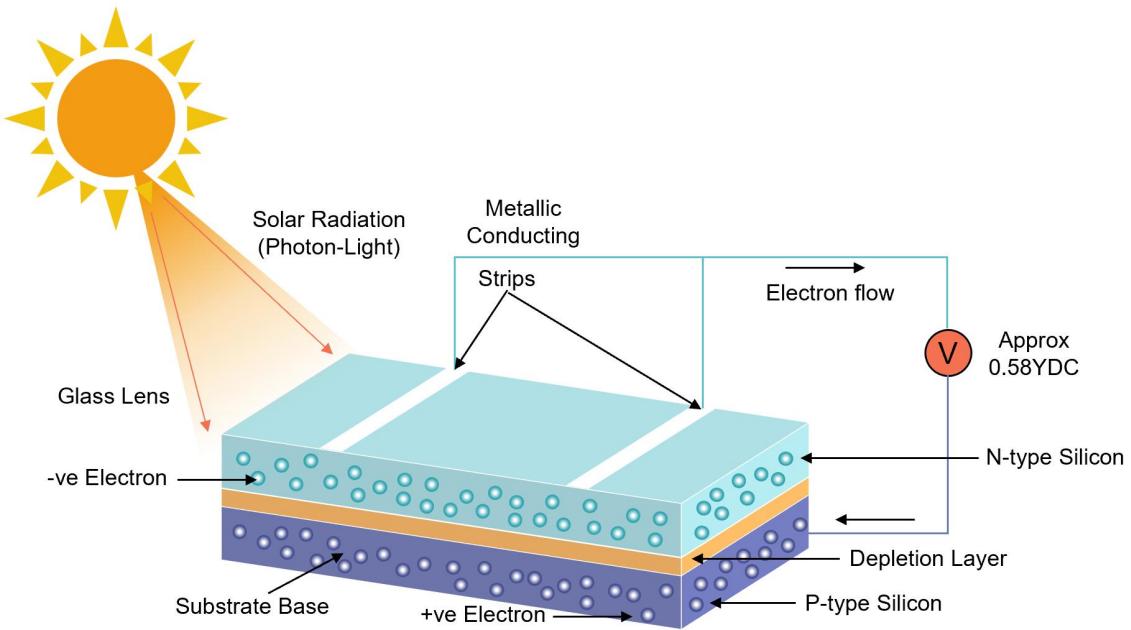


Figure 1. The Basic Structure of a Solar Cell

1.1.2 Solar Cell Applications

Solar panels play a crucial role in the global energy system, not only promoting the development of renewable energy but also being widely applied in various fields. In residential applications, household devices can directly use electricity generated by solar power. For example, solar water heaters can provide hot water for households, and installing photovoltaic cells on the roof can store electricity and reduce energy costs. Additionally, in commercial applications, many businesses and industrial parks have installed large-scale solar power stations to reduce their carbon footprint and operational costs. Besides, solar energy is also widely used in remote areas. For example, schools, community centers, and clinics in remote regions can install solar panels to generate and utilize electricity [5]. Figure 2 shows some examples of different solar cell applications.

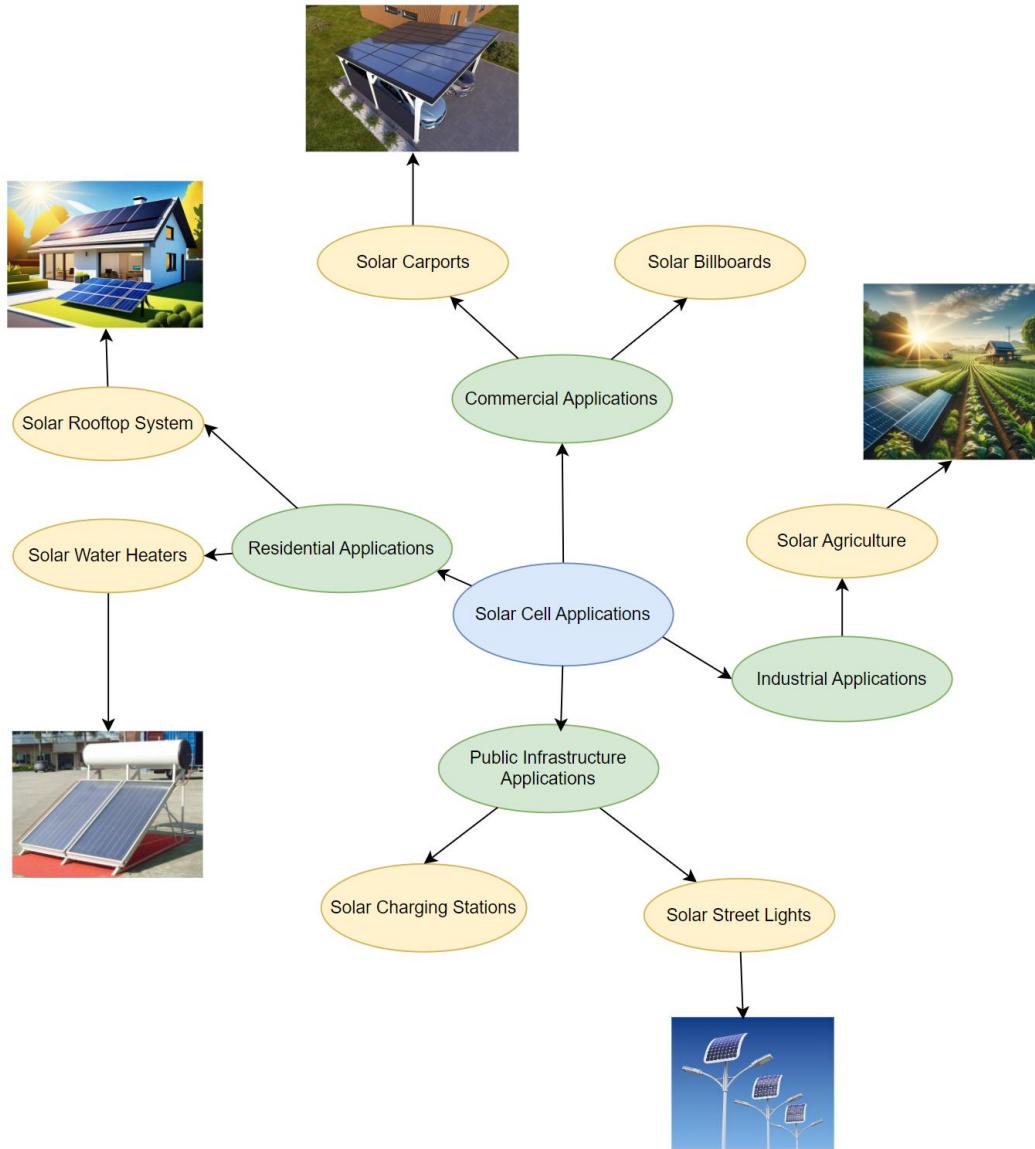


Figure 2. Examples of Different Solar Cell Applications

1.1.3 Solar Cell Defects

PV panels have yet to achieve a perfect conversion rate of the incident solar energy [3]. In addition, because some solar cell defects will occur during manufacturing, installation, throughout their operational lifetime and so on. These defects can negatively impact the efficiency and performance of the solar panels. Here are the common types of solar cell defects: for example, cracks usually appear as visible lines or fractures on the surface of solar panels, which can obstruct the internal current flow and consequently reduce efficiency. In addition, hot spots are localized overheating phenomena on solar panels. Prolonged presence of hot spots may cause permanent damage to the panels and even

pose a fire hazard. Additionally, shading refers to objects such as branches, dust, dirt, or snow covering the surface of a solar panel, leading to a decrease in the output power of the shaded cells. In severe cases, it may result in the formation of hot spots. Not only these listed defects, there are many defects which can cause a variety of issues ranging from minor decreases in efficiency to complete failure of solar panels. The best way to ensure the long-term stable and efficient operation of a solar energy system is to identify and address these issues in a timely manner.

1.1.4 Detection Methods Overview

With the rapid development of the solar photovoltaic (PV) industry, ensuring the efficient operation of solar panels is crucial for improving energy conversion efficiency and reducing maintenance costs. However, solar cell defect detection is challenging due to issues such as interference from complex backgrounds and the diversity in shape of the same class of defects in solar cells [6]. Various methods have been used for defect detection in solar panels, including some traditional detection techniques and the increasingly adopted thermal imaging technology in recent years.

The traditional detection methods mainly include the manual inspection and visible light imaging detection. Manual inspection involves visually examining solar panels for visible damage or using handheld devices for localized testing to detect abnormalities in panel voltage, current, or surface temperature. Therefore, when dealing with a large number of solar panels, manual inspection is highly inefficient and time-consuming. Moreover, the results heavily depend on the experience of the operator, which may make mistakes in judgment. Additionally, it is difficult to detect some hidden defects. Visible light imaging detection mainly relies on regular cameras or drones to capture the surface of solar panels, and defects are identified through image analysis. Therefore, it is only suitable for surface defects and is affected by weather and lighting conditions.

To address the shortcomings of traditional detection methods, thermal imaging technology has become an important tool for solar panel inspection in recent years. Thermal imaging is a method that measures an object's temperature without physical contact or causing any damage [7]. Thermal imaging can accurately detect hot spots, microcracks, identify electrical faults, and detect shading effects when inspecting solar panels. It also enables non-contact, long-distance inspection and can adapt to complex environments. In recent years, with the advancements in deep learning technology, thermal imaging detection can be combined with machine learning algorithms to achieve

automated fault recognition. By combining advanced machine learning techniques with thermal imaging, these anomalies can be easily detected and inferred in a contactless and non-invasive manner. Additionally, the ability to perform efficient and scalable operations on large datasets has further driven the widespread use of thermal imaging with machine learning techniques [7]. This can further improve detection accuracy, providing more intelligent maintenance solutions for the solar photovoltaic industry, reducing energy loss, and enhancing overall power generation efficiency.

1.1.5 Convolution Neural Network Overview

Since Convolutional Neural Network (CNN) has the advantage of automatically extracting features without human intervention, it has become the most widely used model [8]. In addition, CNN has the ability to learn highly abstract features and can efficiently recognize objects, which further highlights its superiority [9]. The weight sharing feature of CNN can reduce training time and the number of training parameters, avoid model overfitting, and also improve generalization ability [10]. Convolutional Neural Networks (CNNs) are widely used in image recognition and classification tasks, particularly suited for processing image data with spatial structural features. Moreover, CNNs have demonstrated unique advantages in thermal image defect detection. This is because thermal images primarily reflect the thermal characteristics of a target through pixel temperature distribution, where defect areas typically exhibit temperature patterns that are significantly different from surrounding regions. These local spatial differences are exactly the kind of features that CNNs are adept at capturing. Additionally, CNNs can automatically extract multi-level image features through convolution operations, thereby enabling more effective learning of discriminative features from temperature distributions.

A Convolutional Neural Network (CNN) typically begins with an input layer, followed by key components such as convolutional layers, pooling layers, and fully connected layers. This architecture enables the step-by-step processing of raw image data, allowing for the effective extraction of important features and supporting accurate classification or prediction decisions [11]. Table 1 shows some details of CNN layers, and figure 3 shows an example of CNN architecture for image classification.

Layer	Detail
Input Layer	<ul style="list-style-type: none">● receive the image data.

Convolutional Layer	<ul style="list-style-type: none"> It contains a set of convolutional kernels (filters), which can do the convolution operations [10]. The input image represents as N-dimensional metrics.
Activation Layer	<ul style="list-style-type: none"> It aims to convert the input received by the neuron into an output [8].
Pooling Layer	<ul style="list-style-type: none"> It can help the network process the data more efficiently while preserving the most important features [10].
Fully Connected Layer	<ul style="list-style-type: none"> It combines and maps the previously extracted features to the output space.
Output Layer	<ul style="list-style-type: none"> The final layer makes different outputs which depend on the task of the model.

Table 1. Details of CNN Layers

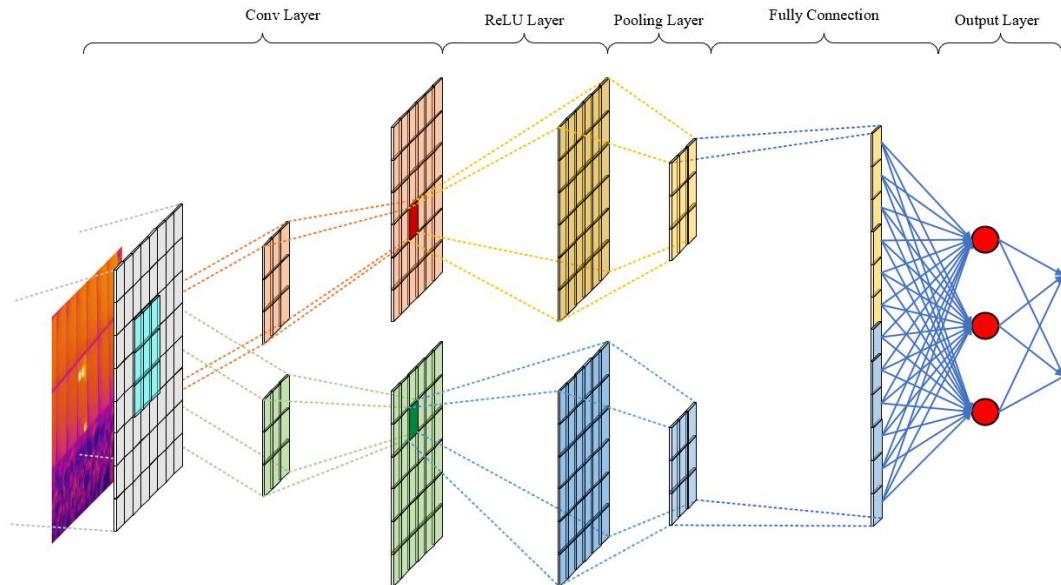


Figure 3. An Example of CNN Architecture for Image Classification

1.1.5.1 Convolution Layer

The convolutional layer is the core of the CNN, used to extract local features from the image. Each convolutional layer performs a sliding operation on the image using multiple filters, extracting different levels of spatial features such as edges, textures, and

corners. Each convolution operation generates a feature map. Figure 4 shows the primary calculations executed at each step of convolutional layer.

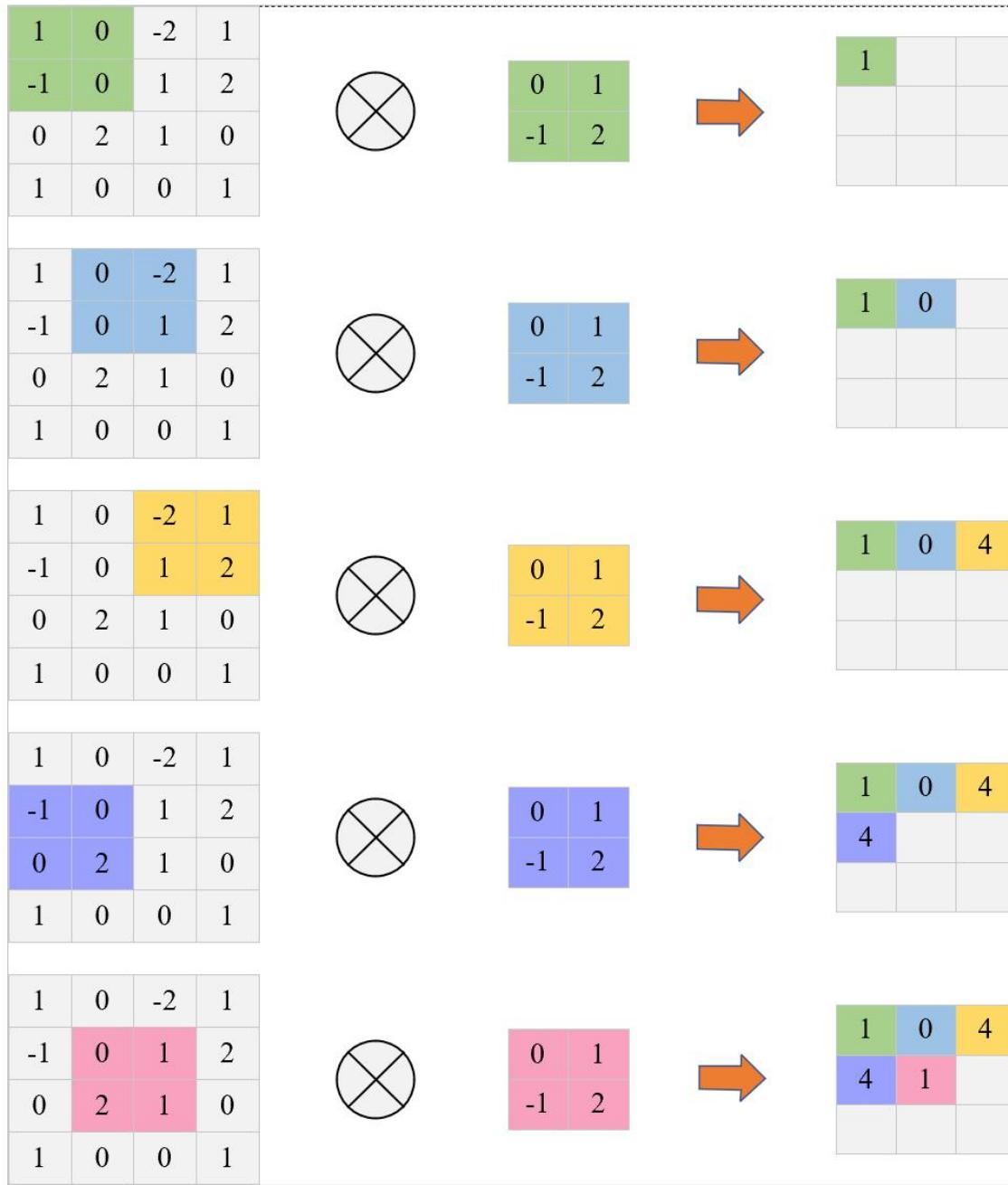


Figure 4. The Primary Calculations Executed at Each Step of Convolutional Layer

1.1.5.2 Activation Layer

CNN typically adds a nonlinear activation function after each convolution. The activation function must also be differentiable, which is a crucial feature as it allows the use of

error back-propagation to train the network [8]. The following types of activation functions are most commonly used in CNN.

- Rectified Linear Unit (ReLU): computation is simple and does not require complex mathematical operations. Additionally, during forward propagation, ReLU provides faster convergence and effectively alleviates the vanishing gradient problem as expressed in equation 1.

$$f(x) = \max(0, x)$$

(1)

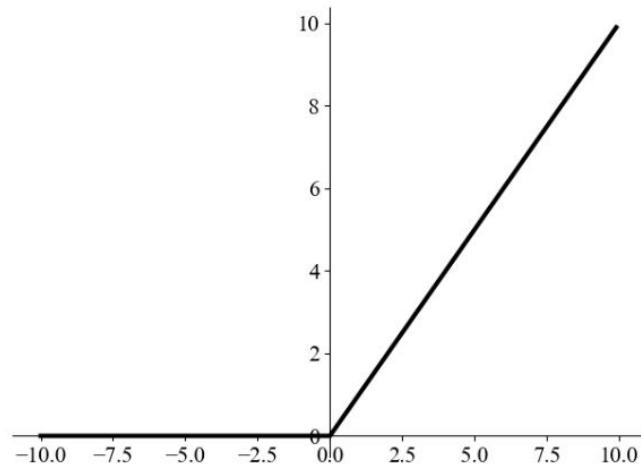


Figure 5. Diagram of ReLU Function

- Sigmoid: the output values are between (0, 1), making it very suitable for the output layer of binary classification problems as expressed in equation 2.

$$\text{Sigmoid}(x) = \frac{1}{1+e^{-x}} \quad (2)$$

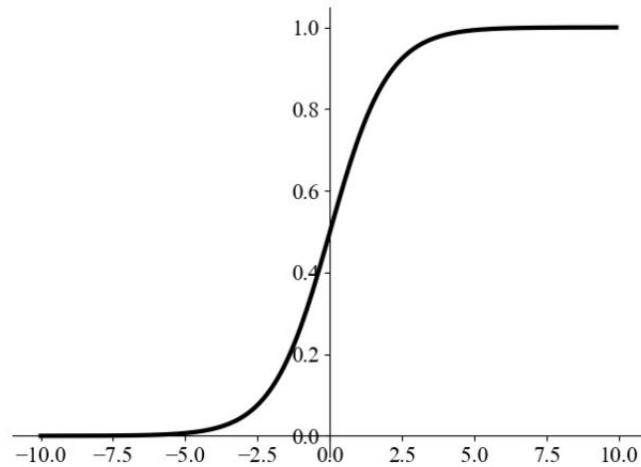


Figure 6. Diagram of Sigmoid Function

- Tanh: the output values are between (-1, 1), and overall, it performs better than Sigmoid as expressed in equation 3.

$$\text{Tanh}(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad (3)$$

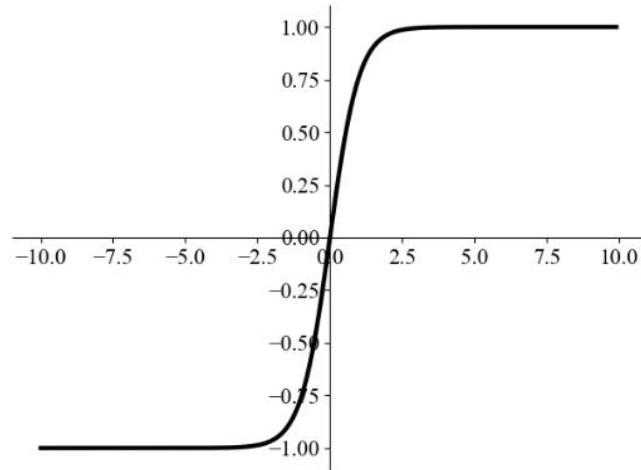


Figure 7. Diagram of Tanh Function

- Softmax: converts each output value into a probability distribution. It is used in the output layer for multi-class problems, converting the model's output values into probabilities and providing the predicted probability for each class as expressed in equation 4.

For an output vector $z = [z_1, z_2, \dots, z_n]$, the Softmax function for the i^{th} output is:

$$\text{Softmax}(Z_i) = \frac{e^{Z_i}}{\sum_{j=1}^n e^{Z_j}} \quad (4)$$

1.1.5.3 Pooling Layer

The pooling layer is used to downsample feature maps, reduce data dimensions, control overfitting, and retain key features. Common operations include Max Pooling, Average Pooling, and Global Average Pooling. It can reduce the size of large feature maps by generating smaller ones, while still preserving most of the essential or dominant features at each step of the pooling process [8]. Figure 8 explains these three pooling operations.

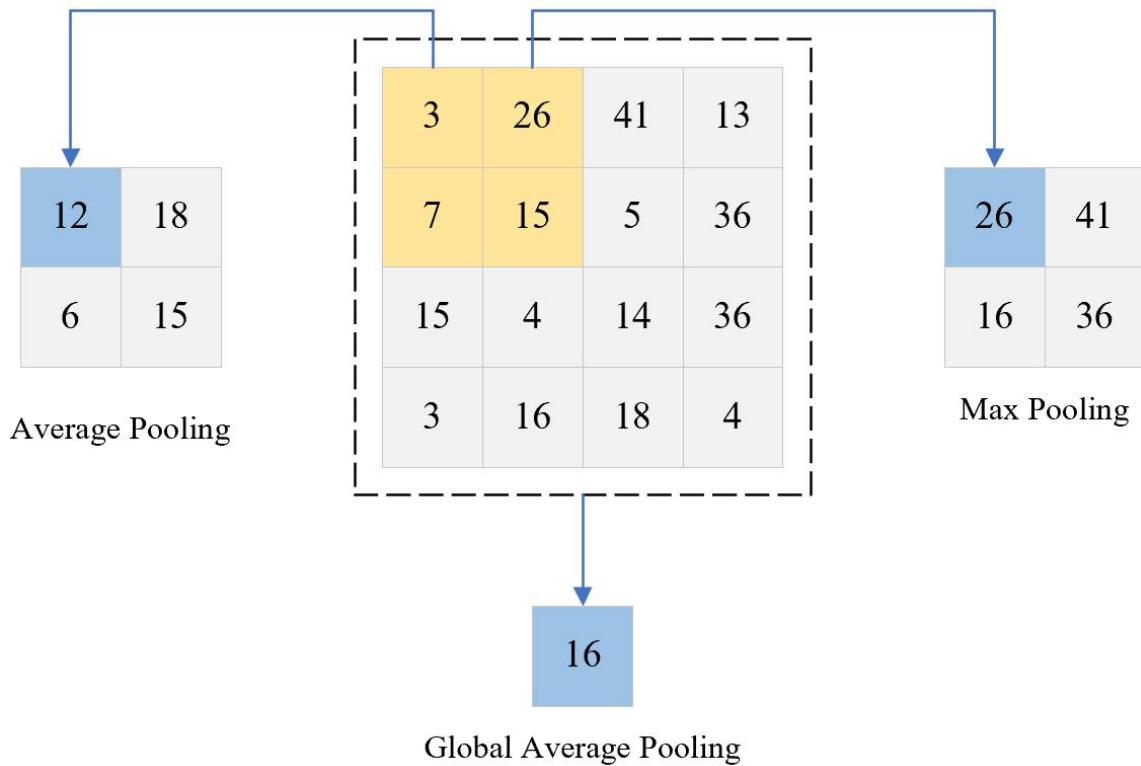


Figure 8. An Example of Three Pooling Operations

1.1.5.4 Fully Connected Layer

After several convolution and pooling layers, the final feature maps are flattened into a one-dimensional vector and fed into one or more fully connected layers for final feature integration and decision-making. The fully connected layers are similar to traditional feedforward neural networks and are a crucial component for classification tasks. The output of the FC layer represents the final CNN output, as illustrated in figure 9.

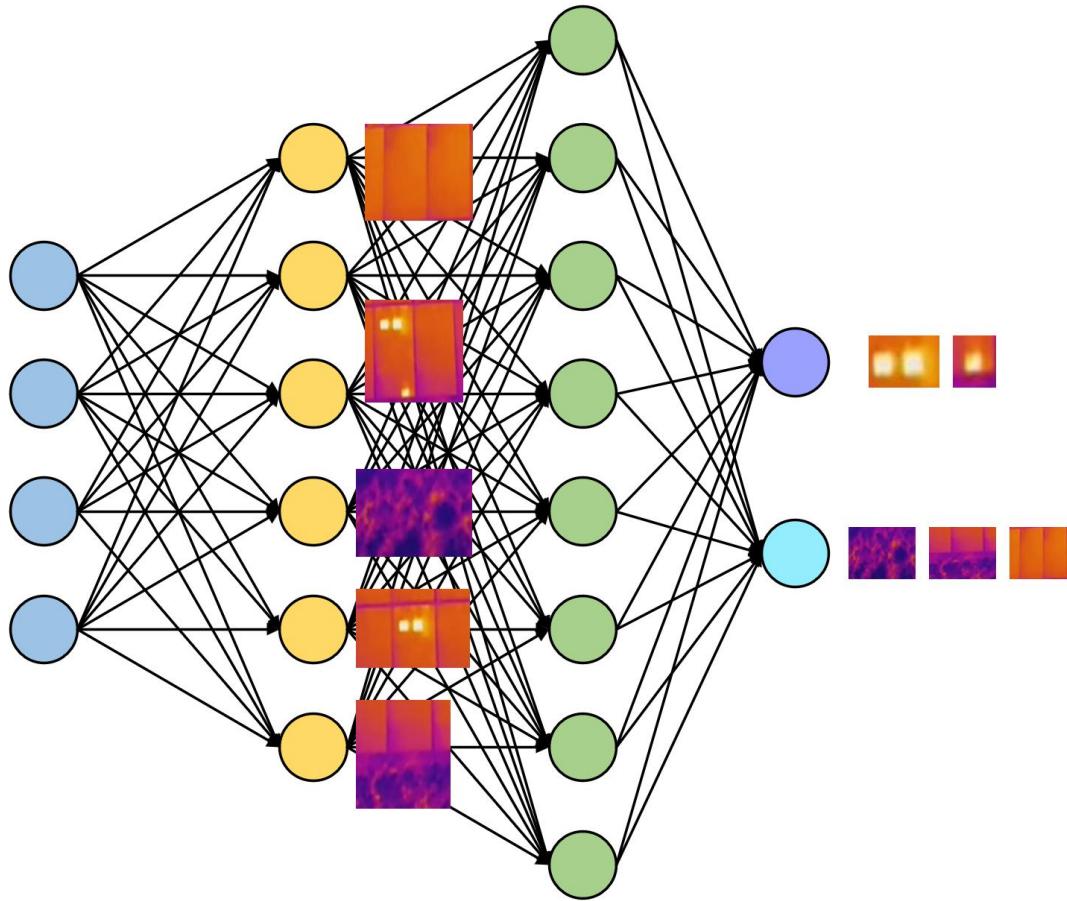


Figure 9. Fully Connected Layer

1.2 Aim

This project aims to develop a deep learning classification model that combines Inception, Attention, and BiLSTM to detect defects in solar panels using thermal infrared images. The project leverages thermal infrared images of solar panels for model training and evaluation, enabling early detection of solar panel degradation, improving lifespan and maintenance scheduling. It assists maintenance personnel in taking targeted measures to prevent further equipment damage. By using this deep learning model, the efficiency and accuracy of maintenance work can be significantly improved, effectively extending the lifespan of solar panels and reducing replacement costs.

1.3 Objectives

This project will use the solar cell thermal images from online sources, utilizing a public dataset. This dataset has 5,352 thermal images of solar cells and the labels are ‘thermal defect’ which means if the XML file includes ‘thermal defect’ the cell is defected. The

dataset will be divided into training, validation, and test sets using three different partition ratios. The first ratio is 80% for training, 10% for validation, and 10% for testing. The second ratio is 70% for training, 20% for validation, and 10% for testing. The third ratio is 60% for training, 20% for validation, and 20% for testing. It is necessary for each training, validation, and test set to contain both defect and no-defect types. The results will then be compared.

This project aims to develop an Inception-Attention-BiLSTM model (IABiLSTM-NET), with hyperparameter adjustments such as batch size, learning rate, dropout rate etc. In addition, the evaluation of the model will include metrics such as Accuracy and Loss. Moreover, performance will be assessed using Precision, Recall, F1-Score, AUC-ROC, Specificity, Sensitivity, and the Confusion Matrix. Moreover, this model will be compared with some pretrained models.

Lastly, the project will be deployed through a GUI which allows uploading solar cell thermal images and then give the classification results.

1.4 Project Overview

This section explores the potential of Inception-Attention-BiLSTM model for solar cell detection, highlighting key stakeholders who will benefit from these advancements in early detection of solar cells.

1.4.1 Scope

This model will be trained to identify failure points which can negatively impact the performance of solar panels. By leveraging this model, failures can be detected early, enabling timely maintenance and repairs. As a result, the lifespan of solar panels can be significantly extended, and their energy production efficiency can be optimized.

This study holds great significance because the early detection of solar cell degradation is crucial to maintaining both the efficiency and longevity of solar energy systems. Specifically, by identifying and addressing issues before they escalate, it can lead to reduced maintenance costs, improved performance of solar panels, and ultimately contribute to the sustainability and cost-effectiveness of solar energy solutions. Moreover, this approach can help mitigate downtime and enhance the overall reliability of solar power systems, making them more viable and efficient for long-term use.

1.4.2 Audience

The outcomes of this project will benefit several key stakeholders, including:

1. **Solar energy companies and operators:** This model can help reduce operational costs, improve solar energy efficiency, and extend the lifespan of solar panels, ultimately leading to better long-term performance.
2. **Countries and governments:** Governments and nations can leverage the project's results to advance the development of renewable and clean energy solutions. By reducing dependence on fossil fuels, countries can enhance energy security and sustainability. Additionally, this technology can help governments lower the overall costs associated with energy production.
3. **Consumers of solar energy:** End users will benefit from improved solar panel efficiency and extended lifespan, which can lead to lower energy costs over time. With more reliable and efficient systems, consumers can enjoy long-term savings and increased energy independence.
4. **Environmental organizations:** By improving the efficiency and longevity of solar panels, the project supports the broader goal of reducing environmental impact and promoting renewable energy sources that help protect the planet.

Chapter 2 Background Review

There have already been several studies focused on detecting the degradation of solar panels. Through advanced data analysis and early detection, these studies aim to improve the maintenance of solar panels, extend their lifespan, and enhance overall energy efficiency, contributing to the sustainability and cost-effectiveness of solar energy systems.

Koshy et al. [12] proposed a deep learning techniques, including ResNet and custom CNNs, achieving an average prediction accuracy of 94% and 12 parameters were classified with 86% accuracy. However, the limitations of the model are that the data is unbalanced and the computational resource is limited. Wang et al. [13] proposed an improved Mask R-CNN, achieving Precision of 75.04%, Recall of 97.4% and mAP of 82.45% improving 13.4% precision, 7.9% recall and 10.81% mAP from Mask R-CNN. But it lacks validation in real-world engineering environments. Haidari et al. [14] proposed a modified VGG16 used for fault classification in a photovoltaic module, achieving a total accuracy of 0.98 and performance good in validation set. The drawback of this model is that the training data includes images with varying distances and resolutions. Dunderdale et al. [15] proposed a method based on deep learning and features to detect and classify defective photovoltaic modules using thermal infrared images. They used VGG-16, MobileNet and SVM in the research and achieve 91.2% of accuracy and can differentiate between defects with up to 89.5% of accuracy. But the dataset used in this project is small. Shaik et al. [2] proposed a novel deep learning architecture for the segmentation of solar plant aerial images, along with a transfer learning-based model for classifying solar panel damage by U-Net and VGG-19. Yousif and Al-Milaji [16] proposed an end-to-end deep learning model that integrates both handcrafted and automatically extracted features to enhance the classification accuracy of photovoltaic (PV) images and achieve an overall accuracy of 90.55% on the photovoltaic image classification task. For the healthy category, the precision, recall, and F1-score were 90.68%, 96.26%, and 93.39%, respectively. For the faulty category, the corresponding values were 90.17%, 77.61%, and 83.42%. However, the limitation is that the dataset is not enough and the computation is complex and high-cost. Jaybhaye et al. [17] proposed a deep learning model based on an improved YOLO-FIRI architecture for damage detection and localization in thermographic images of solar panels. The model achieved an overall mAP@0.5 of 0.69, with a mAP@0.5 of 0.64 for single-cell panels and 0.75 for multi-cell panels. The precision and recall were 0.40 and 0.46, respectively.

The limitations lie in its limited generalization capability, small dataset size, and high computational complexity.

Author	Datasets	Methods	Results
Koshy et al. [12]	InfraredSolarModules	ResNet & custom CNNs	Average prediction accuracy = 94% Classified accuracy = 86%
Wang et al. [13]	Some open-source datasets and a self-collected infrared image dataset of monocrystalline and polycrystalline photovoltaic modules.	improved Mask R-CNN	Precision = 75.04% Recall = 97.4% mAP = 82.45%
Haidari et at. [14]	The datasets contain thermal images of photovoltaic modules obtained from aerial and terrestrial images	modified VGG16	Accuracy = 0.98 Precision = 0.98 Sensitivity = 0.98 F1 score = 0.98
Dunderdale et al. [15]	Obtain from three different PV plants	VGG-16, MobileNet and SVM	Overall Accuracy: 91.2% differentiate between defects with up to 89.5% of accuracy.
Shaik et al.[2]	A dataset of 3580 images (frames) captured at 30 fps across 12 countries	U-Net and VGG-19	Accuracy = 98% F1 score = 99%

	around the globe.		
Yousif and Al-Milaji [16]	EL image dataset	VGG-16 and HOG	Accuracy = 90.55% F1 score = 93.39%
Jaybhaye et al. [17]	1509 thermal images	improved YOLO-FIRI	mAP@0.5 of 0.69, with a mAP@0.5 of 0.64; precision and recall were 0.40 and 0.46

Table 2. Summary of Existing Researches

Chapter 3 Methodology

3.1 Approach

The proposed model is a hybrid deep learning architecture combining Inception modules, Bidirectional LSTM (BiLSTM) and Attention mechanism to effectively capture both spatial and temporal features from thermal infrared images of solar panels. The architecture is composed of several key components, each playing a specific role in enhancing the model's performance for solar panel defect detection. In the following sections will provide a detailed explanation of each individual module and its function within the overall network architecture.

3.1.1 Inception Module

Traditional Convolutional Neural Networks (CNNs) extract features by stacking a series of convolutional layers with fixed kernel sizes. However, this fixed-size kernel design may not effectively capture both local and global information simultaneously. Therefore, the Inception network was proposed to address these limitations. The key idea of the Inception module is to use multiple filter sizes (1×1 , 3×3 , and 5×5) instead of just one fixed size, then concatenate them along with the output of the max-pooling layer and pass them to the next layer [18]. The defects in the dataset are spatially distributed in an irregular manner, so using different convolution kernels helps extract features more comprehensively.

The inception module in the proposed model, the first convolution branch uses 1×1 filter size which aims to transform the number of channels quickly and extract local information. Then the second filter size is 3×3 , and this branch is used to extract medium-range features. And the third filter size is 5×5 , which requires bigger calculation than 3×3 filter size. Next, the pooling layer used to enhance invariance and suppress noise. At last, concatenate these four branches along the channel dimension. In the proposed model, the Inception module takes an input feature map x with a shape of (H, W, C) , representing height, width, and the number of channels. It processes the input through four parallel branches: 1×1 , 3×3 , 5×5 convolutions, and a pooling operation. The outputs from these branches are then concatenated along the channel dimension ($axis = -1$) to form a new, fused feature map. As a result, the module outputs a wider and more enriched representation, which can be further processed by subsequent layers. Figure 10 shows the inception module in the proposed model.

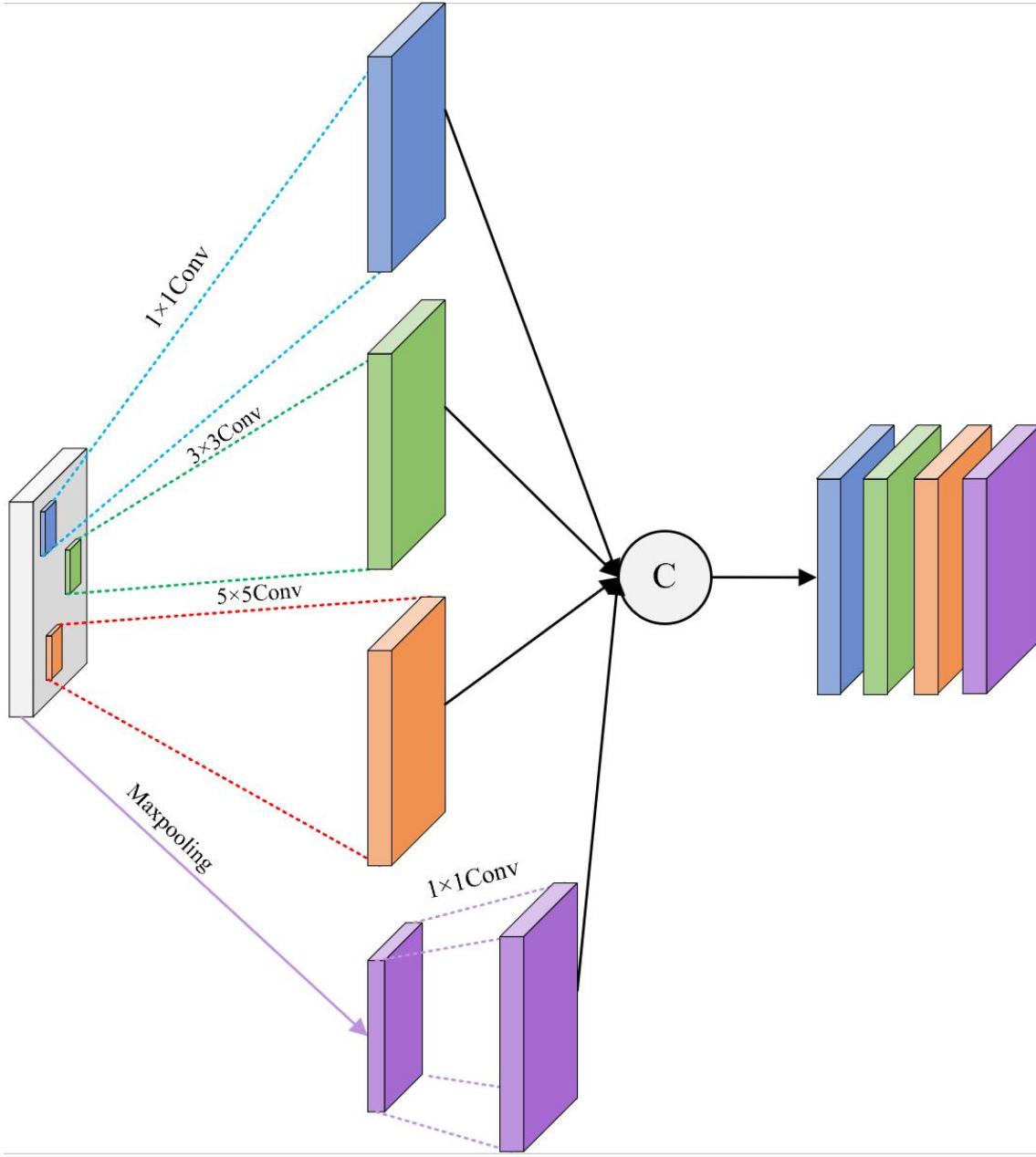


Figure 10. The Inception Module of the Proposed Model

3.1.2 Attention Block

The Attention Mechanism is a technique that enables neural networks to "selectively" concentrate on the most relevant parts of the input data. In addition, the origin of modern attention mechanisms is commonly credited to developments in the field of natural language processing [19]. This ability to focus on key information allows attention mechanisms to greatly enhance the model's performance on complex tasks and helps the model identify which parts are more important for the final prediction automatically.

For thermal imaging, certain time steps or image features may contain more critical information. Through Attention, the model can automatically identify and focus on these key features. Therefore, the attention mechanism helps improve the model's prediction accuracy.

The attention block in the proposed model is a sequential attention mechanism designed to allow the model to automatically focus on the most informative time steps in the sequential input (derived from the BiLSTM output). First, the attention scores are calculated, with an output dimension of (batch size, sequence length, 1). These scores are then flattened into a weight distribution and normalized using the softmax activation function, forming a set of weighted coefficients that represent the importance of each time step. Next, the softmax output is reshaped back to the time step dimension of the original input structure. Finally, the attention weights are multiplied with the input features at each time step, highlighting the important time step information. Figure 11 shows the attention block in the proposed model.

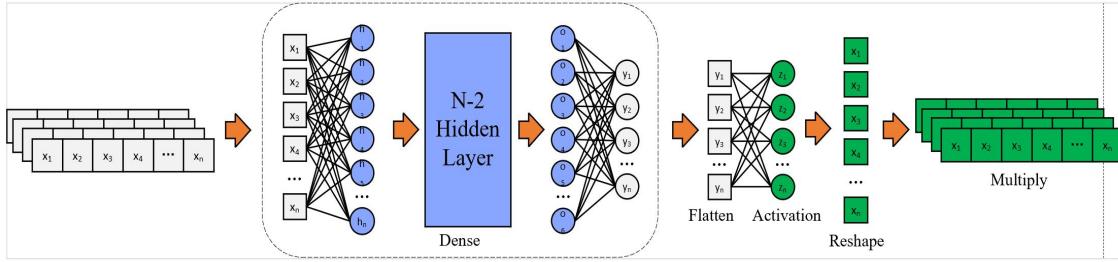


Figure 11. The Attention Block in the Proposed Model

3.1.3 BiLSTM Module

Bidirectional Long Short-Term Memory (BiLSTM) is an extension of the traditional Long Short-Term Memory (LSTM) network model, designed to process sequential data. By simultaneously learning features in both the forward and backward directions of the sequence, it is able to capture more information. In addition, BiLSTM uses two LSTM networks simultaneously, one processing the sequence in the forward direction (from left to right) and the other in the backward direction (from right to left), this allows the model to capture the relationships and dependencies between previous and subsequent time steps in the sequence. Especially when dealing with data that has long-term dependencies, it can significantly improve the model's performance.

The BiLSTM module in the proposed model uses the Reshape Layer to convert the previous feature map into a two-dimensional sequence input suitable for BiLSTM processing. Then, a Bidirectional BiLSTM layer is defined with 64 BiLSTM units, and return sequences equals the True which indicates that the LSTM will return the output at each time step, rather than only the output at the final time step. This allows subsequent layers, such as the Attention mechanism and the fully connected layer, to process the output from each time step. Finally, a Dropout Layer is added to randomly drop 30% of the neurons to prevent overfitting. Figure 12 shows the BiLSTM module in the proposed model.

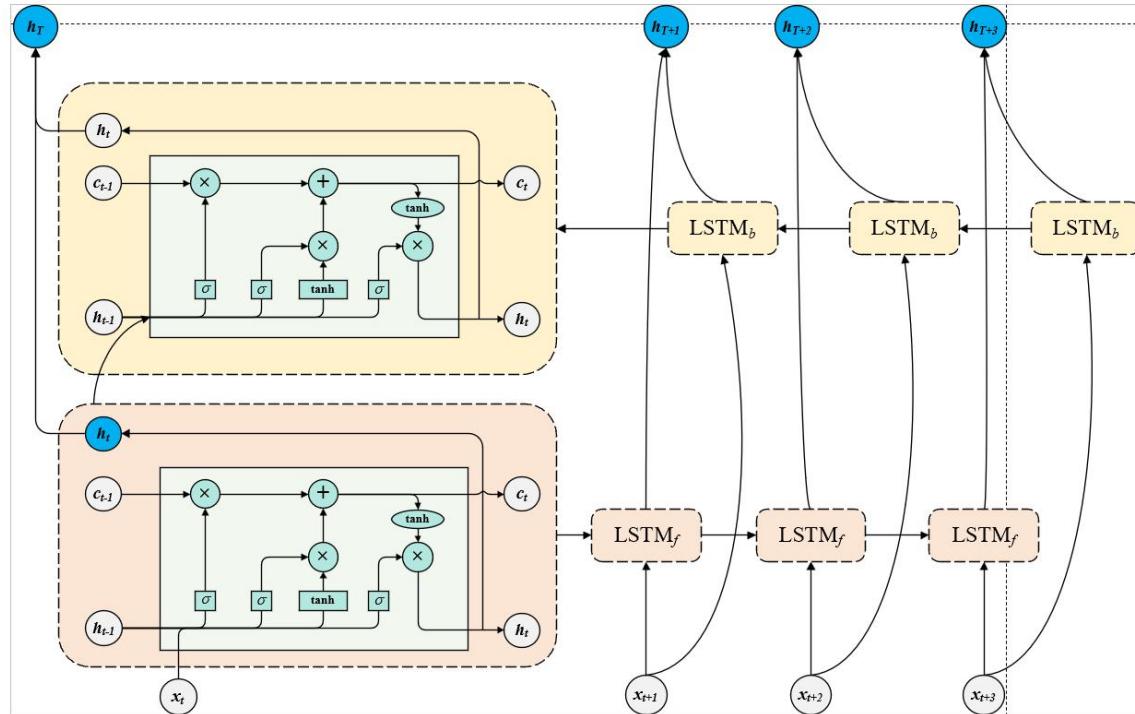


Figure 12. The BiLSTM Module in the Proposed Model

3.1.4 Proposed Model

Overall, the progress of the proposed model has five steps. The first step is to input an RGB image with the shape (128, 128, 3). Then the second step is the inception module which is used to extract features. Next, the BiLSTM is used for time series modeling, followed by the fourth step where the Attention mechanism is applied to weight the important features, and finally, after the fully connected layer and output layer, the final output is the probability of the image belonging to each category. Figure 13 shows the structure of the proposed module.

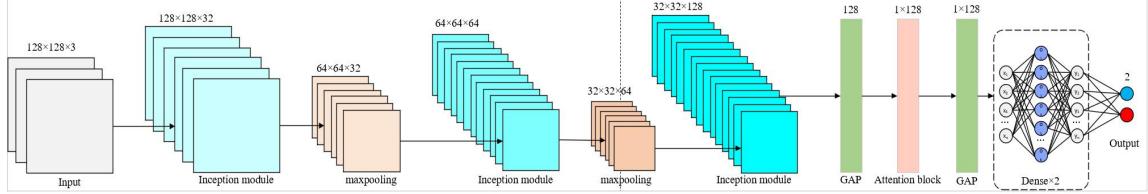


Figure 13. The Structure of the Proposed Module

3.1.5 Focal Loss Function

Focal loss function is a loss function used for binary classification tasks. In addition, focal loss is one of the popular ones in dealing with class imbalance problems as well as improving learning speed [20]. Its purpose is to address the problem of class imbalance, where the model tends to focus too much on the majority class and overlook the minority class during training. In my dataset, the number of defective samples is significantly higher than that of non-defective ones. Traditional binary crossentropy assigns equal importance to each sample, making it difficult for the model to learn from hard examples. Therefore, this model adopts focal loss to mitigate this issue.

The core idea of Focal Loss is to address class imbalance by introducing a modulation factor and class weighting on top of the traditional cross-entropy loss function. It adds a modulation factor $(1 - p_t)^\gamma$ to reduce the loss contribution from easy-to-classify examples, thereby forcing the model to focus more on hard-to-classify samples. At the same time, it introduces a class weight parament α assigning higher weights to the minority class (such as non-defective samples), which further enhances the model's ability to learn from these underrepresented classes. In this way, Focal Loss effectively mitigates the issue of the model being dominated by a large number of easy negative samples during training, and improves the recognition performance on hard and minority class samples.

In this implementation of focal loss, the predicted output values are first clipped using to avoid numerical instability caused by $\log(0)$. Then, the cross-entropy loss for the positive class (i.e., label = 1) is computed. To reduce the focus on easily classified samples, a modulation factor is introduced where its value decreases as the prediction becomes more accurate, and increases when the prediction is wrong, thus assigning more weight to hard-to-classify samples. Next, the loss is further adjusted by multiplying with the class weighting factor alpha, which increases attention on the minority class samples (e.g., non-defective class, label = 0). The product of these components forms the focal

loss for each sample. Finally, a sum is taken along the last dimension to obtain the total loss value for each sample. This process, by modulating difficulty and applying class weights, guides the model to handle class imbalance more effectively.

3.2 Dataset

This part will introduce the dataset information include dataset introduction, data separation, data resize and data augmentations.

3.2.1 Dataset Introduction

The PV failure detector dataset chosen for this project is the solar cells' thermal images. This dataset provides 5,352 samples of infrared image of PV panel, categorized into defect solar cell and non-defect solar cell. Each RGB image has a resolution of 640 * 640 pixels. This dataset, created by the author named E555 Solar Project. In this dataset, an image sample may contain no anomalies, one anomaly label, or multiple anomaly labels. The infrared anomaly labels do not distinguish between temperature anomalies, electrical faults, scratches, damages, etc., and are all uniformly labeled as Thermal Defects. In the annotations, it includes the basic information like filename, path and source. And the image size like width, height and depth. Moreover, the object includes the type name (thermal defect) and other information. In the dataset, the type name is used to judge the status of the solar cell. If the XML file include the 'thermal defect', it means this image is defect, and if not include 'thermal image', it means there's no defect in this panel. Figure 14 displays examples of solar cell thermal images.

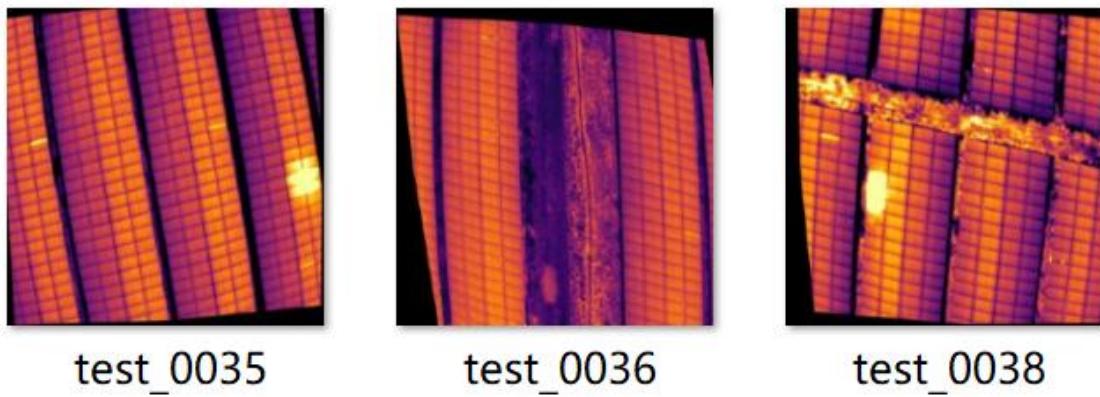


Figure 14. Examples of Solar Cell Thermal Images

3.2.2 Data Separation

This dataset has 4,733 defective images and 619 non-defective images. In this project, three different partition ratios are used since the sample data shows significant variation. In addition, every set need both the defective images and non-defective images. Different splitting ratios can help avoid overfitting and underfitting, thereby improving the model's generalization ability.

The first ratio is 80% for training, 10% for validation, and 10% for testing. In this ratio, the training set accounts for 80%, ensuring that the model has more samples for training, allowing it to learn more features. Additionally, there are enough samples in the validation and test sets to evaluate the model's performance. Figure 15 shows the structure of the dataset split 80:10:10 ratio. Figure 16 shows the class distribution of split 80:10:10.

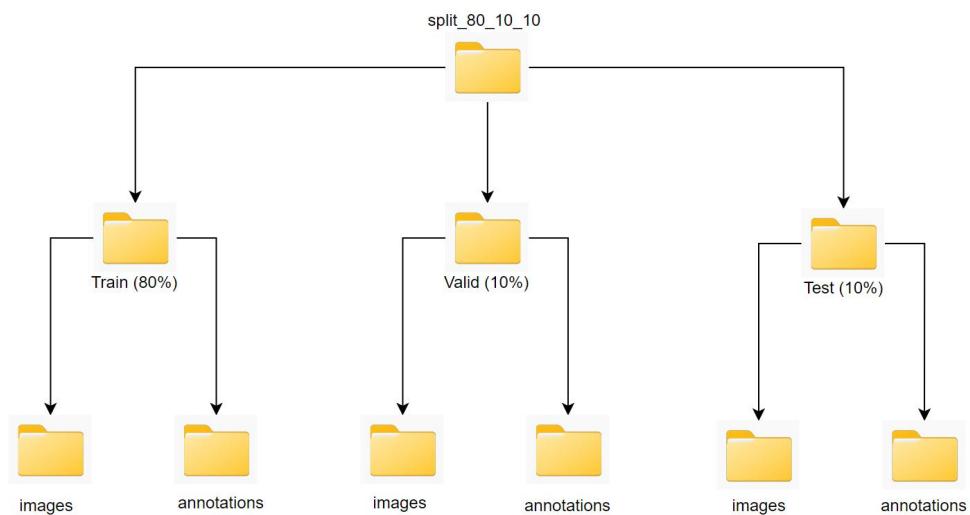


Figure 15. The Structure of the Dataset Split 80:10:10 Ratio

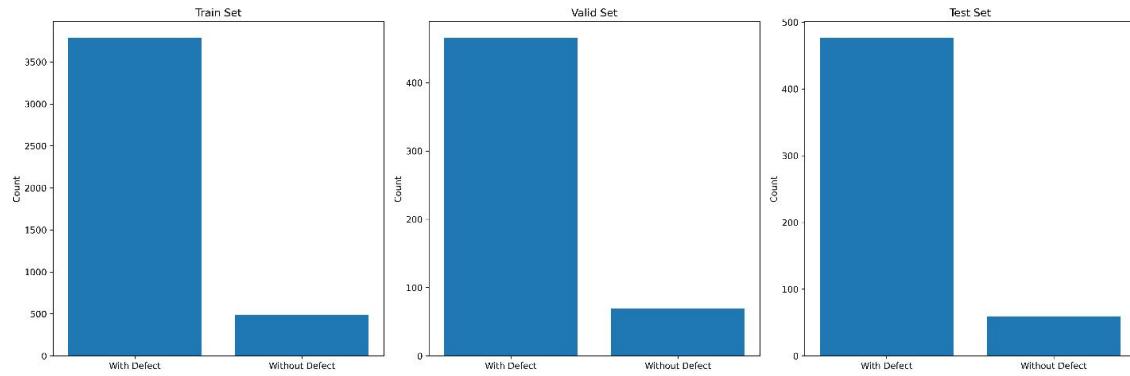


Figure 16. The Class Distribution of Split 80:10:10 Ratio

The second ratio is 70% for training, 20% for validation, and 10% for testing. In this split ratio, 20% validation set provides more information to adjust hyperparameters or implement early stopping, helping to avoid overfitting. Figure 17 shows the structure of the dataset split 70:20:10 ratio. Figure 18 shows the class distribution of split 70:20:10.

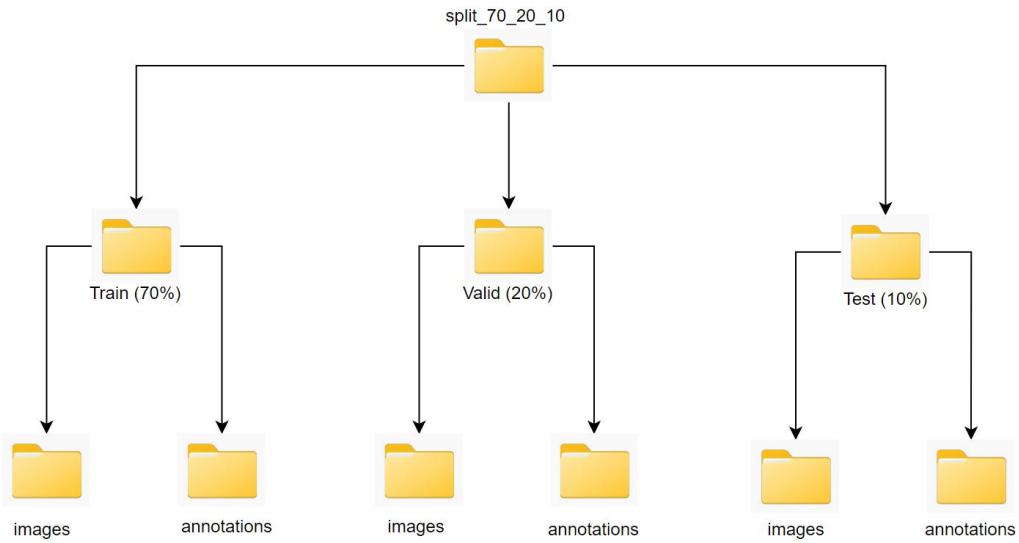


Figure 17. The Structure of the Dataset Split 70:20:10 Ratio

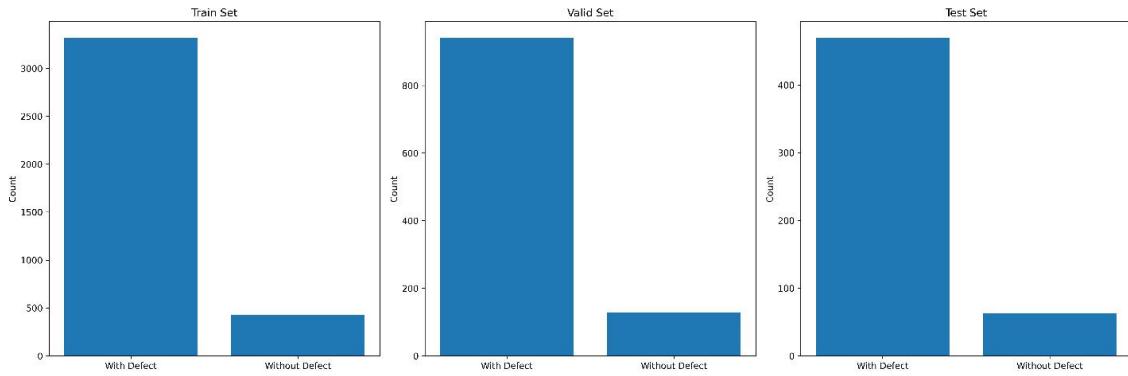


Figure 18. The Class Distribution of Split 70:20:10 Ratio

The third ratio is 60% for training, 20% for validation, and 20% for testing. The proportion of the validation set helps better detect overfitting, ensuring the model's generalization ability on unseen data. It also provides enough data for tuning and ensures the reliability of the evaluation results with the test set. Figure 19 shows the structure of the dataset split 60:20:20 ratio. Figure 20 shows the class distribution of split 60:20:20.

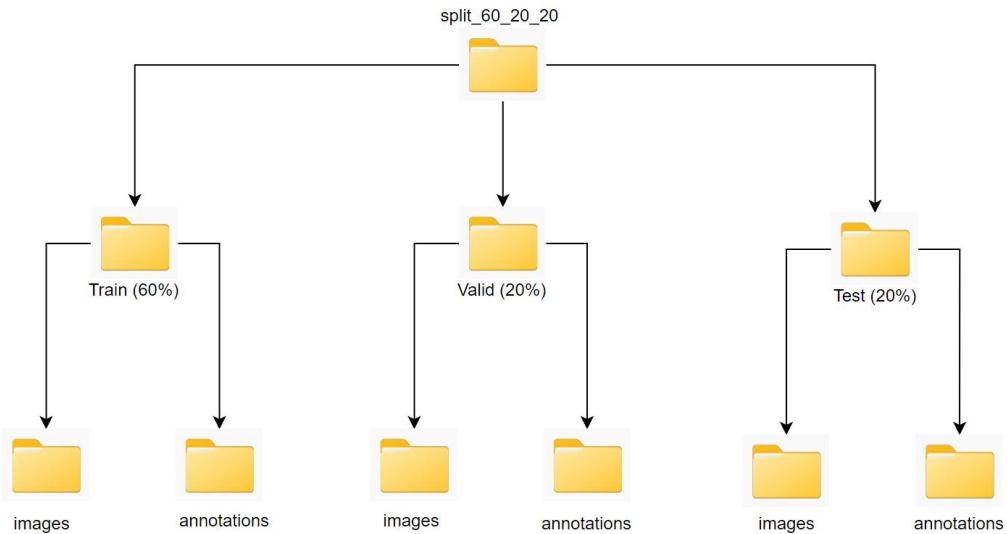


Figure 19. The Structure of the Dataset Split 60:20:20 Ratio

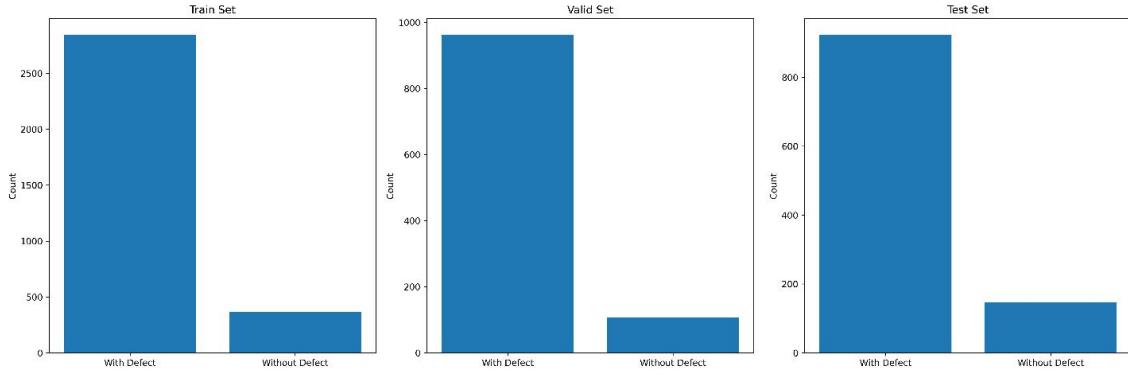


Figure 20. The Class Distribution of Split 60:20:20 Ratio

3.2.3 Data Resize

As the introduction mentioned, the image resolution is $640 * 640$ pixels, which does not fit in the proposed model input. And therefore, this project had resized the images of all magnifications into $128 * 128$.

3.2.4 Data Augmentation

The data augmentation methods used in this code are part of the `ImageDataGenerator` class from TensorFlow's Keras library. These augmentations help improve the model's generalization ability by artificially increasing the diversity of the training dataset.

The dataset used in this project is augmented with seven methods. First, rescaling the pixel values of the images to a range of $[0, 1]$ by dividing the original pixel values (ranging from 0 to 255) by 255. Second, rotating the image by a degree randomly between -30 and +30 degrees, which helps the model become invariant to slight rotations. Third, shifting the image horizontally by a factor of 0.2 (i.e., 20% of the image width) randomly. This helps the model learn to be robust to slight shifts in position. Then, the fourth method is that randomly shifts the image vertically by a factor of 0.2 (i.e., 20% of the image height). Like the width shift, it helps with model robustness. In addition, apply shear transformations (shear angle up to 0.2 radians) to the image. This introduces slight distortions, which can make the model more robust to such variations. Moreover, randomly zooming into or out of the image by up to 20%. This introduces variations in scale. At last, randomly flipping the image horizontally, which is particularly useful in tasks where left-right symmetry is present in the dataset (like many image classification tasks).

For the validation and test data generators, only rescaling is applied (rescale=1. / 255), which ensures the images are scaled to the [0, 1] range but no other augmentations are applied. This is typically done to ensure that the validation and test sets represent data in its original form. Figure 21 displays the augmentation of the image.

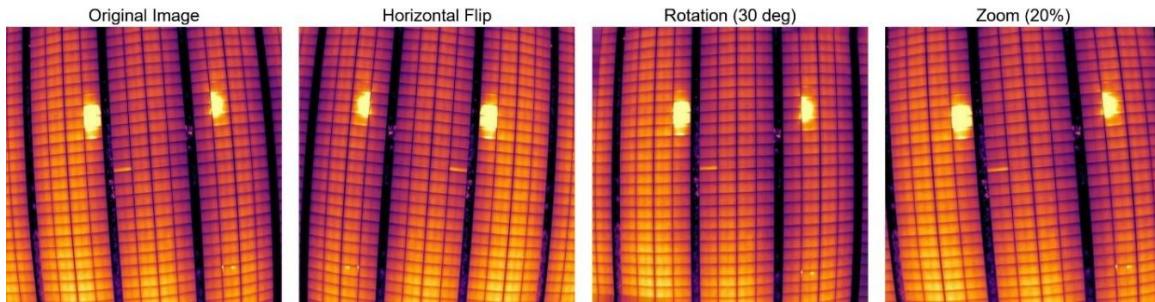


Figure 21. The Augmentation of the Images

3.3 Technology

The technology used in this project is displayed in Table 3.

Software	Framework	Tensorflow Cuda CuDNN
	Language	Python
	Libraries and Application	Numpy, Keras, Matplotlib
Hardware	Central processing unit (CPU)	Intel® Core(TM) i9-14900HX processor, with a base clock speed of 2.20 GHz and a maximum turbo frequency of 5.40 GHz
	Graphic Processing Unit (GPU)	Nvidia GeForce RTX 4070 GPU

Table 3. Summary of Relevant Technology involved in this project

3.4 Data Testing and Evaluation Plan

3.4.1 Data Testing

By checking the datasets downloaded from roboflow, it is estimated that some techniques of data pre-processing will be implemented which will be displayed below.

- Check the dataset sample ratios:** the dataset is divided into three different ratios: The first ratio is 80% for training, 10% for validation, and 10% for testing. The second ratio is 70% for training, 20% for validation, and 10% for testing. The third ratio is 60% for training, 20% for validation, and 20% for testing. Each of the train, validation, and test sets must contain both defective and non-defective images.
- Check the image size:** the images in this dataset should use resize method to change the size from 640 x 640 pixel to 128 x 128 pixel.
- Implement the augmentation method:** only the train set need augmentation and each dataset split ratio should adopt the same method.

3.4.2 Model Performance Evaluation Criteria

This project will evaluate the performance of the model through following standards.

TN: The number of actual negative samples correctly predicted as negative by the model.

TP: The number of actual positive samples correctly predicted as positive by the model.

FN: The number of actual positive samples incorrectly predicted as negative by the model.

FP: The number of actual negative samples incorrectly predicted as positive by the model.

1. Accuracy

Accuracy represents the proportion of samples that the model correctly predicts. The higher the accuracy, the stronger the model's predictive ability as expressed in equation 5.

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (5)$$

2. Focal Loss

$$FL(p_t) = -\alpha((1-p_t)^\gamma \log(p_t)) \quad (6)$$

Besides:

$$p_t = y_{true} \times y_{pred} + (1 - y_{true}) \times (1 - y_{pred})$$

(7)

γ : Controls the degree of focus on hard-to-classify samples, typically set to 2.

α : Controls the importance of positive and negative samples (usually, a higher α value is set for the minority class, such as 0.75).

3. Precision

Precision is used to measure the proportion of predicted positive samples that are actually positive in a classification model. The higher the precision, the fewer false positives the model produces as expressed in equation 8.

$$Precision = \frac{TP}{TP+FP} \quad (8)$$

4. Recall and Sensitivity

These are used to measure the model's ability to correctly identify positive class samples. The higher the recall, the more accurately the model identifies positive class samples as expressed in equation 9.

$$Sensitivity = Recall = \frac{TP}{TP+FN} \quad (9)$$

5. F1-Score

The F1-score ranges between 0 and 1, with values closer to 1 indicating higher precision and recall, and thus better model performance as expressed in equation 10.

$$F1 - score = 2 \times \frac{Precision \times Recall}{Precision + Recall} = \frac{2TP}{2TP+FP+FN} \quad (10)$$

6. ROC Curve

The ROC curve plots the relationship between the True Positive Rate (TPR) and the False Positive Rate (FPR) of a model as expressed in equation 11 and 12.

$$TPR = \frac{TP}{TP+FN} = Recall \quad (11)$$

$$FPR = \frac{FP}{FP+TN} = 1 - Specificity$$

(12)

In an ideal scenario, the ROC curve should be as close as possible to the top-left corner, indicating a high True Positive Rate (TPR) and a low False Positive Rate (FPR).

7. AUC (Area Under the ROC Curve)

AUC > 0.5: This indicates that the model perfectly classifies all samples without any misclassification.

AUC = 0.5: This means the model has no discriminative power, essentially performing as random guessing.

AUC < 0.5: This suggests that the model's performance is worse than random guessing, meaning the model is making incorrect predictions (inverse predictions).

$$AUC = \int_0^1 TPR(FPR) d(FPR) \quad (13)$$

8. Specificity

Specificity measures the ability of a classification model to correctly identify negative class samples. It represents the proportion of actual negative samples that are correctly predicted as negative by the model. The higher the specificity, the better the model can effectively exclude negative class samples and reduce the occurrence of false positives as expressed in equation 14.

$$\text{Specificity (SP)} = \frac{TN}{TN+FP} \quad (14)$$

9. Confusion Matrix

A confusion matrix is a table that contains four basic elements. It lists the comparison between the model's predicted results and the actual labels as expressed in equation 15.

$$\begin{bmatrix} TP & FP \\ FN & TN \end{bmatrix} \quad (15)$$

Chapter 4 Implementation and Results

4.1 Experiment Phases

1. Test three different split ratios on the proposed model and compare the result and decide the best split ratio to use in the further comparison.
2. Compare the split 70:20:10 dataset on the proposed model with the split 70:20:10 dataset on the inception model.

3. Compare the split 70:20:10 dataset on the proposed model with the split 70:20:10 dataset on the ‘inception_attention’ model.
4. Compare the split 70:20:10 dataset on the proposed model with the split 70:20:10 dataset on the CNN model.

4.2 Three different split ratios on the proposed model

This part will train and test the three split ratios dataset on the proposed model using the evaluation criteria and compare the result and find the best split ratio.

4.2.1 Split 80:10:10 dataset on the proposed model

1. Accuracy Curve of Split 80:10:10

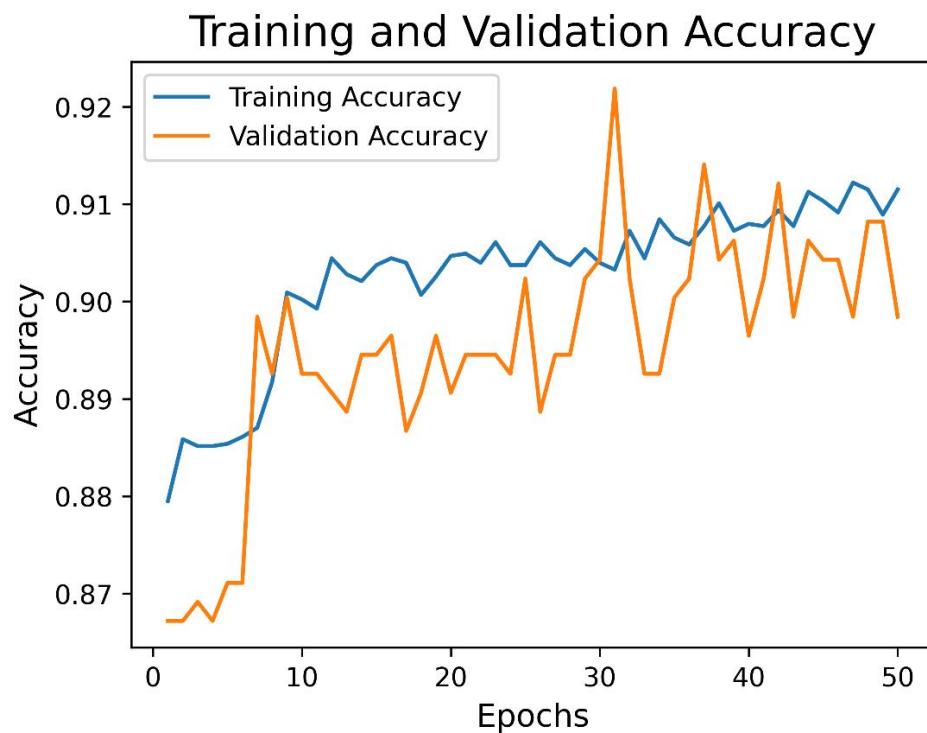


Figure 22. Accuracy Curve of Split 80:10:10

From the training and validation accuracy curves, the training accuracy (blue curve) increases rapidly from the beginning and starts to stabilize after around the 10th epoch, fluctuating between 0.90 and 0.91. Meanwhile, the validation accuracy (orange curve) initially increases slowly, then rises sharply after about the 8th epoch, showing more fluctuation between 0.89 and 0.91. In the end, both training and validation accuracies stabilize around 90%, indicating that the model performs well overall. It effectively learns and generalizes thermal image features of solar cells, and there is no obvious overfitting.

2. Loss Curve of Split 80:10:10

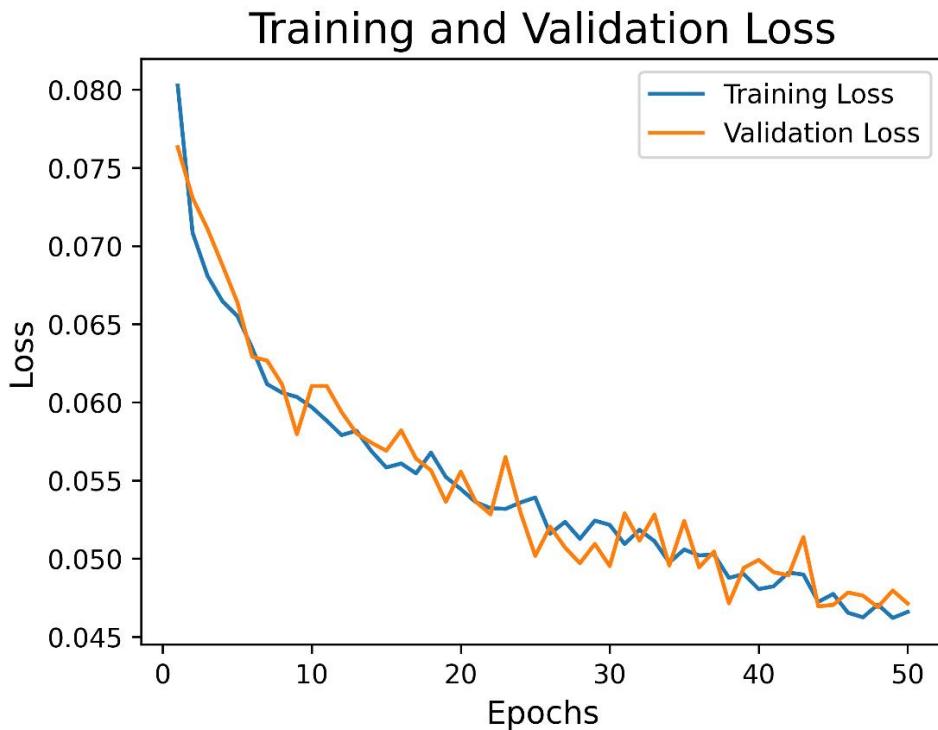


Figure 23. Loss Curve of Split 80:10:10

From the training and validation loss curves, it can be seen that from the 1st to the 50th epoch, both the training loss and validation loss remain at similar levels and steadily decrease. This indicates that the model learns well on the training set and generalizes effectively on the validation set without simply memorizing the data. Additionally, the loss curves are smooth, with no oscillations or sudden spikes, and both training and validation losses eventually converge to around 0.046 to 0.048. This suggests that the loss function converges well, and the model's output probabilities are very close to the true labels.

3. Precision and Recall Curve of Split

80:10:10

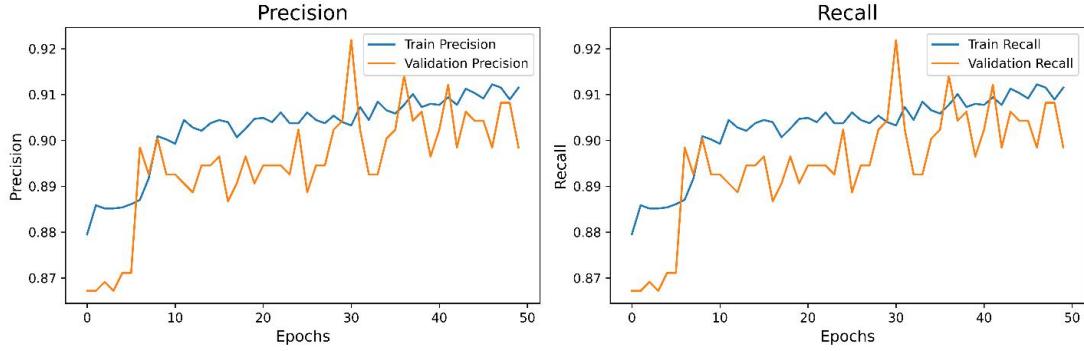


Figure 24. Precision and Recall Curve of Split 80:10:10

The precision and recall curves indicate consistent performance improvements throughout the training epochs. Both metrics reach above 0.90 on the validation set, demonstrating that the model maintains high prediction accuracy while effectively minimizing false negatives and false positives. The close alignment between training and validation curves also confirms strong generalization without signs of overfitting.

4. AUC Curve of Split 80:10:10

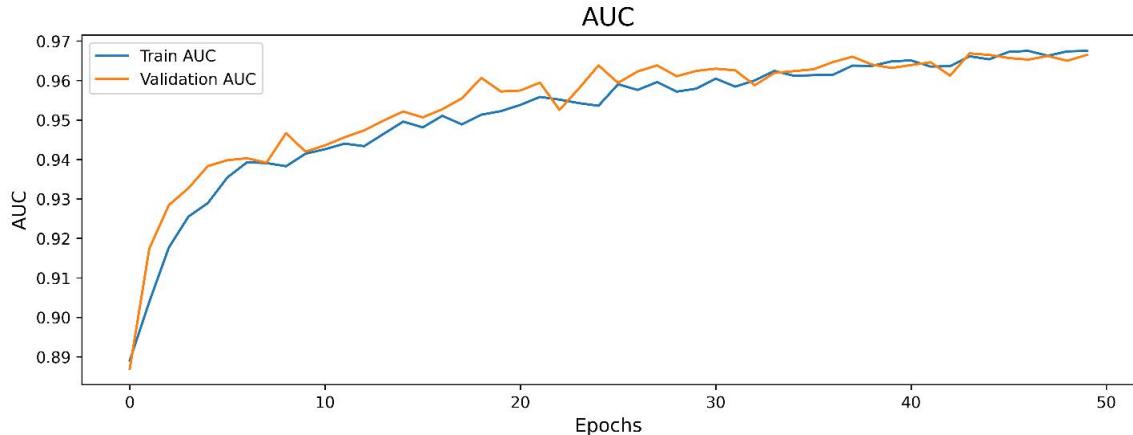


Figure 25. AUC Curve of Split 80:10:10

The model's AUC improved from 0.89 to 0.97 (train set) and 0.90 to 0.96 (validation set), showing strong class distinction and generalization. The near overlap in later epochs confirms minimal overfitting. Small fluctuations indicate effective training strategies. Overall, the model is robust and reliable for binary classification tasks.

4.2.2 Split 70:20:10 dataset on the proposed model

1. Accuracy Curve of Split 70:20:10

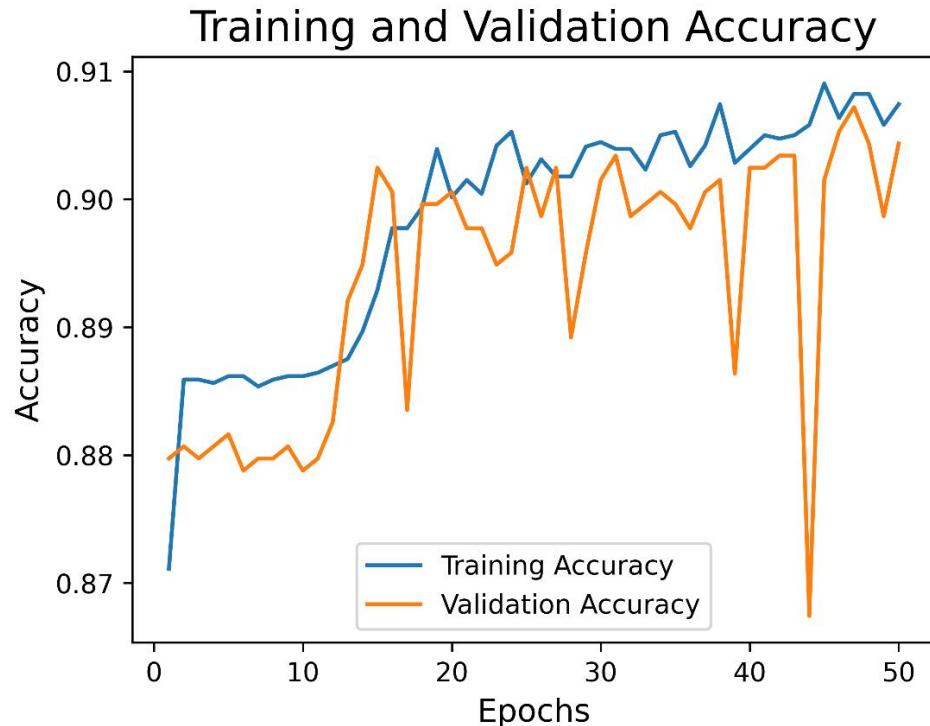


Figure 26. Accuracy Curve of Split 70:20:10

The accuracy curves show strong model performance: training accuracy rises quickly from 0.87 to ~0.91, while validation accuracy steadily improves from 0.87 to ~0.90. Both curves remain stable with minimal gap (less than 0.01), indicating no overfitting and good generalization. The model effectively learns and classifies solar cell thermal images.

2. Loss Curve of Split 70:20:10

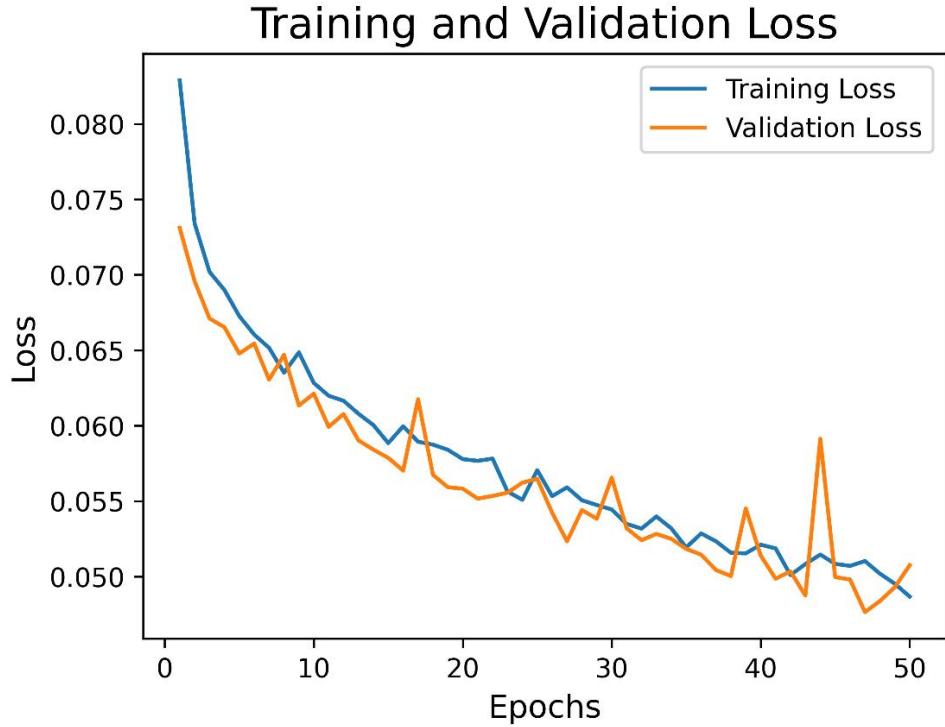


Figure 27. Loss Curve of Split 70:20:10

The loss curves demonstrate stable and effective training. Training loss decreased from 0.080 to 0.050, while validation loss followed closely, ending around 0.050 to 0.055. Both curves show smooth convergence with minimal gap (less than 0.005), indicating no overfitting and strong generalization.

3. Precision and Recall Curve of Split 70:20:10

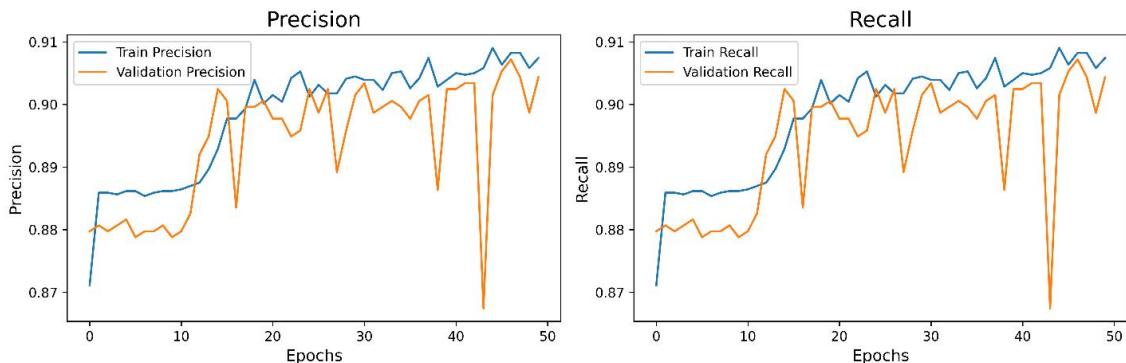


Figure 28. Precision and Recall Curve of Split 70:20:10

The precision and recall curves show a slight downward trend. Train precision drops from 0.89 to 0.87 and recall from 0.91 to 0.87, suggesting the model becomes more

conservative in identifying positives over time. The gap between training and validation metrics is minimal (~0.01), indicating stable generalization with no clear overfitting. Overall, the model maintains a good balance between precision and recall, suitable for solar cell defect detection.

4. AUC Curve of Split 70:20:10

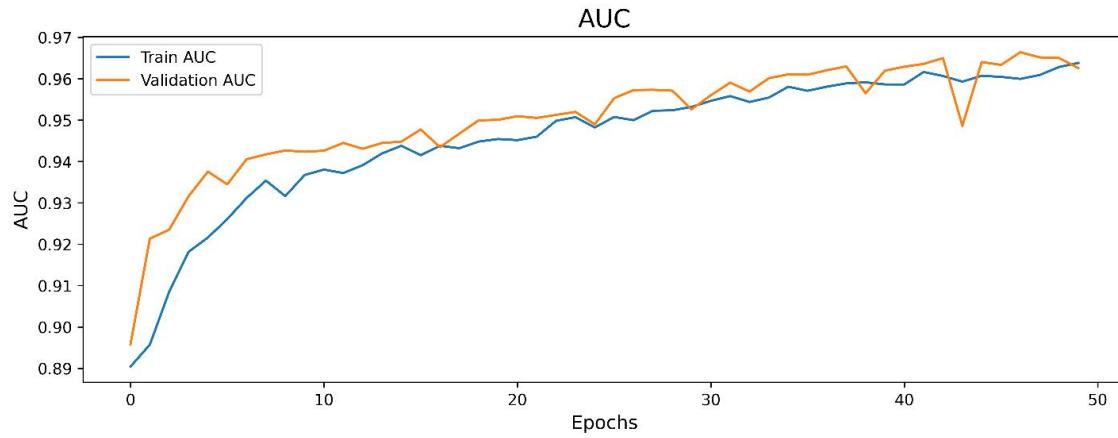


Figure 29. AUC Curve of Split 70:20:10

The AUC curve shows a steady performance improvement, with training AUC rising from 0.89 to 0.97 and validation AUC from 0.90 to 0.96. Both curves align closely, indicating good convergence and strong generalization. Early rapid gains slow over time, suggesting effective learning and eventual stabilization near optimal performance.

4.2.3 Split 60:20:20 on the proposed model

1. Accuracy Curve of Split 60:20:20

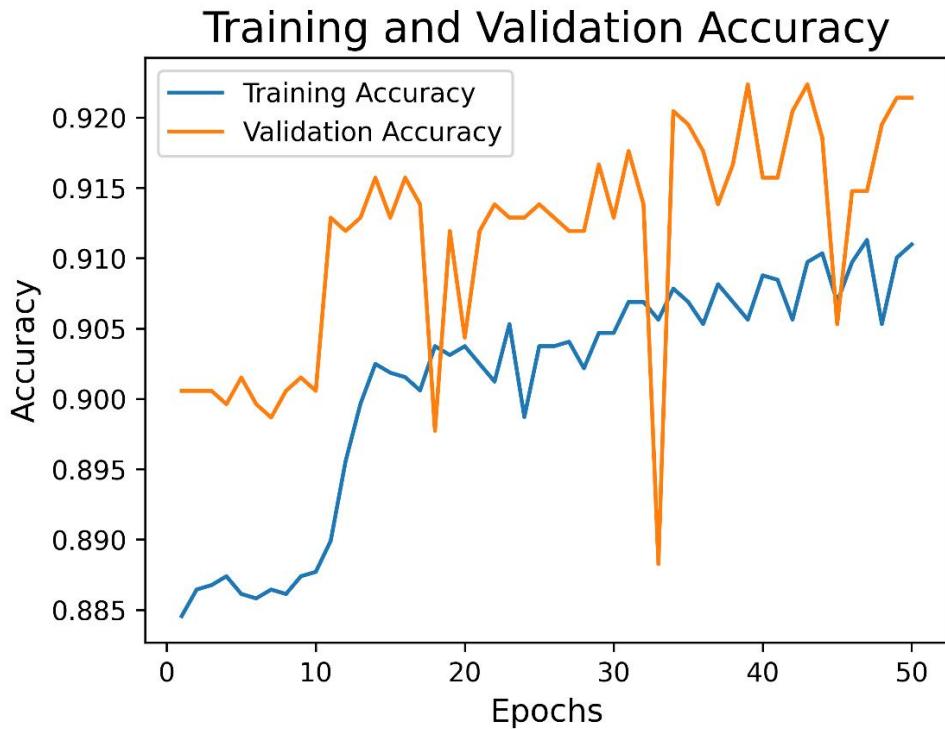


Figure 30. Accuracy Curve of Split 60:20:20

The accuracy curves show a steady and synchronized rise in training (from 0.885 to 0.920) and validation accuracy (from 0.890 to 0.915), indicating stable learning and strong generalization. Minor fluctuations around Epoch 30 are quickly corrected. The final accuracy exceeding 90% on both sets reflects the model's high reliability in classifying solar cell defects using thermal images.

2. Loss of Split 60:20:20

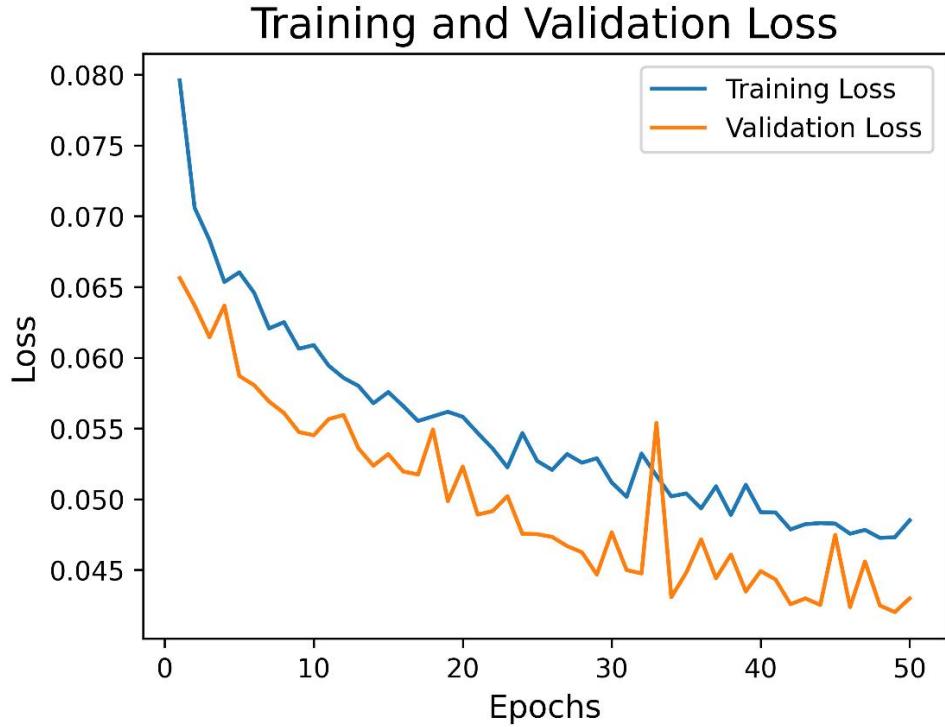


Figure 31. Loss of Split 60:20:20

The loss curves show a steady decline in training (from 0.080 to 0.045) and validation loss (from 0.075 to 0.050), indicating stable learning and effective error minimization. Both curves remain closely aligned with minimal fluctuation, suggesting strong generalization and no overfitting. Final convergence highlights the model's robustness and reliability in learning solar cell defect patterns from thermal imagery.

3. Precision and Recall Curve of Split 60:20:20

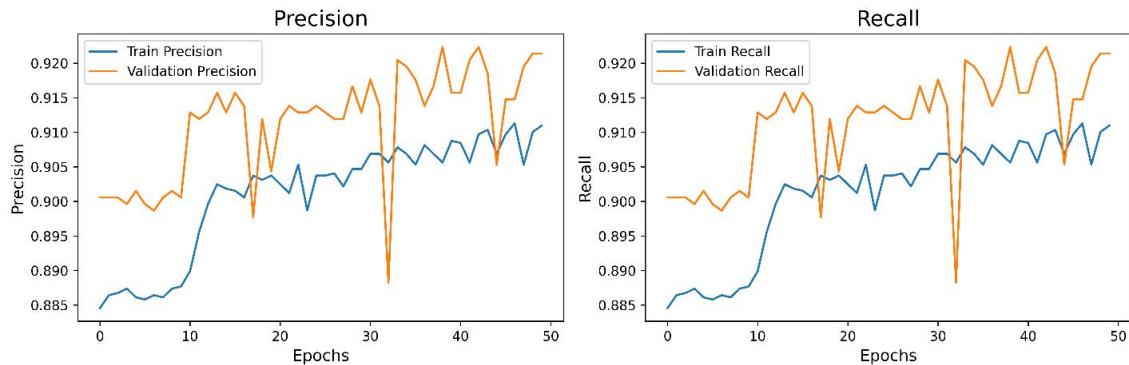


Figure 32. Precision and Recall Curve of Split 60:20:20

Both precision and recall steadily increased from 0.885 to 0.920 (train set) and from 0.890 to 0.915 (validation set), showing improved detection and reduced false positives/negatives. The curves remained smooth with minimal fluctuations, indicating strong generalization, robust training, and balanced performance. Minor mid-training dips were quickly corrected, reflecting stable optimization and effective model design.

4. AUC Curve of Split 60:20:20

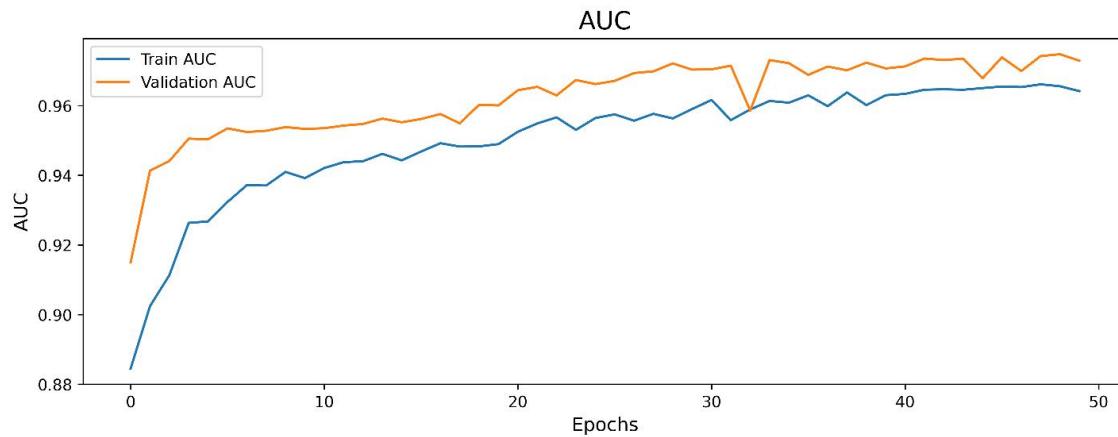


Figure 33. AUC Curve of Split 60:20:20

The AUC steadily rose from 0.88 to 0.96 (train set) and 0.88 to 0.94 (validation set), showing strong model improvement and excellent class discrimination. Curves remained smooth and synchronized, indicating stable learning and high generalization. Minor mid-training dips were brief, with no signs of overfitting. The model demonstrated robustness and effective convergence throughout training.

4.3 Comparison of three split ratio

4.3.1 Accuracy Comparison of three split radio

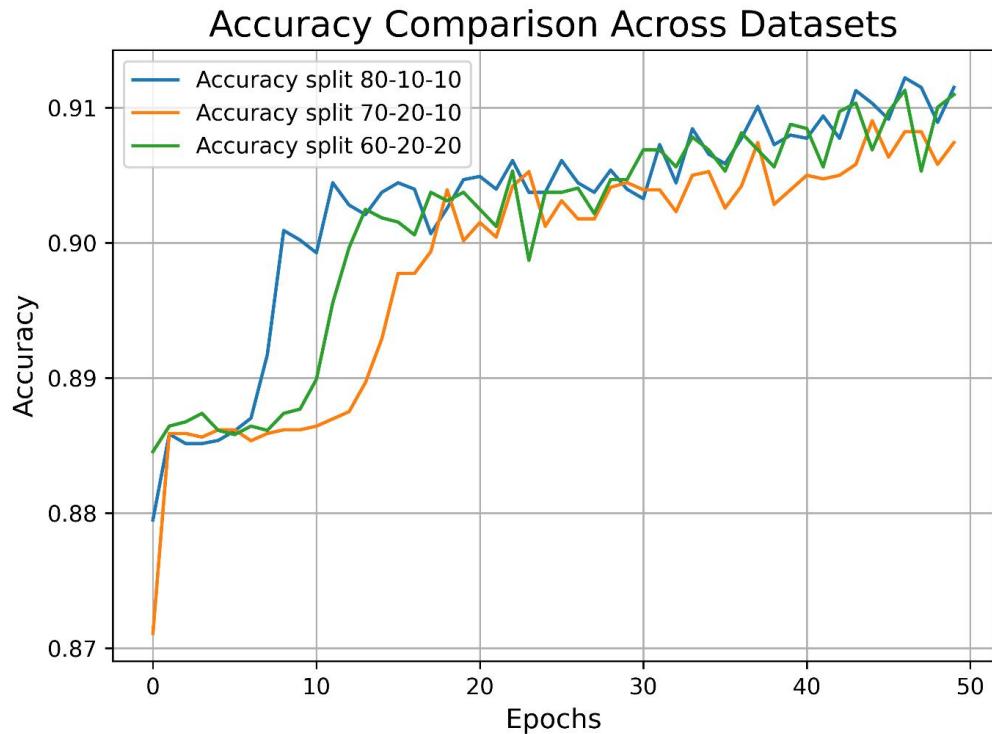


Figure 34. Accuracy Comparison of three split radio

This chart compares model accuracy under three dataset splits: 80:10:10, 70:20:10, and 60:20:20. All curves stabilize after early fluctuations. The 80:10:10 and 70:20:10 splits achieve higher accuracy (~0.91 to 0.92), while the 60:20:20 split is slightly lower (~0.89). Overall, the model performs well across all settings, maintaining accuracy above 0.9.

4.3.2 Loss Comparison of three split radio

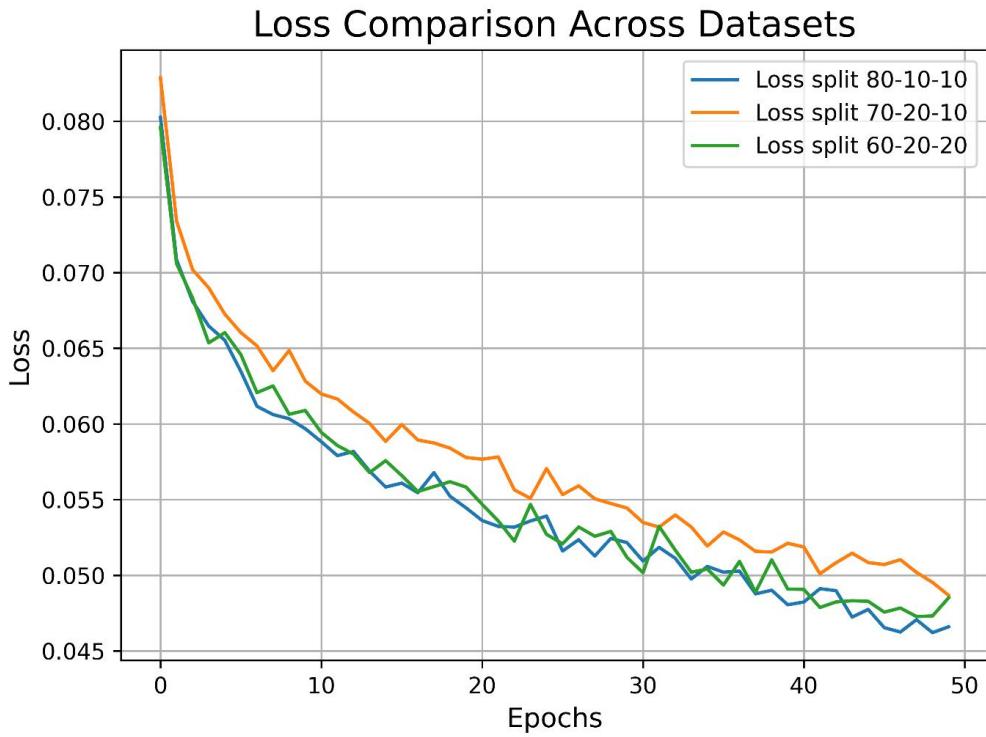


Figure 35. Loss Comparison of three split radio

All lines show a decreasing trend, indicating model improvement. The 80:10:10 split (blue) consistently achieves the lowest loss (dropping from ~ 0.080 to ~ 0.046), suggesting better optimization. The 70:20:10 and 60:20:20 splits show higher but steadily declining losses. All curves stabilize in later epochs, reflecting training convergence.

4.3.3 AUC Comparison of three split radio

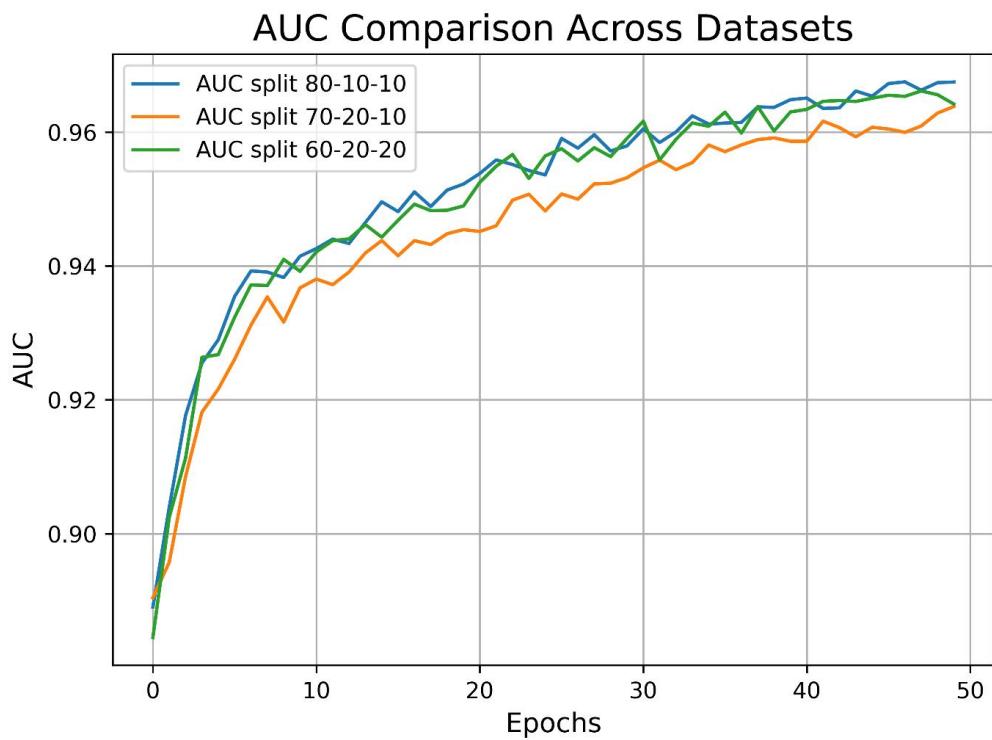


Figure 36. AUC Comparison of three split radio

The chart highlights that all splits lead to improved AUC over time, confirming successful model learning. However, the 80:10:10 split consistently delivers the best performance, achieving the highest AUC and showing a faster and more stable convergence. This suggests that allocating more data to training (as in the 80:10:10 split) may result in better generalization and discrimination ability for this model.

4.3.4 F1 Score Comparison of three split radio

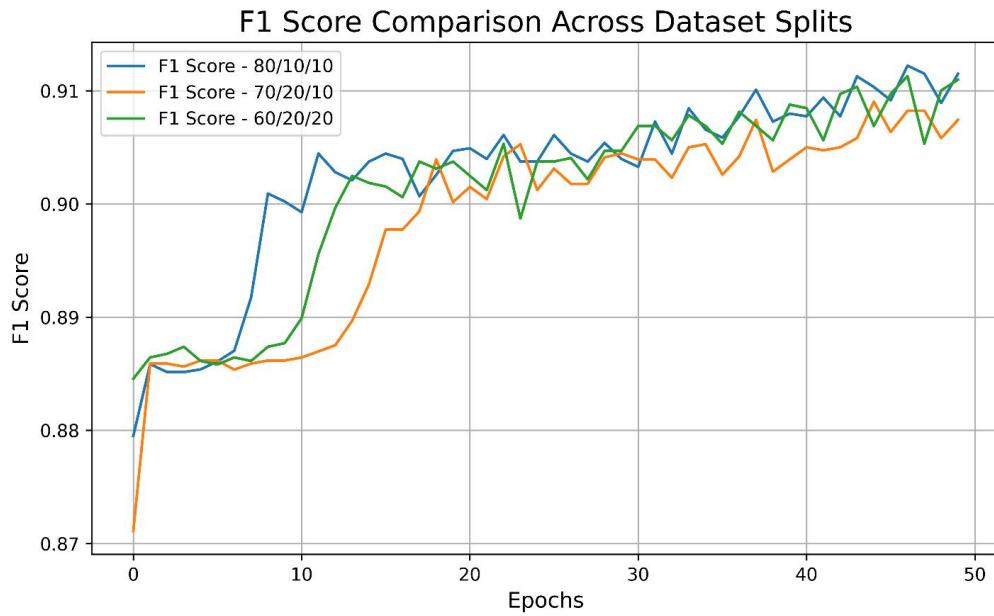


Figure 37. F1 Score Comparison of three split radio

This chart demonstrates that increasing the training set size (as with the 80:10:10 split) can lead to more stable and higher F1 scores in later stages of training. While all three dataset splits allow the model to achieve competitive performance, the 80:10:10 configuration is more effective for optimizing F1 score, especially in the long run.

4.3.5 Confusion Matrix Comparison of three split ratio

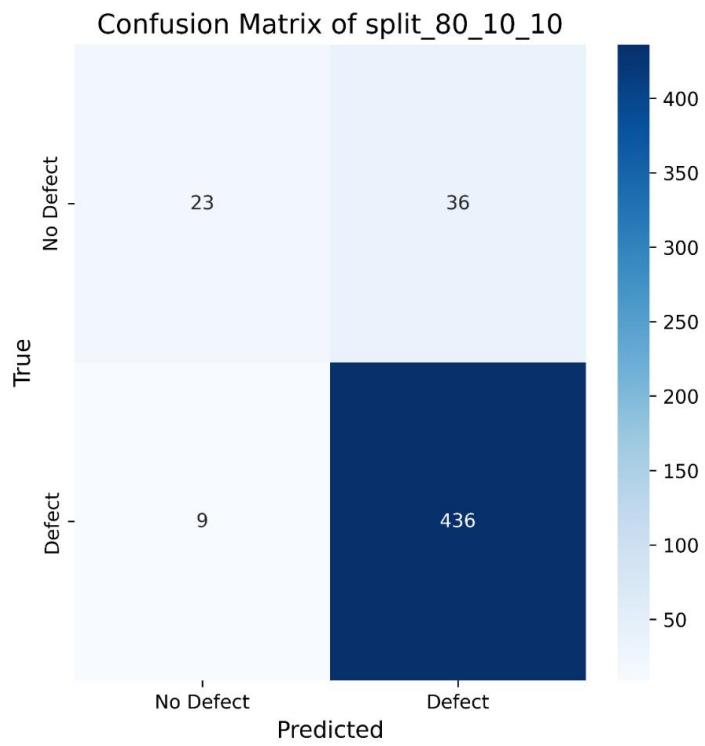


Figure 38. Confusion Matrix of split 80:10:10

The model performs well in predicting defective samples, as the value in the bottom right corner (436) is relatively high, while the value in the bottom left corner (9) is relatively low. This indicates a low error rate when predicting defective samples. However, when predicting non-defective samples, 36 of them were incorrectly classified as defective.

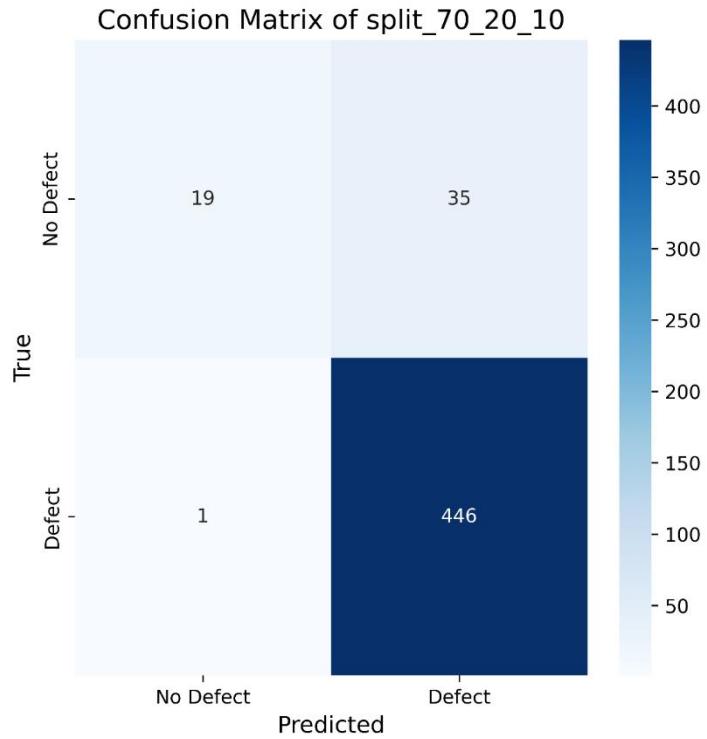


Figure 39. Confusion Matrix of split 70:20:10

The model performs well in predicting defective samples, correctly identifying the majority of defective samples (446), but makes some errors when predicting non-defective samples, especially with 35 non-defective samples being incorrectly predicted as defective, and 1 defective sample being predicted as non-defective.

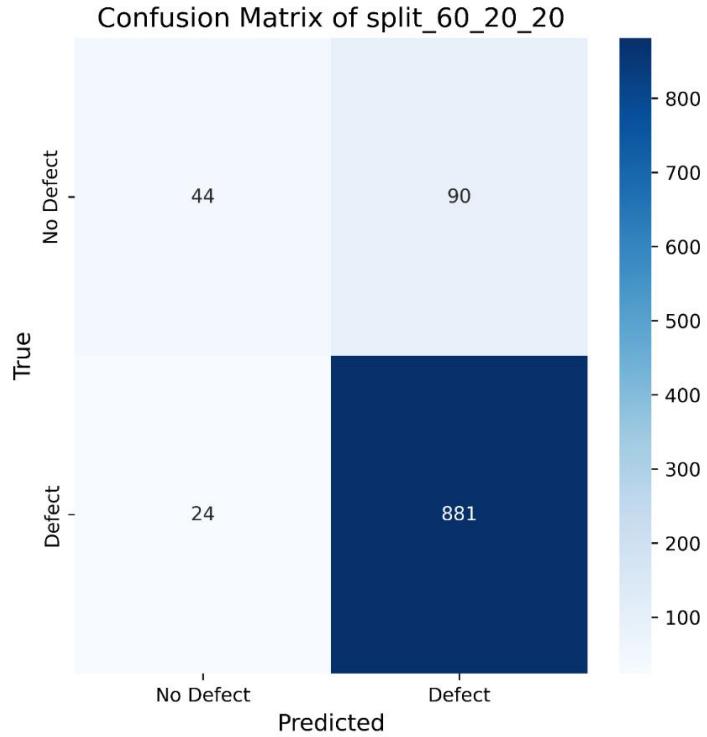


Figure 40. Confusion Matrix of split 60:20:20

The model performs well in predicting defective samples, correctly identifying 881 defective samples. However, it has more misclassifications when predicting non-defective samples, with 90 non-defective samples being incorrectly predicted as defective. Additionally, 24 defective samples were incorrectly predicted as non-defective.

4.3.6 Overall Comparison of three split ratio

Split Ratio	Accuracy	Loss	Precision	Recall/Sensitivity	AUC	Specificity	F1-Score
80:10:10	0.9142	0.0456	0.9142	0.9142	0.9684	0.3898	0.9540
70:20:10	0.9156	0.0464	0.9156	0.9156	0.9677	0.3518	0.9533
60:20:20	0.8870	0.0559	0.8870	0.8870	0.9544	0.3284	0.9392

Table 4. Overall Comparison of three split radio

In summary, the 70:20:10 split ratio performs the best across most evaluation metrics, particularly in accuracy and precision, slightly outperforming the other two ratios. The

80:10:10 ratio follows closely, maintaining a more balanced performance. The 60:20:20 split ratio shows a slight decrease in performance, especially in accuracy and F1 score.

4.4 Comparison of Four Models

In this part, there are comparisons of four models: (a) Proposed model (Inception + Attention + BiLSTM), (b) Inception model, (c) Inception + Attention model, (d) Basic CNN model

4.4.1 Accuracy comparison of four models

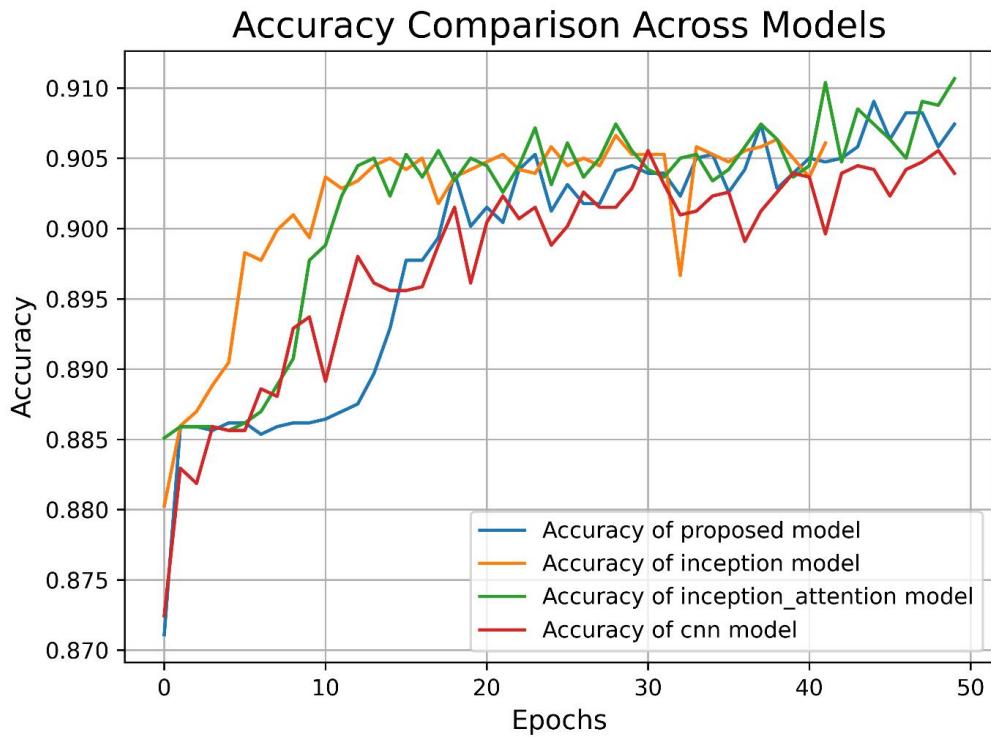


Figure 41. Accuracy comparison of four models

The proposed model consistently demonstrates superior performance, achieving the highest final accuracy of over 0.91 with stable convergence. The inception + attention model also shows strong performance, closely following the proposed model in terms of accuracy and stability. While the inception model exhibits rapid early improvement, it suffers from noticeable fluctuations in later epochs, suggesting possible overfitting. The baseline CNN model lags behind all others, with lower and more volatile accuracy throughout the training process. Overall, the proposed model outperforms the others in both accuracy and robustness, indicating its effectiveness for the given classification task.

4.4.2 Loss comparison of four models

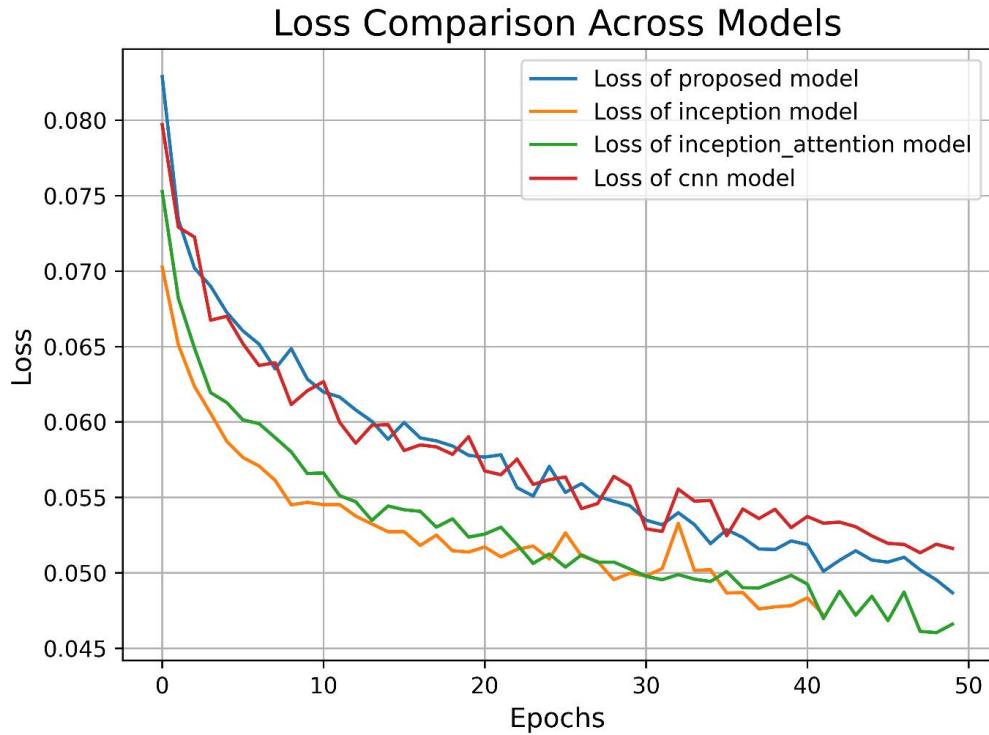


Figure 42. Loss comparison of four models

All models show a general downward trend in loss, indicating effective learning and convergence. Among them, the Inception + attention model achieves the lowest final loss and demonstrates the most stable convergence, highlighting its superior fitting capacity. The Inception model also performs well, with relatively low loss despite some fluctuations. The proposed model exhibits a stable decrease in loss and reaches a comparable final loss value to the Inception model, suggesting strong generalization ability. In contrast, the CNN model shows higher and more unstable loss values, indicating weaker performance in this task. Overall, the Inception + attention model outperforms others in minimizing loss, reinforcing the positive impact of incorporating attention mechanisms in deep learning architectures.

4.4.3 AUC comparison of four models

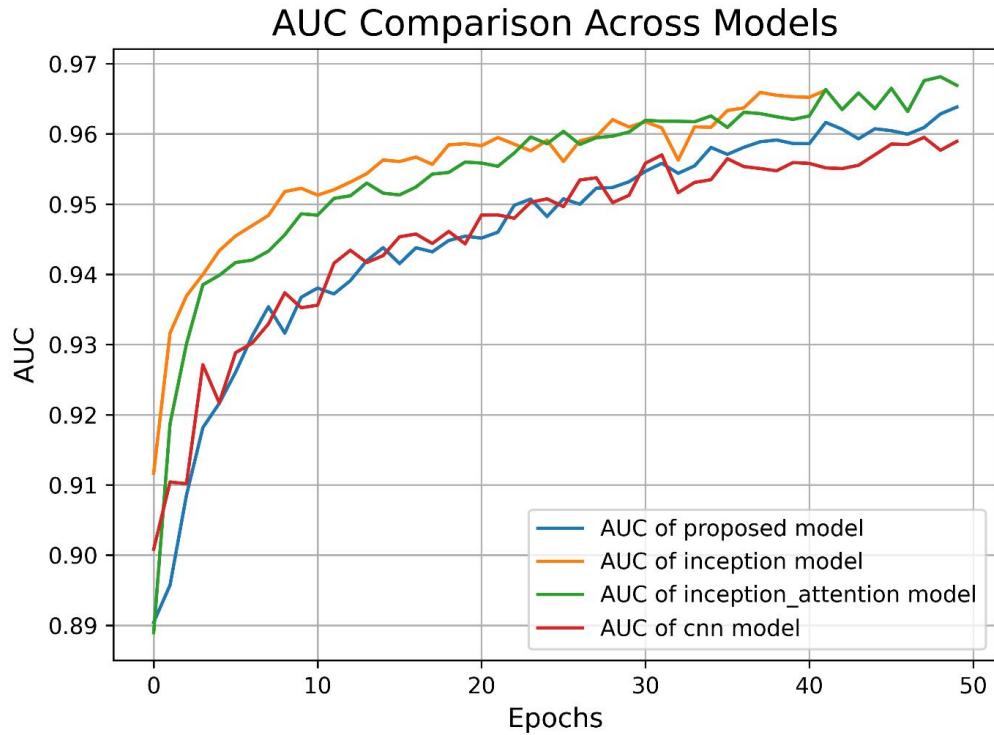


Figure 43. AUC comparison of four models

The Inception + attention model consistently achieves the highest AUC values, demonstrating strong learning and class separation abilities. The Inception model also performs excellently, with a rapid rise in AUC during early epochs and a stable plateau later, ranking second in final AUC. The proposed model starts with a relatively lower AUC but shows steady improvement, eventually reaching a comparable level with the Inception model, indicating robust learning capacity. In contrast, the CNN model lags behind, with lower AUC values and a slower increase, suggesting limited representational power.

4.4.4 F1 Score comparison of four models

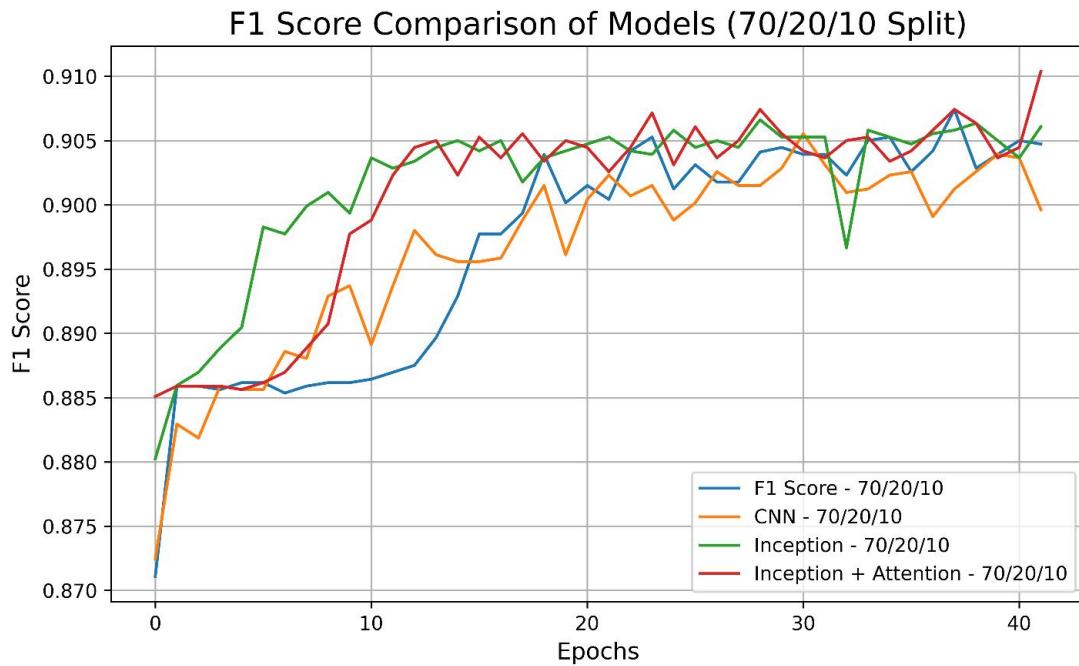


Figure 44. F1 Score comparison of four models

It's evident that all models exhibit a rapid increase in F1 score during the first 10 epochs, followed by a plateau. Among them, the Inception + Attention model consistently outperforms the others throughout the training process, achieving higher F1 scores with smaller fluctuations, indicating better stability and generalization ability. In contrast, while the traditional CNN model also improves rapidly in the early epochs, its overall F1 score is slightly lower and shows greater variance, suggesting it is less effective for this task. In conclusion, incorporating an Attention mechanism into the Inception network significantly enhances model performance in classification tasks.

4.4.5 Confusion Matrix comparison of four models

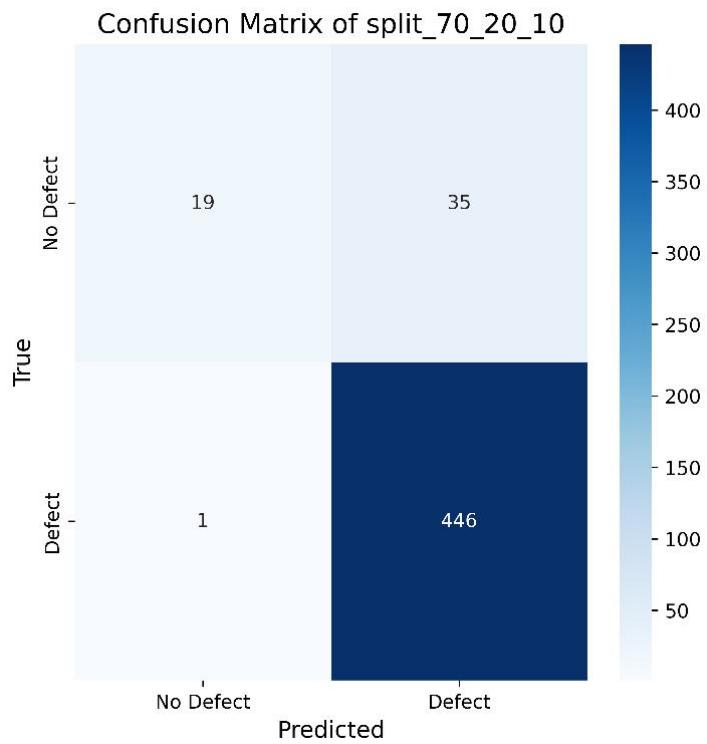


Figure 45. Confusion matrix of proposed model

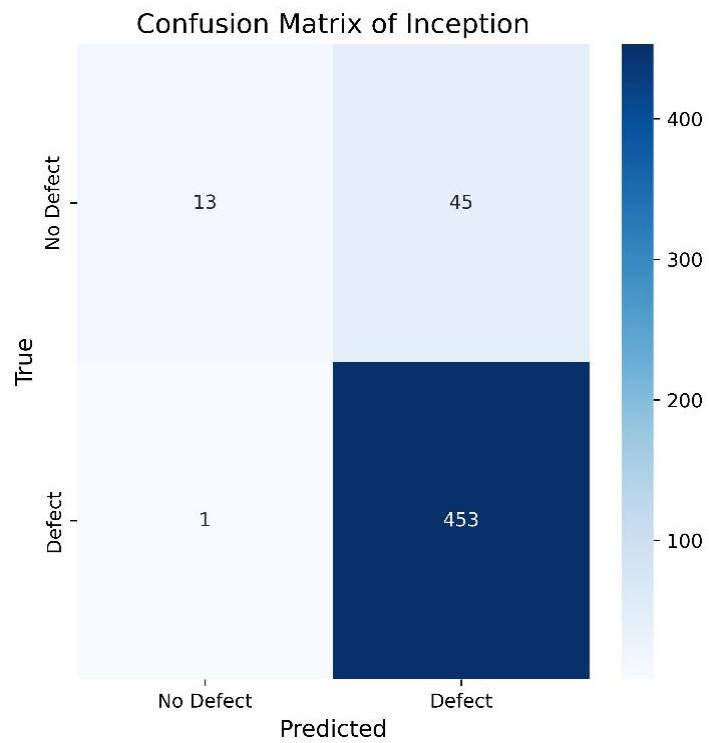


Figure 46. Confusion matrix of Inception model

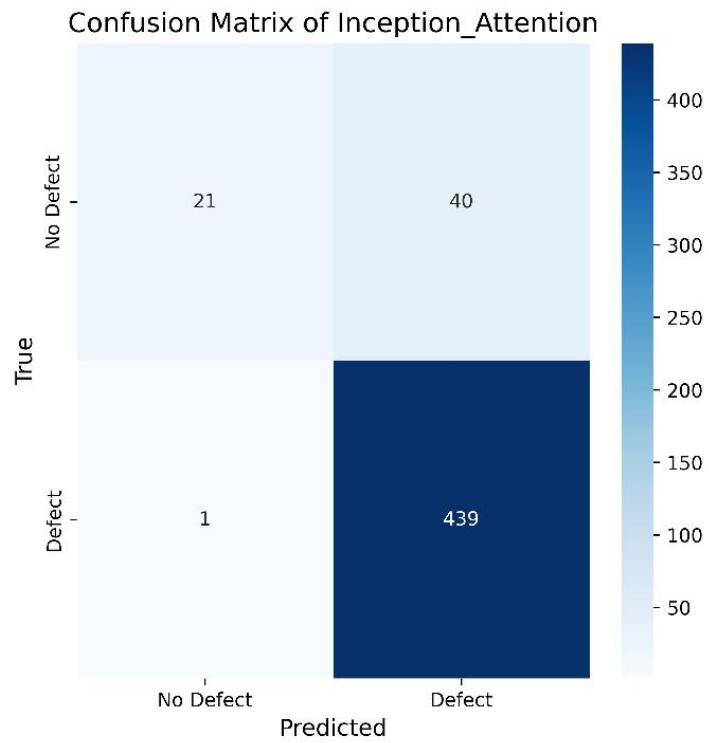


Figure 47. Confusion matrix of Inception + Attention model

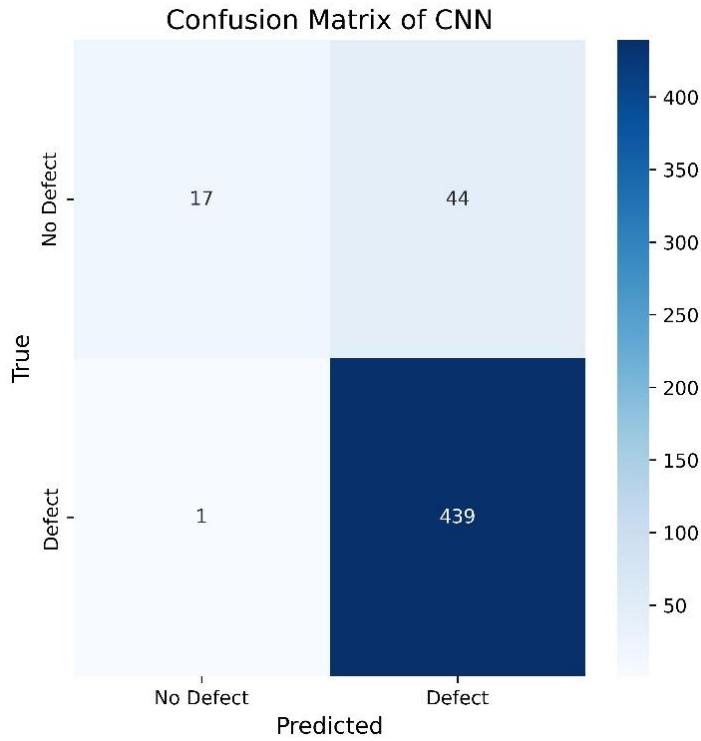


Figure 48. Confusion matrix of CNN model

From the confusion matrices in Figures 45 to 48, it is clear that all four models perform exceptionally well in identifying “Defect” samples, with only one false negative each—demonstrating excellent recall across the board. However, notable differences arise in classifying “No Defect” samples.

The proposed Inception + Attention + BiLSTM model (Figure 45) misclassified 35 non-defect instances, outperforming the Inception model (Figure 46, 45 misclassified) and the CNN model (Figure 48, 44 misclassified), and slightly better than the Inception + Attention model (Figure 47, 40 misclassified). These results indicate that incorporating BiLSTM improves the model’s ability to capture temporal dependencies, leading to higher accuracy in classifying non-defective samples while maintaining high recall.

Overall, the proposed model delivers the best balance between precision and recall among the four, showcasing superior robustness and greater potential for real-world defect detection tasks.

4.4.6 Overall Comparison of four models

Model	Accuracy	Loss	Precision	Recall/Sensitivity	AUC	Specificity	F1-Score
Proposed	0.9156	0.0464	0.9156	0.9156	0.9677	0.3519	0.9533
Inception	0.9043	0.0492	0.9043	0.9043	0.9651	0.2241	0.9550
Inception_Attention	0.9193	0.0460	0.9193	0.9193	0.9697	0.3442	0.9633
CNN	0.9118	0.0556	0.9118	0.9118	0.9565	0.2787	0.9581

Table 5. Overall Comparison of four models

As shown in the results table, the traditional CNN model performs the weakest across all metrics, particularly in terms of AUC (0.9565) and specificity (0.2787), indicating limited capability in distinguishing negative samples under complex conditions. The Inception model shows improved overall performance compared to CNN, with accuracy rising to 90.43% and AUC reaching 0.9651. However, its specificity remains low at 0.2241, suggesting poor recognition of negative cases. The Inception + Attention model, which incorporates attention mechanisms, achieves the best results so far in accuracy (91.93%) and AUC (0.9697), with improvements also seen in the F1-score, demonstrating that attention enhances discriminative ability. Nevertheless, its specificity (0.3442) shows only modest improvement, indicating further enhancement is needed for detecting negative instances.

The Proposed model, which integrates Inception, attention mechanisms, and a BiLSTM module, takes a significant step forward by combining spatial feature extraction with temporal or contextual learning. While maintaining high precision and recall, it also boosts overall discriminative power (AUC = 0.9677) and the ability to recognize negative samples (specificity = 0.3519). Compared to using only Inception or attention mechanisms, the addition of BiLSTM enhances the model's capacity to capture temporal or spatial contextual information, thereby improving its overall classification performance.

This makes the Proposed model the most balanced and robust across multiple metrics, especially well-suited for applications requiring high precision in anomaly detection.

4.5 Model Visualization

The goal of Explainable AI is to make AI systems more comprehensible and transparent without sacrificing performance [21]. In addition, XAI is to enhance model transparency and interpretability, allowing non-experts to understand the model's behavior, increase user trust in AI system decisions, facilitate the diagnosis of biases or issues in the model, and ensure compliance with regulatory requirements.

4.5.1 Grad-CAM

The code loads and normalizes the images through a preprocessing function. It then uses the output of one of the convolutional layers of the model along with the predicted gradients to generate a weighted feature map, resulting in a heatmap. Finally, the code visualizes the original image alongside the generated Grad-CAM heatmap, helping to understand the model's decision-making process. This allows us to observe the areas that the model focuses on when classifying a specific image, thereby enhancing the model's

credibility.

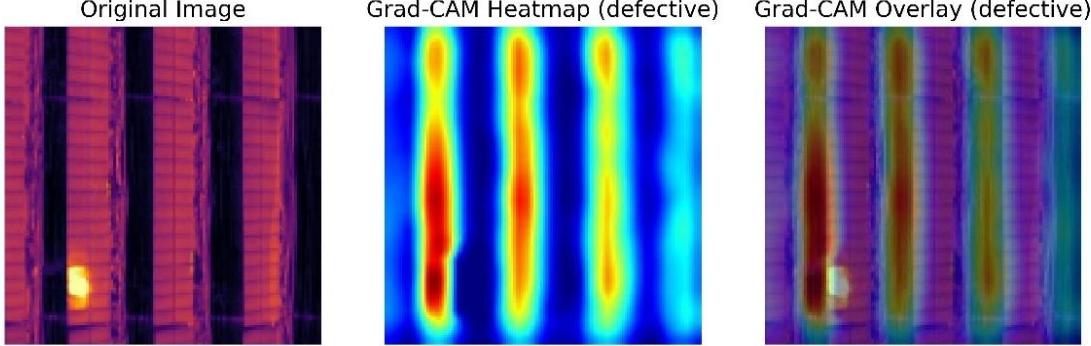


Figure 49. Grad-CAM image example

4.6 Model Deployment

The UI is designed to show the proposed model. On the home page, users can look through the topic of the website and find the links to different functions. The 'Detect Defects' components can navigate users to the detection page. In addition, when the users scrolling down, there is the dataset part which show the introduction of the dataset

and the ‘dataset website’ button can navigate the users to the source website of the dataset. In figure 50 to figure 56 are the instruction of the designed website.



Figure 50. Homepage 1 of the Webpage

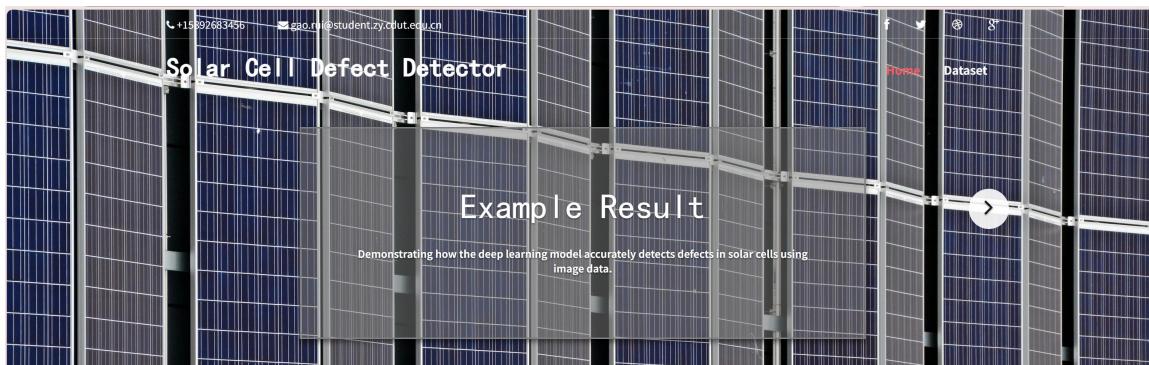


Figure 51. Homepage 2 of the Webpage

Figure 50 and 51 show the homepage of the website which include the introduction of the website.

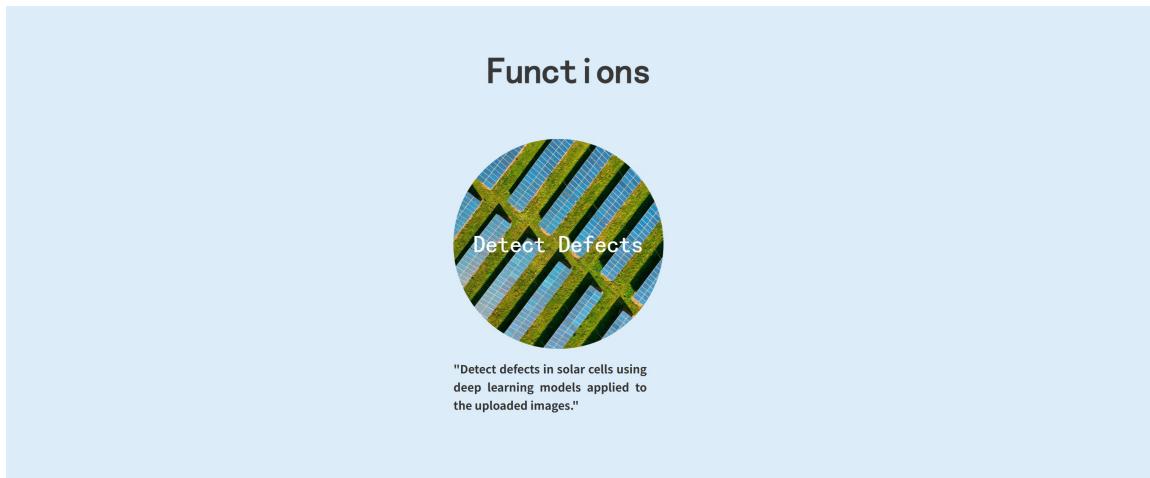


Figure 52. Functions Component of the Webpage



Figure 53. Detection Component of the Webpage

Figure 52 and 53 show the detection function of the website, when the users click the 'Detect Defects' button in figure 52, the website will redirect to the detection page shown in the figure 53. When the users click the 'Choose Image File' button, it can choose an image used in the function and then click the 'Detect Defects' button, it will show the prediction and the Grad-Cam images.

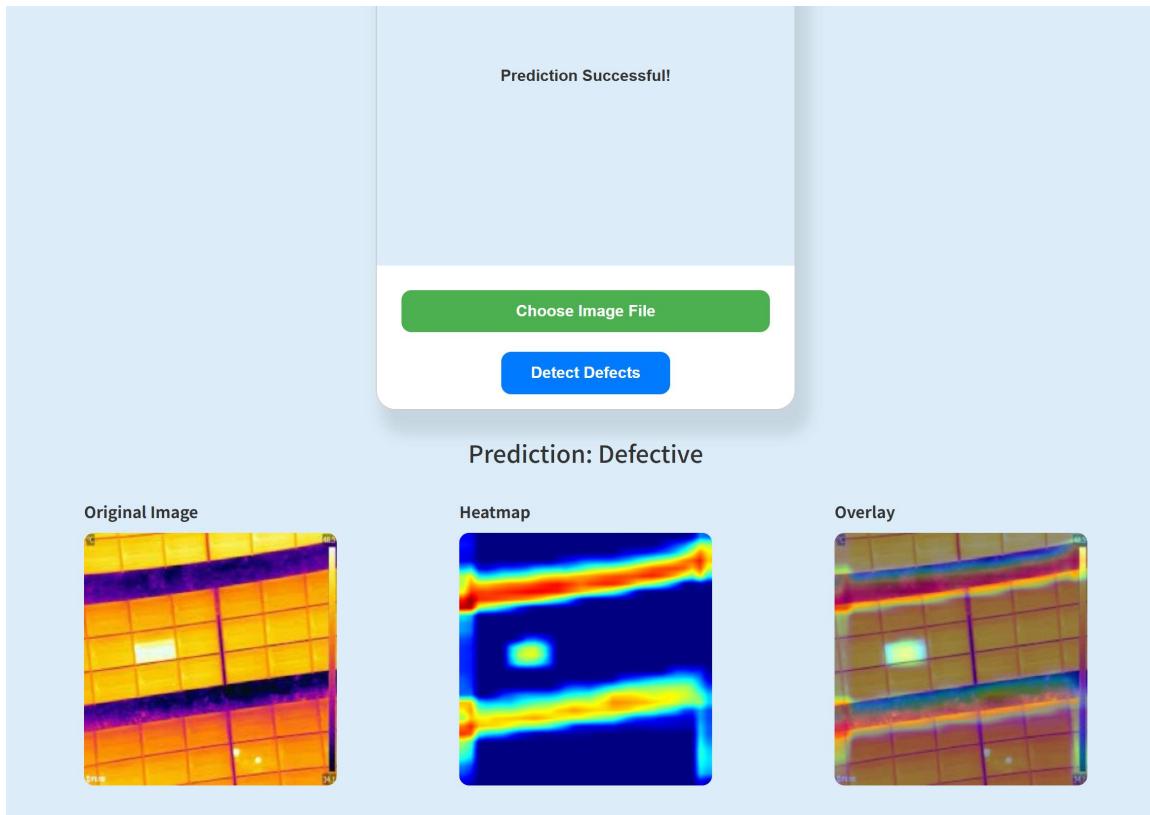


Figure 54. Result of Prediction

As the result, figure 54 shows the prediction result and the Grad-Cam of the image.

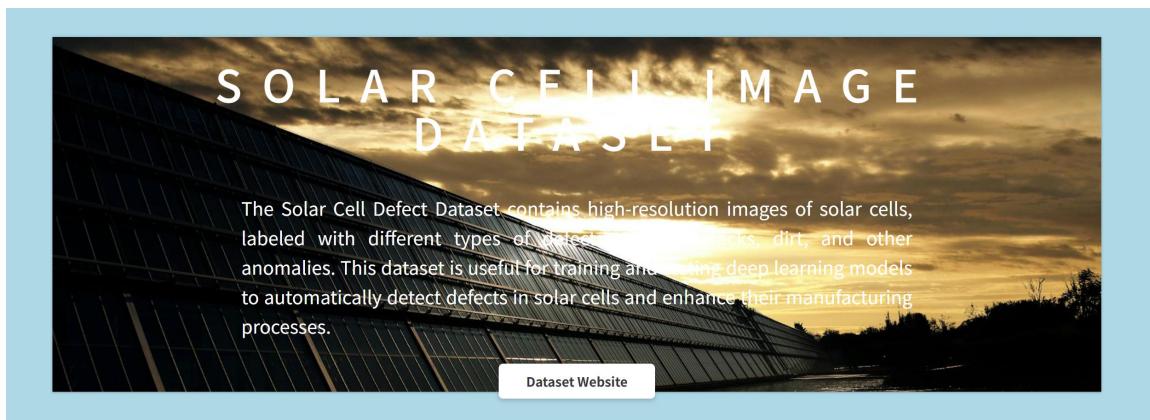


Figure 55. Dataset Component of the Website

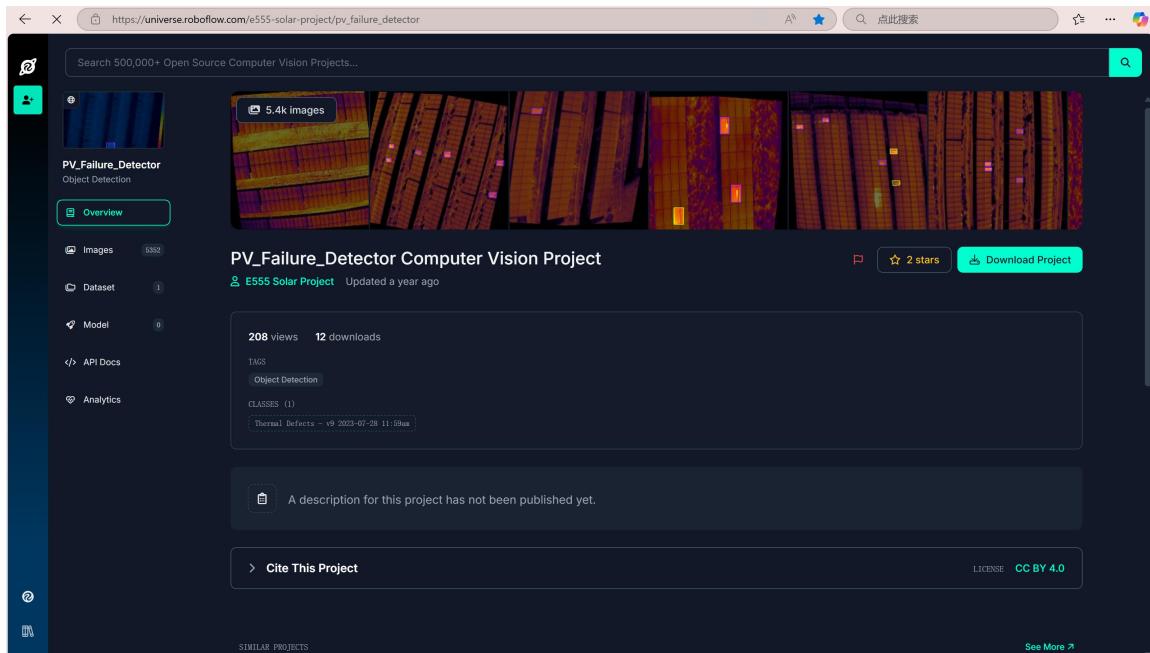


Figure 56. Dataset Website of the Project

Figure 55 and 56 show the dataset part of the website, it includes the dataset introduction and when the users click the ‘Dataset Website’, it will navigate to the dataset website shown in figure 56.

Chapter 5 Professional Issues

5.1 Project Management

5.1.1 Activities

This table shows the activities of the project.

Phase	Activities
1. Understand the aim of the project and have a brief idea about the project.	<ol style="list-style-type: none">1. Read the title of the project and find the keywords of the project.2. Search the keywords in the internet to learn about the background about the topic.3. Search the deep learning models and learn the basic knowledge about the models
2. Research the existing researches and the models they used.	<ol style="list-style-type: none">1. Search the keywords in the internet to find 10 existing works.2. Research the existing works and find the models they used.3. Analyze the result and accuracy of their models.
3. Collect the dataset of thermal imaging data of solar panels and preprocess the data.	<ol style="list-style-type: none">1. Find the dataset which is suitable for the topic on the website.2. View the dataset and check its usability.3. Preprocess the data of the thermal images of solar panels.
4. Select and design the model for analyzing thermal images.	<ol style="list-style-type: none">1. Find and select the suitable models for the topic.2. Learn more about the deep knowledge of CNN models include the basic concepts, components, working principles, advantages, applications,

	<p>etc.</p> <p>3. Design the deep learning model including inception, attention and BiLSTM.</p>
5. Train and test the model with the dataset and optimize it.	<p>1. Train the model with the dataset and find the result of the accuracy and the loss.</p> <p>2. Analyze the result and summarize the advantages and limitations</p> <p>3. Optimize the model result.</p>
6. Model Comparison	<p>1. Choose the suitable model to compare the data performance.</p> <p>2. Compare the three different split ratios of the dataset.</p> <p>3. Design the GUI to virtualize the model.</p>
7. Present the result of the project.	<p>1. Conclude all the contribution and result.</p> <p>2. Prepare the final report and poster.</p> <p>3. Complete the PPT for presentation.</p>

Table 6. Activities of the project

5.1.2 Schedule

NB: 1.1 represent Objective 1, Activity 1.

The schedule is shown in figure 57.

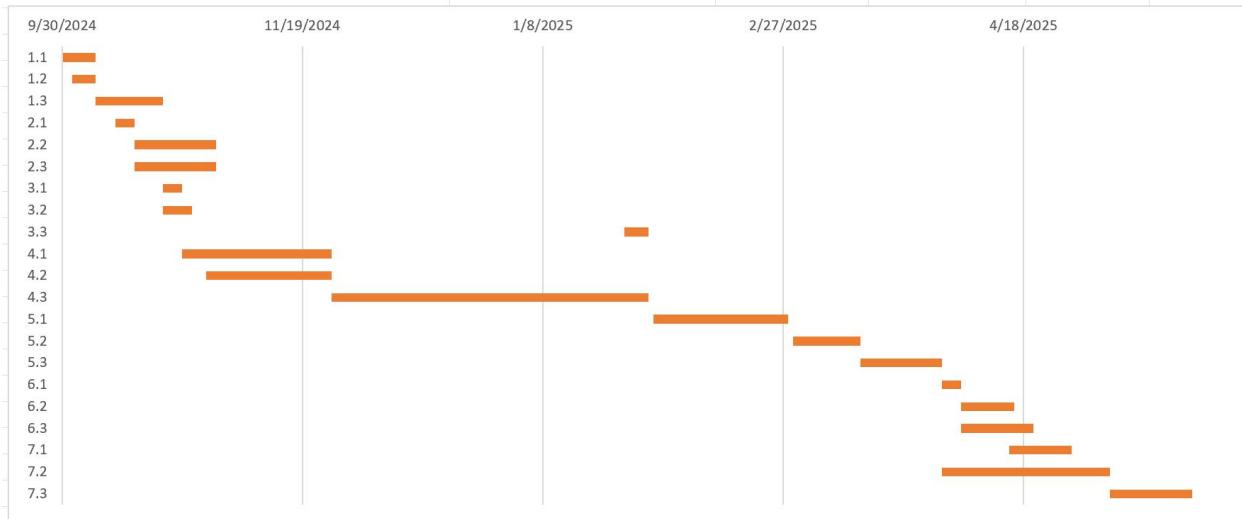


Figure 57. Gantt Chart of the Project

5.1.3 Project Data Management

To manage the different versions of codes modification, I plan to use Github as the version management tools.

URL: <https://github.com/Olivia-Gao/Deep-Learning-Project>

Version Number	Code Name	Contents	Results
1	Proposed model code	The proposed model code	The model result for classification with all measured metrics.
2	Comparison model code	Inception model, inception_attention model, basic CNN model	The model result for classification with all measured metrics.
3	Model deployment	The web-application program where the model can be added. Model .h5 file	Webui and model .h5 files

Table 7. Project Data Management

5.1.4 Project Data Management

All files will be replicated into three copies for fail safe, one on local computer, one on hard drive, one on GitHub.

Following are documents of the Project for uploading and synchronization:

- Reports (Weekly, Proposal, Progress, Final) & Presentation PPT
- Model architecture diagrams
- References
- Datasets Link
- Model evaluation documents
- Model codes (Different versions)

5.1.5 Project Deliverables

1. Code
2. Datasets
3. Ethics for Project
4. Final Report
5. Presentation
6. Progress Report
7. Proposal
8. Reference
9. Testing
10. Weekly Report

5.2 Risk Analysis

This table shows the risks in the project.

Risk ID	Potential Risk	Cause ID	Potential Causes	Severity	Likelihood	Risk Value	Mitigation ID	Mitigation
R1.1	Loss of Project Data	C1.1.1	Poor version control	4	1	4	M1.1.1	Regularly update project to cloud

		C1.1. 2	Hardware failure	4	1	4	M1.1.2	Perform hardware check & backup regularly
R1. 2	Memory Leakage	C1.2. 1	Model training exceeds hardware ability	4	3	12	M1.2.1	Use cloud services for training with larger datasets
R1. 3	Model Training Issues	C1.3. 1	Data imbalance	2	4	8	M1.3.1	Use data augmentation & upsampling techniques
		C1.3. 2	Low quality data	4	1	4	M1.3.2	Find trusted sources for high-quality datasets (e.g., Kaggle)
R1. 4	Software Issues	C1.4. 1	Virtual environment error	4	1	4	M1.4.1	Keep the virtual environment clean and up-to-date
R2.	Missed	C2.1.	Illness	3	1	3	M2.1.1	Maintain a

1	Deadlines	1						healthy work-life balance
		C2.1.2	Poor time management	4	1	4	M2.1.2	Strictly follow a project schedule and set realistic goals

Table 8. Risk Analysis

5.3 Professional Issues

1. Legal Issues: Ensure compliance with data privacy laws (e.g., GDPR) when using thermal images, obtain consent for data usage, and adhere to intellectual property rights. Consider liability concerns if the model's predictions lead to incorrect maintenance decisions that cause damage to infrastructure.
2. Social Issues: Automation of maintenance tasks may lead to job displacement for workers traditionally involved in inspections. The accessibility of advanced technology should be considered to prevent widening the digital divide, ensuring smaller companies and developing regions can benefit from it.
3. Ethical Issues: Address biases in deep learning models to ensure fairness and accuracy. Ensure transparency and accountability in predictions, and provide explainability of AI decisions. Obtain informed consent from solar plant owners, ensuring they are aware of data usage and analysis.
4. Environmental Issues: The project promotes sustainability by improving solar panel efficiency. However, the energy consumption of deep learning model training and potential electronic waste from AI devices should be considered, ensuring responsible disposal and reducing the environmental footprint of the technology.
5. Professional Code of Conduct: Follow the BCS and ACM codes, focusing on integrity, transparency, privacy, and public welfare. The project must ensure ethical use of AI, safeguard against misuse, and prioritize the welfare of individuals and communities affected by the technology's deployment.

Chapter 6 Conclusion

This research has successfully developed a deep learning model for detecting solar panel defects through thermal infrared imaging. By combining Inception modules, attention mechanisms, and BiLSTM, the model effectively addresses both spatial and temporal feature extraction, which are crucial for identifying the defects of solar cells. The integration of a focal loss function helps overcome class imbalance issues, ensuring the model is capable of learning from difficult examples. The experimental results demonstrate that the model outperforms some models like Inception model, Inception + Attention model, and basic CNN model, achieving an accuracy of 91.56%, an AUC of 0.9677, and an F1 score of 95.33%. These results validate the potential of combining advanced deep learning techniques for improved defect detection in solar panels. Furthermore, the use of Grad-CAM for visualizing model decisions enhances its interpretability, providing insights into the areas the model focuses on during classification.

Despite these promising outcomes, there are some limitations to consider. The dataset used lacks diversity and is relatively small, which may affect the model's performance when deployed in real-world scenarios. Additionally, the model's dependency on labeled data for training poses a challenge, especially in large-scale applications where manual annotation is costly and time-consuming. The current model is also computationally expensive, making it less suitable for real-time or large-scale deployment without further optimization. Moreover, the system has not yet been tested in real-time field conditions, such as during drone-based inspections, which limits the understanding of its practical viability.

In conclusion, this work provides an effective and automated solution for solar panel defect detection, which could significantly improve maintenance efficiency and reduce costs. The integration of explainable AI techniques and practical deployment mechanisms further contributes to the model's relevance in real-world applications. As renewable energy systems continue to grow, this research holds substantial potential to support the smart operation and maintenance of solar energy infrastructure.

References

- [1] J. Pastuszak and P. Węgierek, "Photovoltaic Cell Generations and Current Research Directions for Their Development," *Materials*, vol. 15, no. 16, p. 5542, Aug. 2022, doi: 10.3390/ma15165542.
- [2] A. Shaik, A. Balasundaram, L. S. Kakarla, and N. Murugan, "Deep Learning-Based Detection and Segmentation of Damage in Solar Panels," *Automation*, vol. 5, no. 2, pp. 128–150, May 2024, doi: 10.3390/automation5020009.
- [3] J. C. Sánchez Barroso, N. Barth, J. P. M. Correia, S. Ahzi, and M. A. Khaleel, "A computational analysis of coupled thermal and electrical behavior of PV panels," *Sol. Energy Mater. Sol. Cells*, vol. 148, pp. 73–86, Apr. 2016, doi: 10.1016/j.solmat.2015.09.004.
- [4] "Photovoltaic Cell: Definition, Construction, Working & Applications," GeeksforGeeks, Mar. 04, 2024. <https://www.geeksforgeeks.org/photovoltaic-cell/>
- [5] A. Mohammad Bagher, "Types of Solar Cells and Application," *Am. J. Opt. Photonics*, vol. 3, no. 5, p. 94, 2015, doi: 10.11648/j.ajop.20150305.17.
- [6] M. Zhang and L. Yin, "Solar Cell Surface Defect Detection Based on Improved YOLO v5," *IEEE Access*, vol. 10, pp. 80804–80815, 2022, doi: 10.1109/ACCESS.2022.3195901.
- [7] A. N. Wilson, K. A. Gupta, B. H. Koduru, A. Kumar, A. Jha, and L. R. Cenkeramaddi, "Recent Advances in Thermal Imaging and its Applications Using Machine Learning: A Review," *IEEE Sens. J.*, vol. 23, no. 4, pp. 3395–3407, Feb. 2023, doi: 10.1109/JSEN.2023.3234335.
- [8] L. Alzubaidi *et al.*, "Review of deep learning: concepts, CNN architectures, challenges, applications, future directions," *J. Big Data*, vol. 8, no. 1, p. 53, Mar. 2021, doi: 10.1186/s40537-021-00444-8.
- [9] S. Indolia, A. K. Goswami, S. P. Mishra, and P. Asopa, "Conceptual Understanding of Convolutional Neural Network- A Deep Learning Approach," *Procedia Comput. Sci.*, vol. 132, pp. 679–688, 2018, doi: 10.1016/j.procs.2018.05.069.

- [10] A. Ghosh, A. Sufian, F. Sultana, A. Chakrabarti, and D. De, “Fundamental Concepts of Convolutional Neural Network,” in *Recent Trends and Advances in Artificial Intelligence and Internet of Things*, vol. 172, V. E. Balas, R. Kumar, and R. Srivastava, Eds., in Intelligent Systems Reference Library, vol. 172. , Cham: Springer International Publishing, 2020, pp. 519–567. doi: 10.1007/978-3-030-32644-9_36.
- [11] A. Younesi, M. Ansari, M. Fazli, A. Ejlali, M. Shafique, and J. Henkel, “A Comprehensive Survey of Convolutions in Deep Learning: Applications, Challenges, and Future Trends,” *IEEE Access*, vol. 12, pp. 41180–41218, 2024, doi: 10.1109/ACCESS.2024.3376441.
- [12] L. Koshy, M. V. Vaishnav, S. Sunil, S. V. Abraham, and S. Vidhyadharan, “A Thermal Image-based Fault Detection System for Solar Panels,” in *2024 International Conference on Cybernation and Computation (CYBERCOM)*, Dehradun, India: IEEE, Nov. 2024, pp. 246–250. doi: 10.1109/CYBERCOM63683.2024.10803145.
- [13] F. Wang, Z. Wang, Z. Chen, D. Zhu, X. Gong, and W. Cong, “An Edge-Guided Deep Learning Solar Panel Hotspot Thermal Image Segmentation Algorithm,” *Appl. Sci.*, vol. 13, no. 19, p. 11031, Oct. 2023, doi: 10.3390/app131911031.
- [14] P. Haidari, A. Hajiahmad, A. Jafari, and A. Nasiri, “Deep learning-based model for fault classification in solar modules using infrared images,” *Sustain. Energy Technol. Assess.*, vol. 52, p. 102110, Aug. 2022, doi: 10.1016/j.seta.2022.102110.
- [15] C. Dunderdale, W. Brettenny, C. Clohessy, and E. E. Van Dyk, “Photovoltaic defect classification through thermal infrared imaging using a machine learning approach,” *Prog. Photovolt. Res. Appl.*, vol. 28, no. 3, pp. 177–188, Mar. 2020, doi: 10.1002/pip.3191.
- [16] H. Yousif and Z. Al-Milaji, “Fault detection from PV images using hybrid deep learning model,” *Sol. Energy*, vol. 267, p. 112207, Jan. 2024, doi: 10.1016/j.solener.2023.112207.
- [17] S. Jaybhaye, O. Thakur, R. Yardi, V. Raut, and A. Raut, “Solar Panel Damage Detection and Localization of Thermal Images,” *J. Fail. Anal. Prev.*, vol. 23, no. 5, pp. 1980–1990, Oct. 2023, doi: 10.1007/s11668-023-01747-z.

- [18] T. Barakbayeva and F. M. Demirci, “Fully automatic CNN design with inception and ResNet blocks,” *Neural Comput. Appl.*, vol. 35, no. 2, pp. 1569–1580, Jan. 2023, doi: 10.1007/s00521-022-07700-9.
- [19] G. Brauwers and F. Frasincar, “A General Survey on Attention Mechanisms in Deep Learning,” *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 4, pp. 3279–3298, Apr. 2023, doi: 10.1109/TKDE.2021.3126456.
- [20] M. Kerkhof, L. Wu, G. Perin, and S. Picek, “Focus is Key to Success: A Focal Loss Function for Deep Learning-Based Side-Channel Analysis,” in *Constructive Side-Channel Analysis and Secure Design*, vol. 13211, J. Balasch and C. O’Flynn, Eds., in Lecture Notes in Computer Science, vol. 13211. , Cham: Springer International Publishing, 2022, pp. 29–48. doi: 10.1007/978-3-030-99766-3_2.
- [21] S. Ali *et al.*, “Explainable Artificial Intelligence (XAI): What we know and what is left to attain Trustworthy Artificial Intelligence,” *Inf. Fusion*, vol. 99, p. 101805, Nov. 2023, doi: 10.1016/j.inffus.2023.101805.

Appendices

The project is on Github, URL is as follow:

<https://github.com/Olivia-Gao/Deep-Learning-Project>

The dataset link:

https://universe.roboflow.com/e555-solar-project/pv_failure_detector