# Autonomous Fault Diagnosis of Commercially Available PV Modules Using High-End Deep Learning Frameworks

By

Ihtyaz Kader Tasawar
17321038
Abyaz Kader Tanzeem
17321039
Tahmid Ahmed
17121095
Shah Faiza Zarin
17121037

A thesis submitted to the Department of Electrical and Electronic Engineering in partial fulfillment of the requirements for the degree of
B.Sc. in EEE

Department of Electrical and Electronic Engineering
Brac University
June 2021

# Declaration

It is hereby declared that

1. The thesis submitted is my/our own original work while completing degree at Brac University.
2. The thesis does not contain material previously published or written by a third party, except where this is appropriately cited through full and accurate referencing.
3. The thesis does not contain material which has been accepted, or submitted, for any other degree or diploma at a university or other institution.
4. I/We have acknowledged all main sources of help.

**Student's Full Name & Signature:**

| | |
|---|---|
| **Ihtyaz Kader Tasawar**<br>17321038 | **Abyaz Kader Tanzeem**<br>17321039 |
| **Tahmid Ahmed**<br>17121095 | **Shah Faiza Zarin**<br>17121037 |

# Approval

The thesis/project titled “Autonomous Fault Diagnosis of Commercially Available PV Modules Using High-End Deep Learning Frameworks” submitted by

1. Ihtyaz Kader Tasawar (17321038)
2. Abyaz Kader Tanzeem (17321039)
3. Tahmid Ahmed (17121095)
4. Shah Faiza Zarin (17121037)

of Spring, 2021 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of B.Sc. on June 9. 2021.

**Examining Committee:**

Supervisor:
(Member)

_____
Md. Mosaddequr Rahman, PhD
Professor and Chairperson, Department of Electrical and Electronic Engineering
Brac University

Program Coordinator:
(Member)

_____
Abu S.M. Mohsin, PhD
Assistant Professor, Department of Electrical and Electronic Engineering
Brac University

Departmental Head:
(Chair)

_____
Md. Mosaddequr Rahman, PhD
Professor and Chairperson, Department of Electrical and Electronic Engineering
Brac University

## Ethics Statement

## Abstract/ Executive Summary

Conventional methods of fault diagnosis for PV Systems are quite challenging and inefficient, particularly with regards to large-scale PV arrays. Early and effective diagnosis of system faults is also imperative in order to minimize cost and sustainable damage. Hence, over the recent years, numerous effective and efficient monitoring and diagnostic techniques to detect faults in PV systems have been studied and propositioned. As such, autonomous fault diagnosis and classification of PV systems has taken the PV domain by storm and has spectacularly developed in eminence; attaining substantial significance in the domain of deep learning. Over the last few years, various deep learning frameworks have been studied and proposed in the detection & classification of faults in PV modules with the aid of thermal images. Some of the most prominent deep learning frameworks constitutes of ANN & CNN. This study involves utilization of Convolutional Neural Networks (CNN), namely, VGG-16/VGG-19 and EfficientNet, in order to assess their performance and reliability in diagnosing module defects through significant hotpots within PV modules by employing pre-processed thermal images.

**Keywords:** Deep Neural Network; Convolutional Neural Network; Infrared Image Processing; Photovoltaic Cell; Fault Diagnosis; Hotspot Detection

## Dedication (Optional)

We would like to devote our work to the sheer effort and untarnished commitment of our parents, who have endured and endeavored tirelessly in order to ensure the development of our academic status.

## Acknowledgement

We would like to impart our most profound gratitude and indebtedness towards our Creator Allah (S.W.T), for endowing us with the capabilities and provisions to come this far and making us so privileged.

In addition, we would like extend our sincerest and heartfelt gratitude towards our thesis supervisor Professor Dr. Md. Mosaddequr Rahman Sir for his continuous and valuable support, guidance and untarnished commitment. Without his consistent counsel, we wouldn't have been able to make it this far in our work. Moreover, we would also like to express our most sincere gratitude and appreciation towards Mr. Mohaimenul Islam for his profound dedication and commitment in ensuring the consistency and validity of our work.

# Table of Contents

# List of Tables

## List of Figures

## List of Acronyms

| | |
|---|---|
| IR | Infrared |
| CNN | Convolutional Neural Network |
| RL | Rectified Linear Unit |
| FC | Fully Connected |
| VGG | Visual Geometry Group |
| RGB | Red Green Blue |
| FLOPS | Floating Point Operations Per Second |
| PV | Photovoltaic |

# Chapter 1

# Introduction

## 1.1 Work Significance – Autonomous Fault Diagnosis

Over the course of a few decades, there has been a substantial surge in concern with regards to the large-scale consumption and reliability of prevalent energy sources such as fossil fuels and coal which prominently contribute to different types of pollution and global warming. With the rising demand for electrical energy due to the surge in population and economic growth, it has become imperative to address and confront this demand through exploitation of climate-friendly energy sources. On account of that, electrical energy production has been tailored and revolutionized to adhere to renewable and climate-friendly energy sources such as solar, water and wind [1]. Among which, exploitation of solar energy through utilization of Photovoltaic Systems has become increasingly prominent in recent years as demonstrated in [2]. Bangladesh is one of many countries who have inclined towards PV systems [3] for a more reliable, economic and ecological solution for electricity generation. This surge in prominence has been primarily due to the many merits of photovoltaic (PV) systems, such as generation of clean and non-polluting energy, abundant source, convenient installation & system maintenance, noise-free, and cost-effective.

With increasing prominence in utilization of PV systems, enhanced power efficiency and performance of the systems has been emphasized and sought after extensively over the years [4]. The performance and efficiency of a PV system may be subjected to constraints and degradation by elements such as aging, different system faults or defects, and environmental conditions. Some of the most common faults include shading conditions, cracks & dents, open-circuit faults, and faults due to wear and tear from the environment. The development of such defects induces an expansive dissipation of power in the defective area of the module and thus

creating a region of hotspot. Consequently, the output power of the module shrinks and reduces the efficiency of the module. As such, ensuring optimum performance and efficiency through effective detection and diagnosis of prominent PV system faults, that can potentially degrade the system's output power and lifespan, has become the primary focal point over the recent years [5]. Generally, faults are diagnosed through manual inspection and rectified through maintenance accordingly. However, employment of such labor is very much time consuming, challenging, inefficient and unsafe with regards to extensive PV arrays. Needless to say, proper and prompt diagnosis of these faults is also imperative to minimize cost and sustainable damage.

## 1.2 Efficient Fault Detection in PV Systems

In the recent years, numerous effective and efficient monitoring and diagnostic techniques to detect faults in PV systems have been studied and propositioned [6, 7, 8, 9, and 10]. Essentially, the basis of many fault detection techniques involves inspecting the heat signatures or hotspots of the PV modules caused by defects that could potentially hamper the normal operation of the modules. Gradually, autonomous fault detection and classification of PV systems has taken the PV domain by storm and has spectacularly developed in eminence; attaining substantial significance in the domain of deep learning networks. Typically, image-processing based techniques such as Infrared Thermography (IRT) [12] and Electroluminescence (ELI) Imaging [11] are employed concurrently with deep learning networks for automatic fault detection and classification of PV modules. While EL Imaging can be an effective aspect in autonomous fault detection, it can be inefficient in detection and diagnosis of faults such as discoloration and delamination. Hence, IRT takes precedence over ELI due to its ability to manifest hotspots accurately. Other autonomous diagnostic techniques include utilization of machine learning models such as Fuzzy Logic, Decision Tree and SVM models. In such prominent techniques, the potency heavily relies on the amount of data you can dispense for the model's training

phase. The greater the size of the dataset, better the chances of developing an efficient and accurate model.

## 1.3 Objective - Deep Learning in Autonomous Fault Diagnosis

Deep Learning Frameworks such as Artificial Neural Networks (ANN) & Convolutional Neural Networks (CNN) through utilization of Infrared Imaging have significantly grown in prominence in recent years. In this study, we have employed Infrared Imaging to investigate and assess the use of Convolutional Neural Networks frameworks, VGG-16, VGG-19 and EfficientNet in order to distinguish infrared images of faulty PV modules with prominent hotspots from that of normal modules. Additionally, our study also employs different image processing techniques in order to pre-process the infrared images and label them accordingly to train the neural networks and assess their performance.

## 1.4 Literature Review

As the faults of PV systems are being better understood over time, many different techniques have been developed and tested for the detection of PV system faults. Fault detection techniques on the DC side of PV systems are mainly based on electrical characterization, visual inspection, ultrasonic inspection, electroluminescence (EL) imaging, infrared (IR) imaging and lock in thermography (LIT) as outlined in [13]. Over the last couple of years, direct monitoring methods such as electroluminescence imaging and infrared thermography have garnered great interest due to the fact that they can both detect and locate PV module defects with high accuracy and shorter detection time in comparison to other methods [14]. More recently, the rapid advancement in the field of artificial intelligence and computer vision has opened up wide range of opportunities to develop a highly robust, fast and automatic fault detection technique. The integration of machine learning techniques and image processing methods with IR imaging

for automatic PV fault detection and diagnosis has occupied much of the space within the literature.

By standards, PV modules are first visually inspected for visible faults before any further techniques of fault detection are carried out on them. The practical guidelines and instructions for effectively carrying out the process of visual inspection of mono-polar grid-connected and grounded PV systems have been outlined in [15, p. 13]. While on the subject of PV fault detection techniques, Electroluminescence (EL) imaging is a non-destructive testing method that involves the introduction of a DC current into the PV module, so that areas of the cell or module with defects appear darker than rest of the areas on electroluminescence images [13], [16]. Spataru et.al [17] investigated cell crack defects by analyzing EL intensity distribution of PV systems, both on module and cell level, operated under high and low current bias. They were able to evaluate module-level percentage of cell cracks in terms of partial and complete defects that will cause the most power loss. Furthermore, an automatic method for the detection of solar cell defects has been studied in [18] using electroluminescence imaging, VGG-16 convolutional neural network architecture and image preprocessing techniques such as distortion correction, segmentation and perspective correction. However, due to limited dataset and overfitting, the CNN architecture failed to correctly detect defects with a balanced error rate of 25.40% with oversampling and 19.57% with data augmentation. The balanced error rate improved down to 7.73% when both oversampling and data augmentation were used simultaneously.

Even though EL imaging can produce high quality images for analysis and can detect and locate module defects with high accuracy, it is only effective for small-scale indoor measurements. This is because EL image diagnosis can be time consuming and costly due to the fact that it needs a very large power supply for the inspection of large-scale PV plants. EL imaging also needs to be carried out in a dark environment where the modules do not operate at maximum

power-point (MPP) conditions and does not provide any information regarding the thermal effects of possible faults in the modules [14].

Alternatively, fault diagnosis using infrared thermography (IRT) is a non-contact and non-destructive technique that has been captivating great interest in the field of PV systems monitoring due to its potential in overcoming many of the limitations that electroluminescence imaging and other defect detection methods exhibits. Following are some of the reasons that make infrared thermography highly preferable over other methods for detecting and diagnosing PV module defects:

- It is applicable for large-scale inspection of outdoor PV systems having large number of arrays [14].
- It can be performed under real field conditions without interrupting the operation of the modules [19].
- It provides information about output power loss as demonstrated in [20].
- Requires very few components and does not need sensors.
- Infrared thermography is fast and accurate in detecting hotpots and locating the physical position of the fault.

IRT-based defect diagnosis provides information regarding the thermal response of existing defects in an operating PV module by comparing it with the thermal pattern of the whole module. The defects appear as hot spots on infrared images indicating regions that have abnormally higher temperatures than the rest of the module and contributes to significant power loss. Hence, it is of utmost importance that PV systems undergo hotspot inspections to avoid damages and infrared thermography is a fitting method to locate the hot spots and find the exact physical location of the existing fault.

In their study [21], Buerhop et.al analyzed the robustness of IR-imaging for detecting faults of PV-generators under operating conditions. The modules that were suspected to be faulty in IR images were verified by taking down all the modules and measuring their respective power losses via I-V characteristics measurement. Their results indicated that 19% of all the modules had a power output less than 91% of the nominal power and this was consistent with the conclusions that were drawn from IR imaging. They finally concluded that IR-imaging is an effective, reliable and contactless method for monitoring PV-plants under operating conditions as it can detect modules which demonstrate reduced efficiency with high accuracy. Tsanakas and Botsaris [22] investigated the reliability of IR imaging in assessing the performance of PV modules by comparing the operating temperatures measured from the thermal images with the model-based temperature estimates of the modules. Their results were promising since they were able to detect an unseen 2mm crack on one of the modules which was affecting its performance. They deemed infrared thermography to be effective and authentic but suggested that emissivity problems and environmental conditions should be considered.

For large-scale maintenance of PV plants that consists of several hundred modules, automating the condition monitoring process is essential for fast, reliable and consistent results. As computer vision and artificial intelligence transformed its way into industrial and medical applications, it has also been extended to PV systems fault detection and diagnosis providing many advantages.

As suggested in [23, 24], a conventional automatic diagnosis system based on infrared thermography involves three main steps:

- obtaining raw data by IRT monitoring method
- thermal image processing
- diagnosis/classification

Infrared image monitoring alone cannot constitute an efficient fault detection method since the IR images do not always lay out the proper information required for identifying the defects successfully. This is on account of low resolution of the images, noisy images, effect of climatic conditions and reflection or shading during the inspection process [25]. Therefore, additional analysis of the thermal images by digital image processing techniques is necessary to pick up detailed features and regions of interest (ROI). Digital image processing involves manipulating digital images by applying enhancement, restoration, analysis or compression. When dealing with large-scale PV plants, further analysis through digital image processing is crucial for thorough analysis of the condition, performance and efficiency of each module. There are a vast number of renowned image processing techniques which have been reported in the literature over the years and they are pretty straightforward and effective. In [26], image processing techniques such as thermal contrast computation, differential absolute contrast (DAC), Pulsed phase thermography (PPT), thermographic signal reconstruction (TSR) and Principal component thermography (PCT) were discussed and demonstrated with the use of a Kevlar composite panel having 16 artificial defects; DAC and TSR (2nd derivative) were the two techniques that showed promising results as they were able to identify the highest number of defects. In another study [23], standard thermal image processing techniques and edge detection were applied to thermal images of defective PV modules for hot spot detection. Firstly, basic processing methods such as ROI analysis, line profile analysis and image histogram analysis were implemented to demonstrate that an irregular temperature distribution in a module indicated the presence of hot spots. Canny edge detection and image segmentation were then applied together to successfully diagnose 13 out of 14 defective cells in PV-1 (array) and 27 out of 29 defective cells in PV-2 (array). Almeida et.al [27] utilized a powerful region-based segmentation tool known as Watershed Transformation for diagnosing faults in surge arresters. In their study, the proposed methodology was able to classify the faults as normal,

faulty, light or suspicious and in addition, it was deemed capable of detecting both severe and developing faults; it had an overall accuracy of 90.6% with SiC surge arresters and 85.9% with ZnO surge arresters. Other image processing methods such as Laplacian model and Panel Energy Image have been investigated in [26, 28] and proved to be reliable and effective in diagnosing faults in PV systems.

In the final step of the IRT-based automatic diagnosis system, the relevant features obtained from image processing are then fed to a classifier. Classifiers learn these features automatically and realizes complex patterns to make intelligent decisions based on the data [29]. In this way, they can help distinguish thermal images that correspond to defective modules from those that do not. In the recent past, machine learning-based image classification has been the primary approach in many field applications such as medicine, computer vision and also PV systems monitoring. Since ML is a data-driven approach, it requires a large historical dataset [30] and the availability of large datasets in the field of PV systems monitoring makes it convenient for PV fault classification. In addition, they are capable of handling the non-linear and complex settings of PV systems where unpredictable weather conditions affect the output of the PV system [31].

Jiang and Maskell [32] implemented a 2-layered ANN in conjunction with standard analytical methods for fault detection and diagnosis in string-based PV systems. The multi-layered ANN is first trained with an input dataset consisting of irradiance and module temperature values measured from healthy PV systems. It is then used to determine the expected power output of the PV system, which is compared with the actual power produced by the PV system. Finally, the variation between the two is further utilized by analytical methods to diagnose and identify the type of faults. The methodology proved to have a good accuracy and fast detection response.

In a similar study [33], a 3-layered feed forward neural network has been utilized for detecting and locating short circuit faults in a 3x2 PV array. In order to achieve high accuracy, 12 ANN models were developed for investigating 12 cases of short-circuit faults. The models were trained with different input values of irradiance level (E), cell temperature ($T_c$), Maximum Power Point (MPP) voltage ($V_{mp}$) and current ($I_{mp}$) so that they can respond well to different environmental conditions. The trained ANN models are then validated by using it to estimate the terminal voltages of each module under different input scenarios. A controller, based on the actual string module voltages, is used to activate the ANN model that works best under fault condition, while the rest of the models remain deactivated. The results of the study were very promising as the proposed method can locate short-circuit faults in the same string with high accuracy.

To research other ML techniques, Dunderdale et.al [34] used infrared imaging to analyze the performance of deep learning-based classifiers such as VGG-16 and Mobile-Net, and feature-based classifiers such as Scale Invariant Feature Transform (SIFT), in detecting and classifying different PV faults. Due to limited dataset, the study used data augmentation techniques and optimizers such as stochastic gradient descent (SGD) and Adam optimizer for increasing the classification accuracy. Their results showed that the feature-based methods were able to classify the defects with an average accuracy of 91.2%, while Mobile-Net architecture (with Adam optimizer) and VGG-16 architecture (with SGD optimizer) achieved classification accuracies of 89.5% and 85.8% respectively. Therefore, the study came to the conclusion that Mobile-Net Convolutional Neural Network had the best classification accuracy, and deep learning models in general, are much better at classifying module defects than feature-based models.

Furthermore, another study [35] implemented the VGG-16 Convolutional Neural Network for assessing its performance in predicting PV cell degradations. The VGG-16 network was trained

with a dataset consisting of 3336 (2525 normal cells and 811 degraded cells) infrared images that were captured by a drone. In order to provide a precise and noise-free dataset for input, the images were manually interpreted and labelled using human observers. Data augmentation techniques like flipping and rotation were also performed on the images in order to balance the classes in the training data and ensure effective results. From the results, it can be seen that the VGG-16 architecture achieves better classification accuracy with a balanced dataset in contrast to an unbalanced dataset. Moreover, the network had the best accuracy of 75% (0.75 F1-score) when a balanced dataset comprising of rotated images were used.

Other popular machine learning methods which have proven to be effective in the PVSM field includes: k-Nearest-Neighbors (kNNs), ANNs optimized by Genetic Algorithms (GAs/ANN) and kernel extreme learning machine (KELM). The studies are given in [36, 37].

With the increasing application of large-scale PV systems, developing an automatic inspection method that is fast, simple and reliable is very important and demanding. While it is evident from the literature that the use of Artificial Neural Networks (ANNs) has been immense for automatic diagnosis of PV systems, researchers are reluctant to explore other machine learning techniques such as Convolutional Neural Networks (CNN) and Extreme Learning Machine (ELM). This is due to the fact that researchers tend to rely on previous works involving ANNs, which proved to be able to produce accurate and reliable results. Despite their admirable performances, ANNs have their own drawbacks when it comes to classifying images. As mentioned in [38], one of the drawbacks is that training an ANN model requires high computational power particularly when dealing with large images. With an increase in the image size, the number of trainable parameters increases significantly and can easily result in over-fitting. Additionally, it can be difficult to fine-tune an ANN model since it cannot maintain the spatial features of an image which are important for accurate classification. While an ANN architecture is made up of neurons which are connected to every other neuron, the

neurons in a CNN layer are connected only to a small number of neurons in the previous layer and not to all other neurons. This allows CNNs to automatically learn complex features of an image such as spatial features, which are essential for accurately identifying and locating objects in an image [39]. Furthermore, as opposed to ANNs, CNN models require fewer parameters to train due to parameter sharing and this results in reduced computational cost [38]. More recently, a new learning algorithm for single-hidden layer feedforward neural networks (SLFNs) known as Extreme Learning Machine (ELM), has emerged to be more advantageous than state-of-the-art learning algorithms such as SVMs and deep learning algorithms. First suggested by Huang, Zhu and Siew [40], ELMs have a learning speed that is extremely faster than classical learning algorithms like back-propagation algorithm and gradient-based learning algorithms. Such training efficiency is achieved by utilizing random feature mapping, where hidden nodes are generated randomly and do not need to be iteratively trained. ELMs can also achieve better generalization performance over traditional feedforward neural networks by reaching the smallest training error and smallest norm of output weights [41]. Last but not the least, ELMs can provide straightforward solutions without any issues of overfitting, which makes it suitable for problems with small datasets.

This study proposes a conventional monitoring approach based on infrared imaging, for fault diagnosis of PV cells. Initially, the acquired thermal images of the cells in RGB format are pre-processed in 3 different ways: temperature difference ($\Delta T$) analysis, thresholding and contour detection and data augmentation. In line with the $\Delta T$ value and a threshold value for the percentage degradation, each cell was labelled as either defective or non-defective. Moreover, due to the availability of a very limited dataset, we implemented data augmentation techniques such as flipping and rotating the images, to increase the training dataset for the deep learning classifiers. In the final step of our approach, the proposed deep learning models VGG-16,

VGG-19 and Efficient-Net, are trained with the labelled dataset and a testing dataset is used later to compare the performances of the classifiers.

## 1.5 Thesis Organization

In further proceedings of the book, contents have been divided into four more major chapters in order to thoroughly elaborate and illustrate our work. Earlier in this Chapter, we have provided an elaborated introduction to our work, with incentives and objectives. In Chapter 2, we have shed some light on the proposed methodology and the different techniques that were employed in order to obtain our results. Some results were illustrated in the form of images and tables as-well. Subsequently, in Chapter 3, we have provided some insight on the deep learning networks that we have used, their architectures and their significance in image processing tasks. Within Chapter 4, results of our work have been illustrated and compared using tables and graph plots. Finally, Chapter 5 provides an overview of our work and discusses the prospect and potential of our work.

# Chapter 2

# Proposed Methodology

## 2.1 Image Classification and Prediction

Over the recent years, Convolutional Neural Networks have to be proven to excel immensely when it comes to image classification and has shown outstanding effectiveness in terms of predictions. The basis of image classification with the aid of any Convolutional Neural Networks is labelling the input images that will be fed as input for the network to analyze and train on. The deep learning classification algorithm then makes use of different learnable parameters such as weights and biases in order to put more emphasis on certain aspects of the images in order to be able to differentiate them from one another. These aspects can be controlled with the help of matrix of weights termed filters (features) that signify different aspects of an image such as vertical edges and horizontal edges, etc. Deeper into the network, the filters are designed to realize more complex attributes of an image.

In our study, we have proposed several different pre-existing Convolutional Neural Network Architectures in order to oversee autonomous defect detection in Infrared images of commercially available solar cell modules. Our proposed methodology involves the follow neural network architectures – VGG16, VGG19 & EfficientNet (B5-B7); trained accordingly in order to differentiate between defective and non-defective. The method we have proposed has several different phases which are illustrated in figures 2.1 and 2.2.

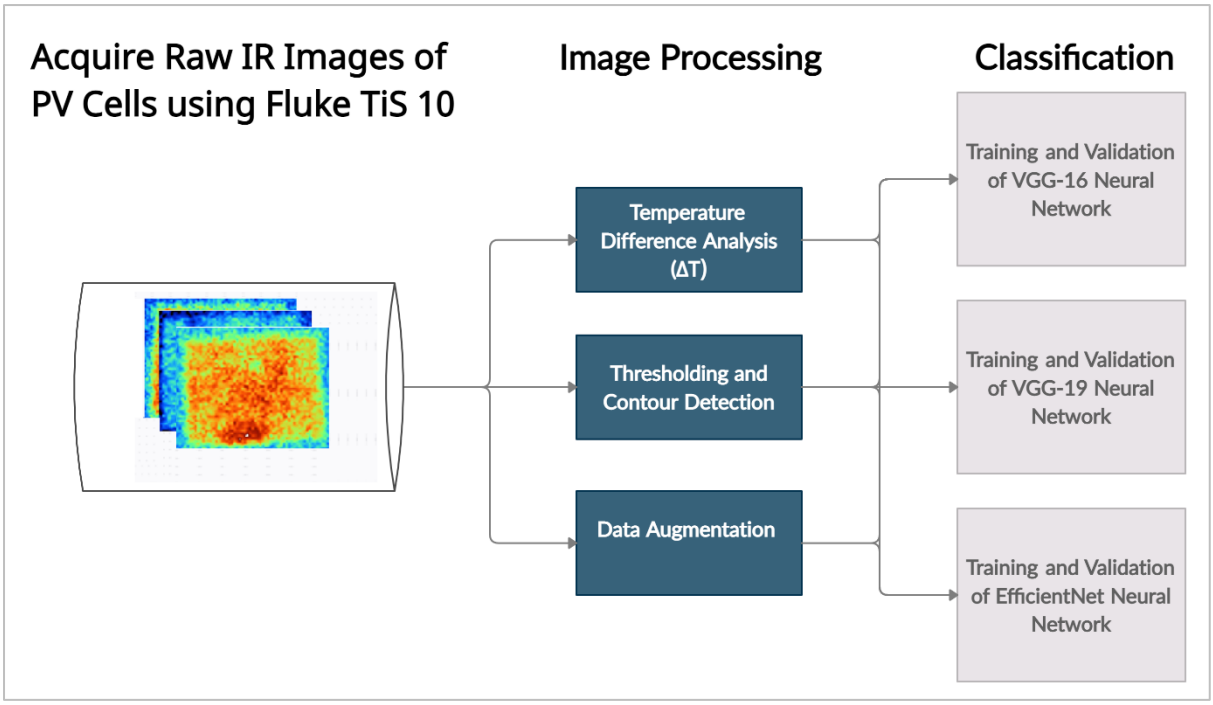



Figure 2.1 Block Diagram Depiction of Proposed Methodology

## 2.2 Infrared Thermography

Infrared Theormography involves the study of infrared emission off of any heat emitting surface with the use of thermograms with the objective of inspecting heat distribution over the surface. Use of Infrared images in order to monitor and inspect the electrical condition of an electrical module has become very much prominent over the years, with the objective of early fault diagnosis by studying the thermal anomalies on the surface with the help of the infrared images. It does this by sensing infrared energy emission (temperatre measurement) of an electrical module. Abnormal generation of heat due to high electrical resistance within any electrical module will usually reorient the surface heat signature due to large change in heat and usually signifies an electircal failure.

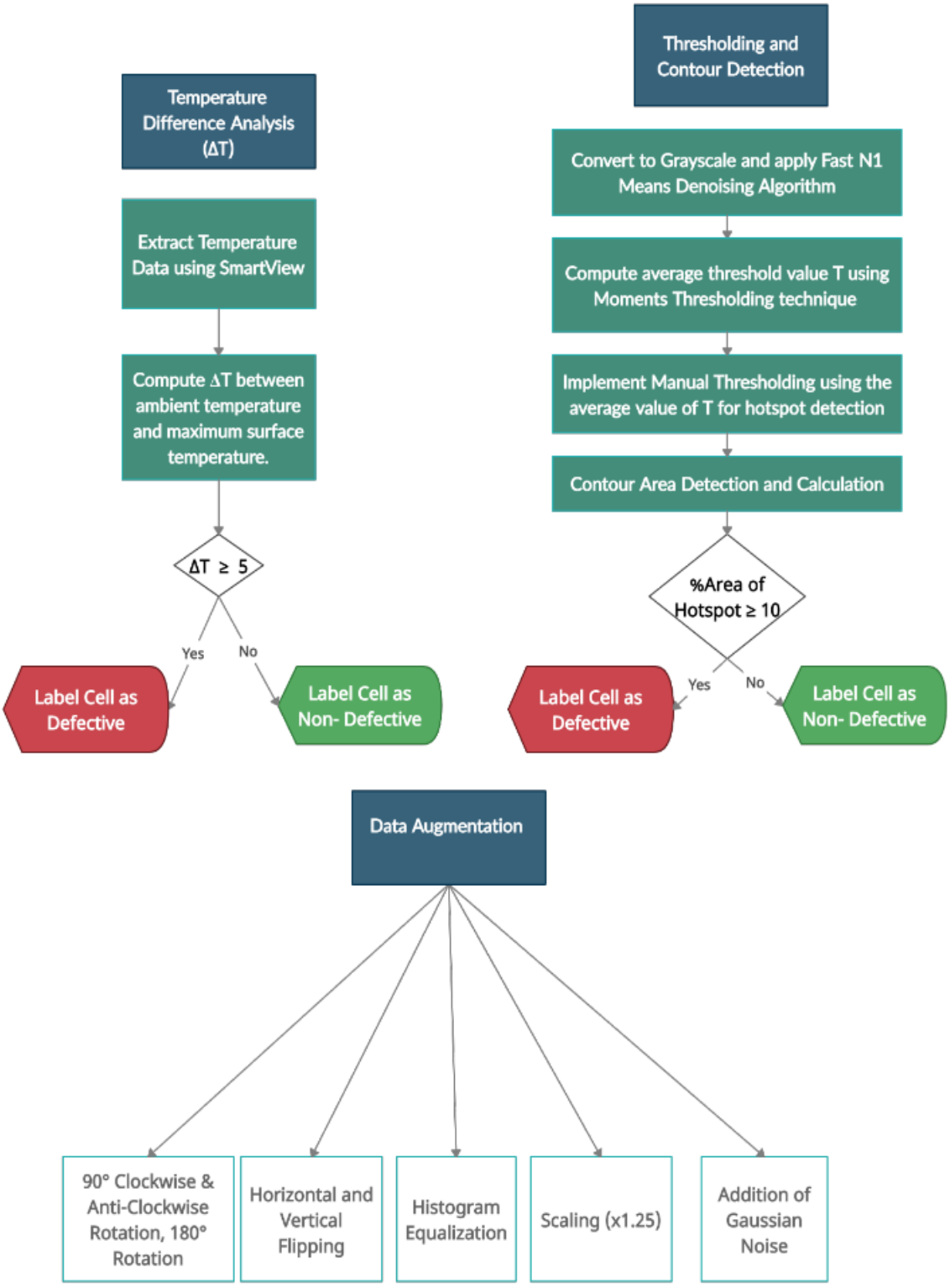



Figure 2.2 Workflow Chart of Image Processing Techniques

### 2.2.1 Need for Automation

Conventionally, direct measurement, inspection and analysis of electrical equipments with the help of infrared images is pretty convenient. However, the analysis results are subject to influence by certain factors such as different environmental effects and equipment's condition. As such, direct inspection without consideration of such factors will certainly generate inaccurate measurements and hence incorrect conclusions will be made. Consequently, it has become imperative to pursue more ehnanced inspection techniques by the implementation of autonomous diagnosing procedure in order evaluate an electical equipments condition, primarily via image processing and aritifical intelligence.

### 2.2.2 Thermal Anomalies

Electrical modules usually come with power rating that signifies the amount of energy that the device can accommodate without becoming defective. Excessive power generation leads to overheating which will eventually reduce the efficiency and lifetime of the device itself. Typical faults within an electrical module may arise from factors such as poor electircal connections, short or open-circuit conditions and improper installation. Usually, these type of faults lead to the generation of abnormal amount of heat, which can be recognized by studying the heat signatures of an infrared image; a high temperature point or area, compared to the background or a referance value, indicates the portential location of the defect.

### 2.2.3 IR Imaging Approach

It is important to mention that when it comes to inspection of a large electrical system constituing several hundreds of equipments, analysis becomes much more sophisticated and expensive; more of a reason why an autonomous diagnosis method is imperative that offers enhanced accuracy and efficiency. As such, due to surge in developments in computer vision

using image processing as its basis, coupled with significant inclusion of artificial intelligence, it has become prominent in fault monitoring and diagnosis of electrical equipments and modules.

Our dataset constitutes infrared images of PV modules that depicts the normal or abnormal distribution of heat signatures over the module itself. The images of the modules have been captured using an IR camera termed Fluke TiS 10. In our study, we have pursued the active approach, indoor IR Thermograhpy, in order to classify whether a module is defective or normal. This is depicted in Figure 2.3.

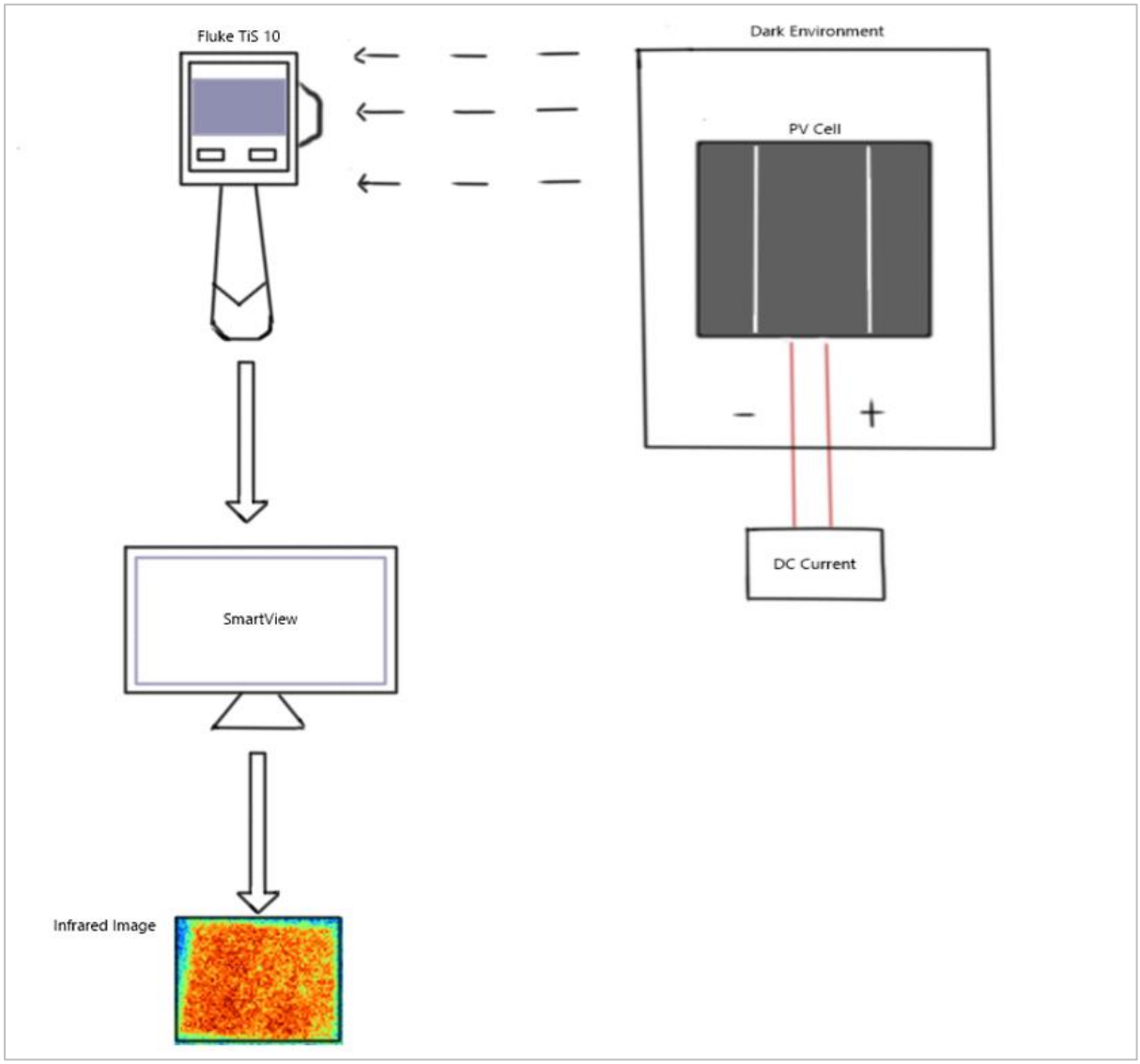



Figure 2.3 Active Approach Indoor IR Thermography

## 2.3 Dataset

Initially, we prepared a dataset containing raw Infrared images of different PV modules via IR imaging. For this, we have employed indoor IR imaging which was carried out in a dark environment, also adhering to other required conditions. Image obtained from IR imaging were then processed in SmartView Imaging Software. Subsequently, we pursued different image processing techniques in order to attempt to correctly classify and label these images for our algorithms to study, train and make predictions on. The prepared dataset comprises of a total of 15 infrared images of solar panel modules pertaining to 15 different modules, each taken one minute after consistent current flow within the module. It is important to note the significance of using 1-minute images; to detect any potential abnormal changes in temperature difference due to the flow of current. The images were prepared and processed in JPEG format.

Deep Neural Networks usually require a significantly large dataset to train on in order to obtain a good or mediocre performance. However. In our case, due to covid restrictions, extending the size of the dataset was not viable and we were compelled to improvise. Figure 2.4 illustrates an image constituted by the dataset.

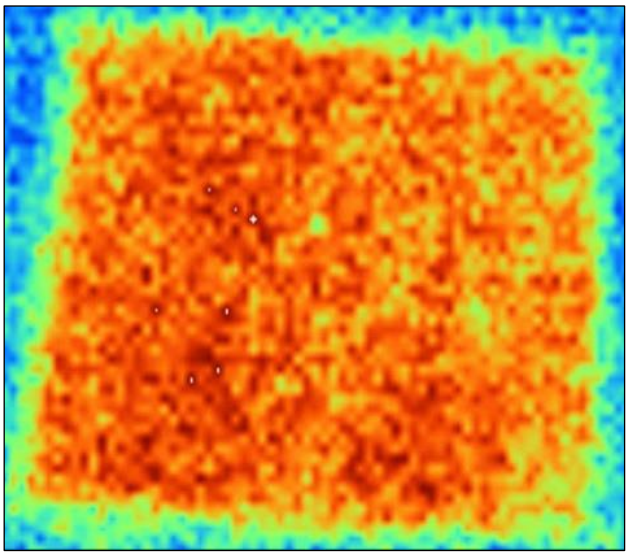


Figure 2.4 Sample Image of a Photovoltaic Cell

## 2.4 Significance of Image Processing

In order to differentiate between defective and normal modules among our test samples, we pursued 3 different image processing techniques, 2 of which were used to prepare our dataset for training, validation and dataset. Additionally, we also employed another image processing technique in order to increase the size of our dataset and observe its impact on the performance of the neural network algorithms.

- Determination of temperature difference ΔT
- Thresholding and Hotspot Contour Detection
- Histogram Analysis (Results were not used for Training)
- Data Augmentation

## 2.5 Determination of Temperature Difference (ΔT)

In order to determine whether the acquired infrared images of the PV cells, indicate if they are defective or non-defective, one of the approaches we implemented involves evaluating the difference between the maximum surface temperature of the PV cell and its minimum ambient temperature. The IR images were analyzed using the Fluke SmartView thermal imaging software, which automatically pinpoints and labels the maximum and minimum temperatures in an infrared image. For this approach, we assumed that a temperature difference greater than or equal to 5°C would indicate a defective cell and a healthy cell otherwise. The formula for computing the temperature difference ΔT is given as:

$$\Delta T = (Maximum\ Temperature - Ambient\ Temperature) \quad (1)$$

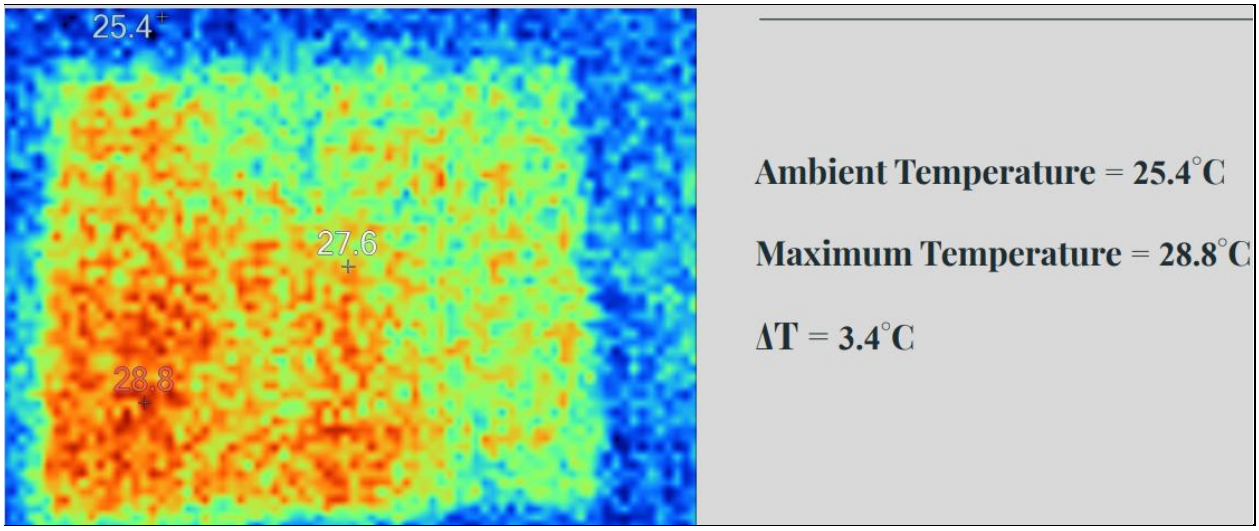



Figure 2.5 Non-Defective PV Cell (Cell #1)

Figure 2.5 shows a healthy cell which has a point that indicates a maximum surface temperature of 28.8°C and a point that is at a minimum ambient temperature of 25.4°C. The temperature difference ΔT is thus 3.4°C, which is less than our predefined condition of 5°C.

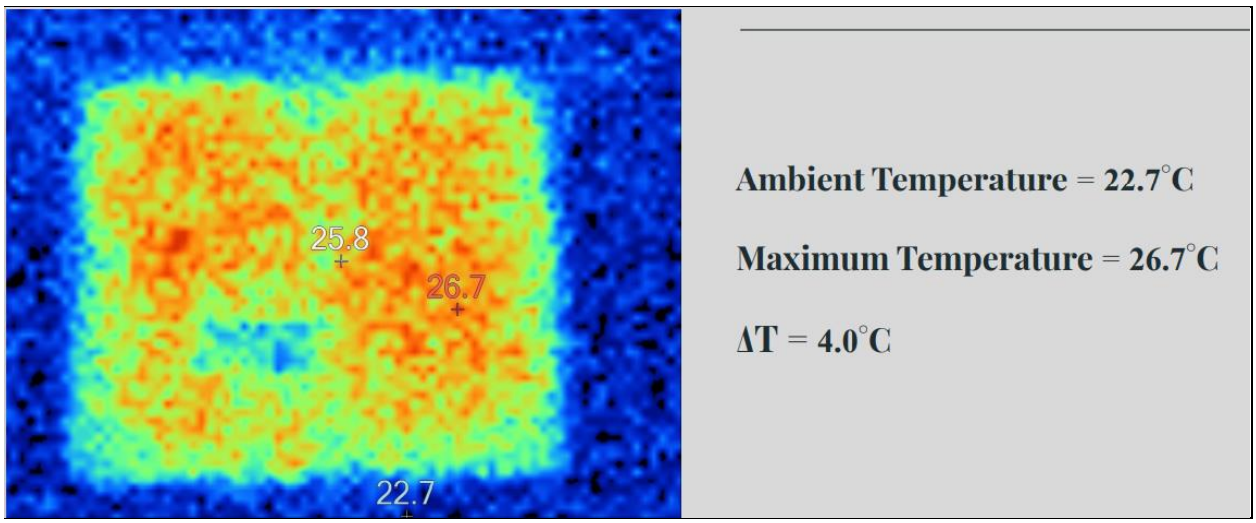



Figure 2.6 Non-Defective PV Cell (Cell #7)

Figure 2.6 shows another healthy cell which has a point that indicates a maximum surface temperature of 26.7°C and a point that is at a minimum ambient temperature of 22.7°C. The temperature difference ΔT is thus 4.0°C, which is less than our predefined condition of 5°C.

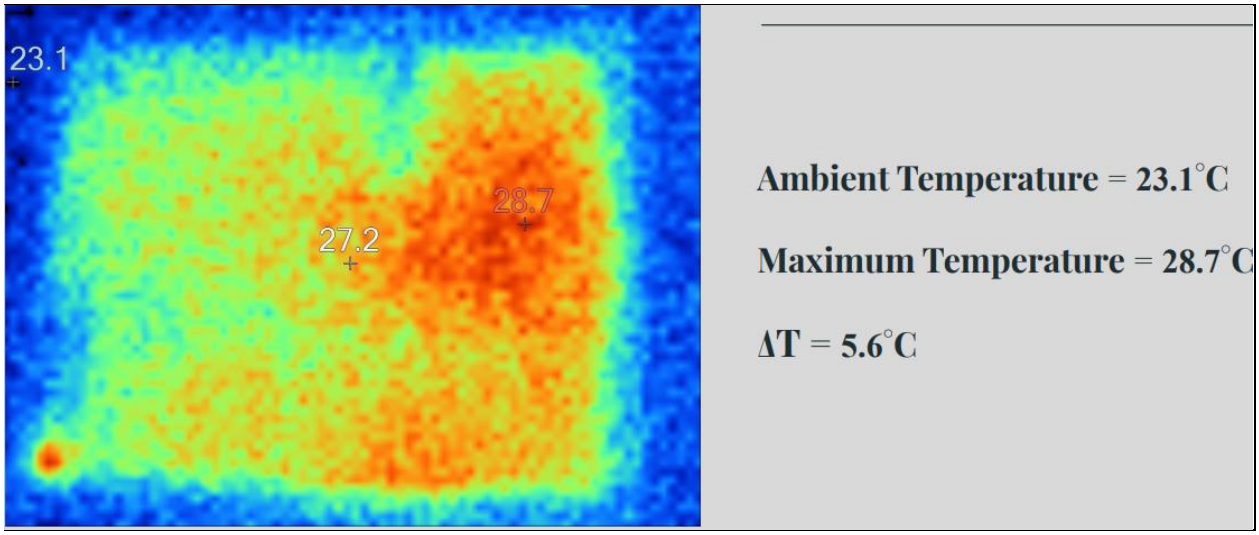



Figure 2.7 Defective PV Cell (Cell #4)

Figure 2.7 shows a defective cell which has a point that indicates a maximum surface temperature of 28.7°C and a point that is at a minimum ambient temperature of 23.1°C. The temperature difference ΔT is thus 5.6°C, which is more than our predefined condition of 5°C.

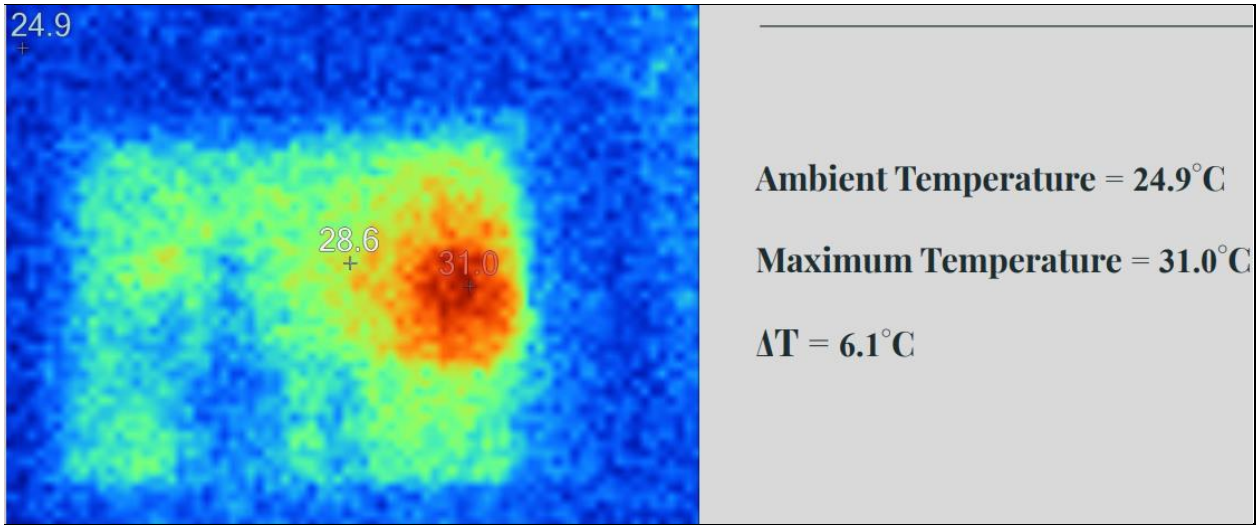



Figure 2.8 Defective PV Cell (Cell #6)

Figure 2.8 shows another defective cell which has a point that indicates a maximum surface temperature of 31.0°C and a point that is at a minimum ambient temperature of 24.9°C. The temperature difference ΔT is thus 6.1°C, which is more than our predefined condition of 5°C.

The infrared images were taken within the first minute of the cells' operation and in total, 15 different photo-voltaic cells were used for the analysis. Table 2.1 summarizes the results of the ΔT analysis of all the cells, which shows that 3 of the cells are defective while the rest are healthy. In order to provide a better perspective of the results of our analysis, a histogram in Figure 2.9 shows the distribution of the defective and non-defective cells with respect to the temperature difference ΔT.

### 2.5.1 Splitting the dataset

In order to train the different neural network algorithms, we then split the dataset that was labelled using temperature difference. The dataset was split into training set, validation set and testing set. The training set constituted a total of 5 images (1 defective, 4 normal), the validation set constituted a total of 5 images (1 defective, 4 normal) and the test constituted a total of 5 images (1 defective, 4 normal).

Table 2.1 Temperature Difference (ΔT) of all 15 PV Cells

| Cell No. | Ambient Temperature, $T_A$(°C) | Maximum Temperature, $T_m$ (°C) | $\Delta T = T_m - T_A$ (°C) | Condition of the Panel |
|---|---|---|---|---|
| 1 | 25.4 | 28.8 | 3.4 | Non-Defective |
| 2 | 25.0 | 29.1 | 4.1 | Non-Defective |
| 3 | 26.2 | 30.2 | 4.0 | Non-Defective |
| 4 | 23.1 | 28.7 | 5.6 | Defective |
| 5 | 25.9 | 29.6 | 3.7 | Non-Defective |
| 6 | 24.9 | 31.0 | 6.1 | Defective |
| 7 | 22.7 | 26.7 | 4.0 | Non-Defective |
| 8 | 24.3 | 28.1 | 3.8 | Non-Defective |
| 9 | 25.6 | 31.6 | 6.0 | Defective |
| 10 | 24.6 | 28.7 | 4.1 | Non-Defective |
| 11 | 26.1 | 29.7 | 3.6 | Non-Defective |
| 12 | 24.3 | 28.6 | 4.3 | Non-Defective |
| 13 | 24.5 | 28.0 | 3.5 | Non-Defective |
| 14 | 24.5 | 28.2 | 3.7 | Non-Defective |
| 15 | 26.1 | 30.1 | 4.0 | Non-Defective |

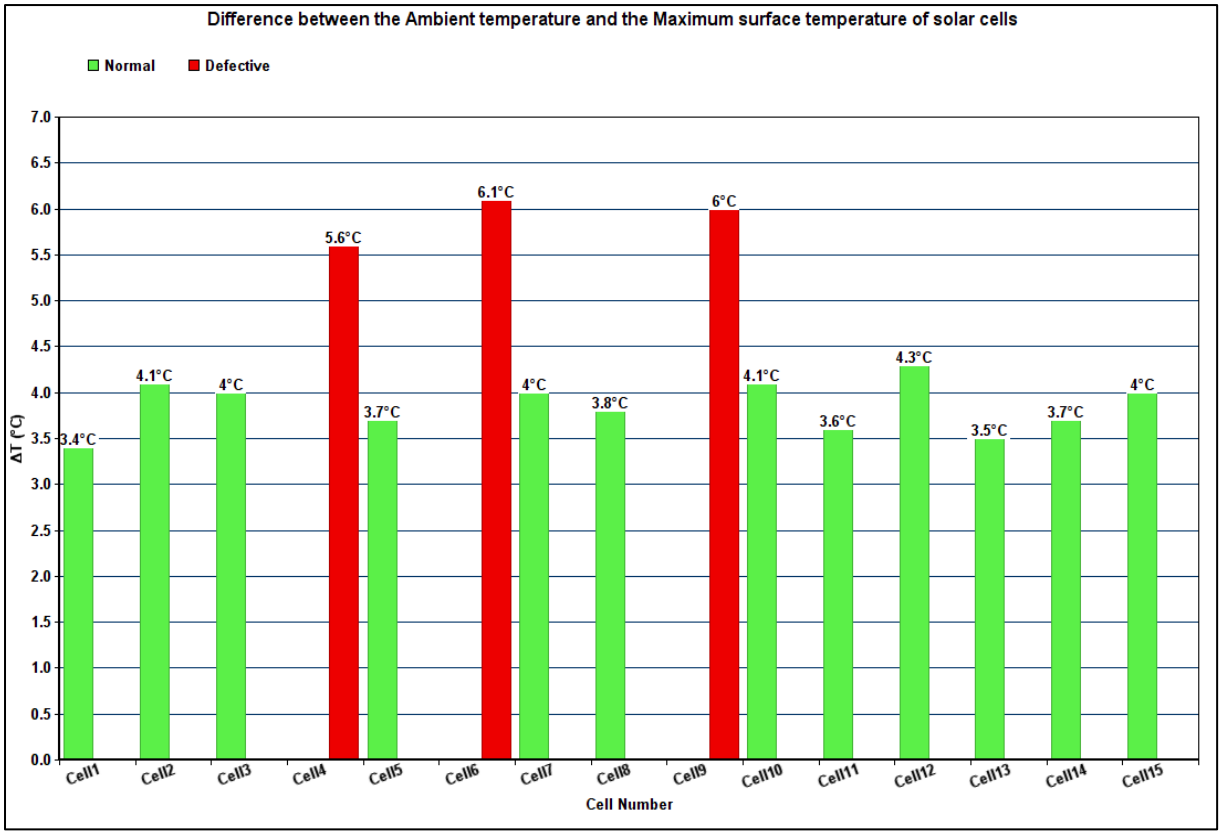



Figure 2.9 Histogram distribution showing (ΔT) of all the Cells under analysis

## 2.6 Thresholding and Contour Detection

In this particular approach, we have attempted to detect the hotspot contour on the infrared image of the module and determine a rough estimation of the area taken up by the hotspot with respect to the module area. We have chosen contours as they are useful in emphasizing an area of interest (hotspot) on the image by using a bounding shape that is used to indicate the region of significant hotspot. Essentially, contours are drawn as a curve joined by a series of continuous points that constitute the same pixel color or intensity. Initially, we applied image segmentation on the images with the objective of making the image less sophisticated and more convenient and simpler to analyze the image pixels. The whole method was implemented using python OpenCV on Google Collaboratory.

### 2.6.1 Denoising

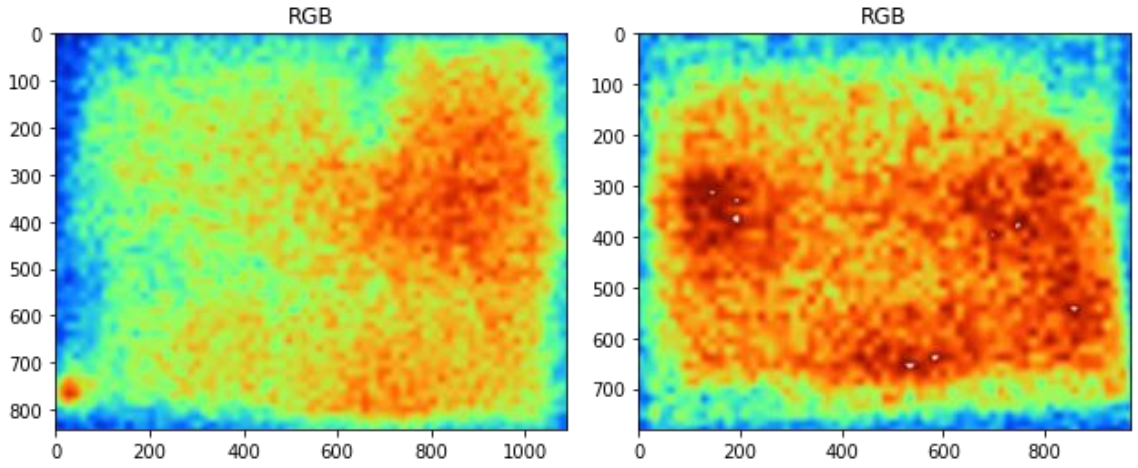
The infrared image we obtained using the IR camera was in RGB format. In order to pixelate and remove noising from the image, we converted the image into grayscale and then applied FastN1MeansDenoising in order to remove significant noise (as noise introduces unwanted influence in the results) from the image. The Denoising algorithm only is applicable on grayscale images. This denoising technique operates by adjusting the color of a pixel after the image is scanned using a filter window. It does this adjustment by determining the average of the colors of the neighboring pixels. If most of the surrounding pixels within a particular filter window are found to be quite similar, the noise standard deviation will be low, which will make it simpler to obtain the noise and subtract it from the image. The algorithm was implemented by following these significant parameters:

- templateWindowSize – 7; refers to the number of pixels within the template patch used to compute weights. An odd number is recommended.

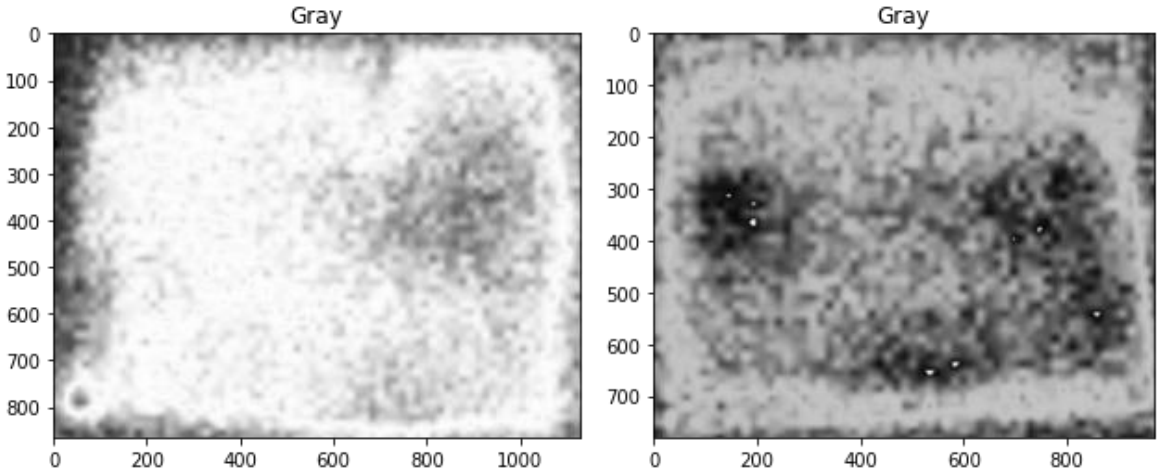
- searchWindowSize – 21; refers to the number of pixels within the search window that is used to determine weighted average for a particular pixel. Greater the 'searchWindowSize', more time is invested in denoising. An odd number is recommended.

- h – 10; a parameter that can be altered to regulate the strength of the filter. A bigger value extensively removes noise but also takes out significant image details. Conversely, smaller value of h will strive to preserve more of the image details but fails to reduce significant amount of noise. Hence, we have used a larger value of h.

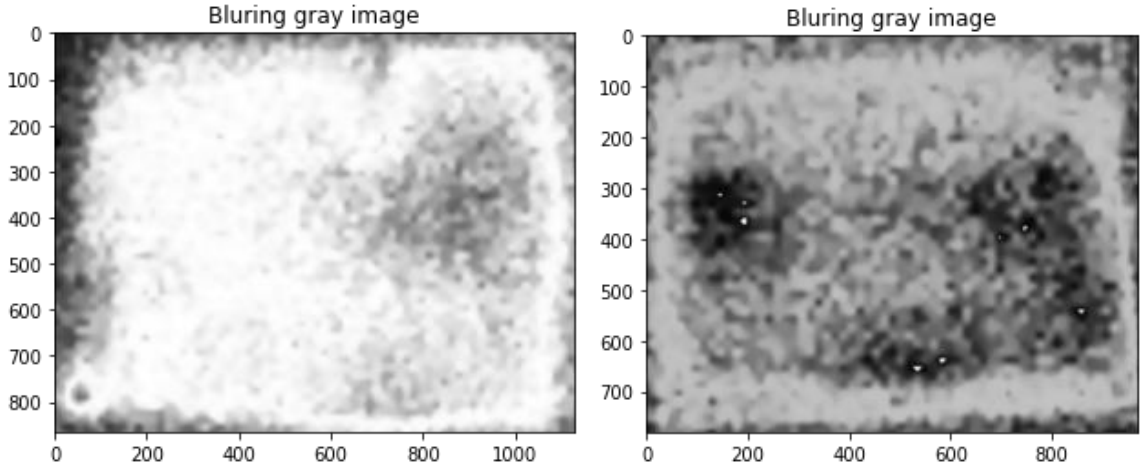
Figure 2.10 illustrates examples of the images obtained as a result of this technique.




(a)




(b)




(c)

Figure 2.10 Applying FastN1MeansDenoising; (a) Original RGB Images (b) Grayscale Images (c) Grayscales Images after Denoising

### 2.6.2 Thresholding

Subsequently, we then employed a thresholding method in order to highlight the significant hotspots. To be more particular, we pursued a binary inverted approach to thresholding. In such a technique, the pixels in an image are replaced with either black or white pixels depending on a threshold value of pixel intensity (between black and white). In this study, if the pixel intensity value was greater than the threshold value, we assigned that pixel to be a particular maximum value and if it was otherwise, it would be assigned a value of 0, in order to create two distinctions. This can be represented by the following function: -

$$\texttt{dst}(x,y) = \begin{cases} 0 & \text{if } \texttt{src}(x,y) > \texttt{thresh} \\ \texttt{maxval} & \text{otherwise} \end{cases} \tag{2}$$

It is important to mention that this technique requires the input of a grayscale image in order to function. In our case, we have assigned a 'maxval' of 255 for pixels that have intensity values greater than the threshold value.

Moreover, in order to determine an optimum threshold value 'thresh' for binary thresholding, we have used another software called 'ImageJ' that implements 'moments thresholding' [42] on the grayscale image in order to determine the optimum threshold value. It functions by making use of the gray level difference between object borders and background, and also the difference among pixels in a particular region. The threshold value within an image is determined by calculating the weighted average of the image pixels' intensities of a particular region (centroid), in our case, the hotspot region.

In order to obtain the optimum threshold value, we determined all the threshold values pertaining to all the images within our dataset and averaged them. Table 2.2 illustrates this process.

Table 2.2 Threshold Values Determined using ‘Moments Threshold’

| **Image Sample** | **Threshold** |
|---|---|
| 1.jpeg | 137 |
| 3.jpeg | 138 |
| 9.jpeg | 133 |
| 11.jpeg | 137 |
| 13.jpeg | 143 |
| 16.jpeg | 119 |
| 23.jpeg | 137 |
| 24.jpeg | 137 |
| 32.jpeg | 138 |
| 33.jpeg | 90 |
| 35.jpeg | 138 |
| 38.jpeg | 135 |
| 48.jpeg | 127 |
| 55.jpeg | 128 |
| 74.jpeg | 133 |

Average = (137+138+133+137+143+119+137+137+138+90+138+135+127+128+133)/15

= 131.3

≈ 131 (Optimum Threshold Value)

The obtained optimum threshold value was then used in the binary inversion threshold algorithm as ‘thresh’ in order to signify which pixels should be assigned the ‘maxval’ and which ones are to be assigned a value of 0. The resulting images are as illustrated in Figure 2.11.

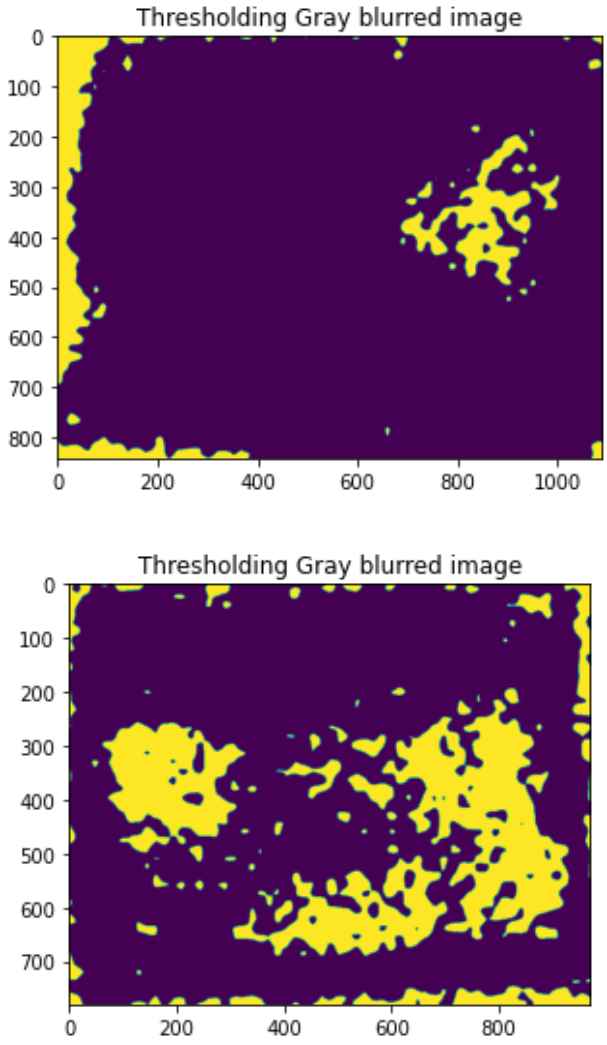



Figure 2.11 Applying Binary Inverted Thresholding

### 2.6.3 Contour Detection

Subsequently, we then apply a contour retrieval function in order to determine the contours on the thresholded image. We get an output image which displays the detected contours and additionally, the algorithm calculates percentage area taken up the different contours and sums them up to find the total contour area. We then manually analyze the images in order to exclude contour area highlighted outside the module and determine a rough estimate of the contour detected within the module, which signifies the significant hotspot region that is defective. This is illustrated in Figure 2.12.

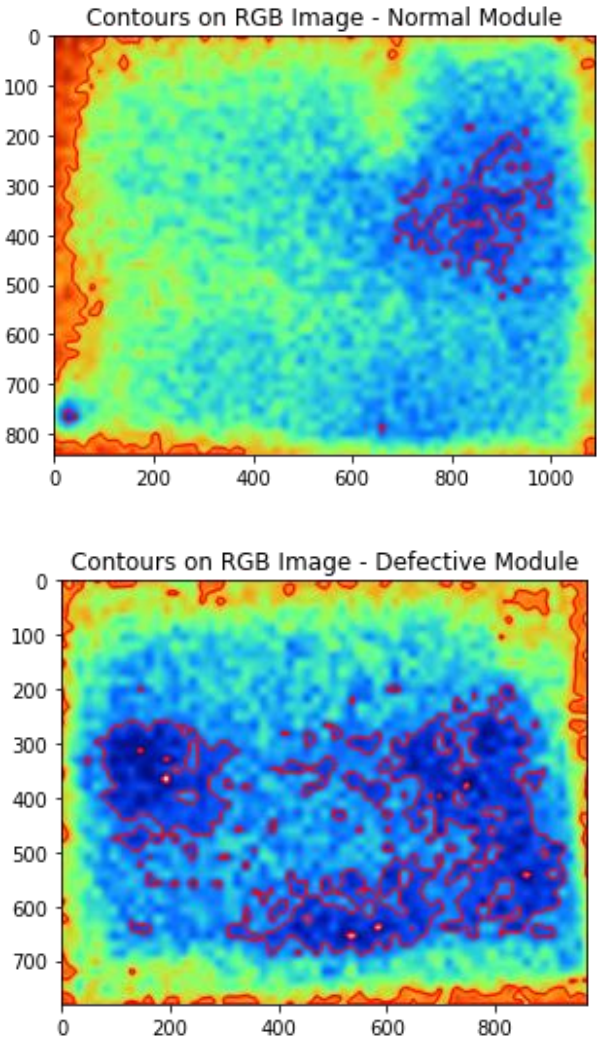


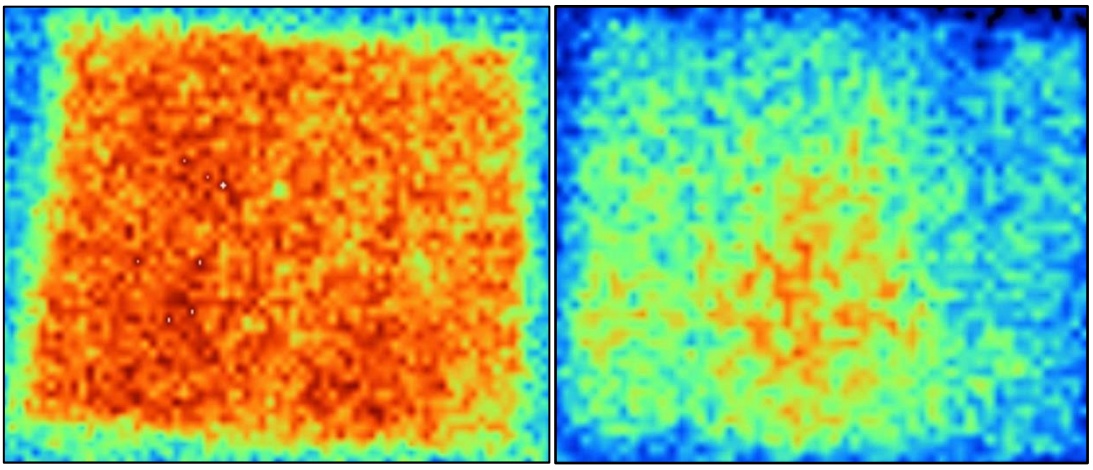

Figure 2.12 Contours on RGB Image

### 2.6.4 Pertaining Dataset

Any image that constituted a total contour area percentage of more than or equal to 10 percent within the photovoltaic cell module, was classified as a ‘defective’ module. Otherwise, it was classified as a ‘normal’ module.

In order to train the different neural network algorithms, we then split the dataset that was labelled using thresholding and contour detection. The dataset was split into training set, validation set and testing set. The training set constituted a total of 6 images (2 defective, 4 normal), the validation set constituted a total of 5 images (2 defective, 3 normal) and the test

constituted a total of 4 images (1 defective, 3 normal). Some examples are illustrated in Figure 2.13.



(a) (b)

Figure 2.13 Classified Images using Threshold and Contour Detection Approach (a) Defective Module (b) Normal Module

Algorithm for thresholding and contouring was taken from [43].

## 2.7 Pixel Distribution Analysis

Furthermore, an additional analysis was carried out using histograms generated from the infrared images with the help of Fluke SmartView thermal imaging software. The histogram represents the frequency of pixels in the images distributed over different surface temperatures. For this analysis, the pixel count $P_m$ at the maximum surface temperature $T_m$ of each cells were determined and then calculated as a percentage of the total pixel count in the images $P_T$. The formula for computing the percentage of pixels at maximum surface temperature is given as:

$$(P_m/PT) \times 100 \quad (3)$$

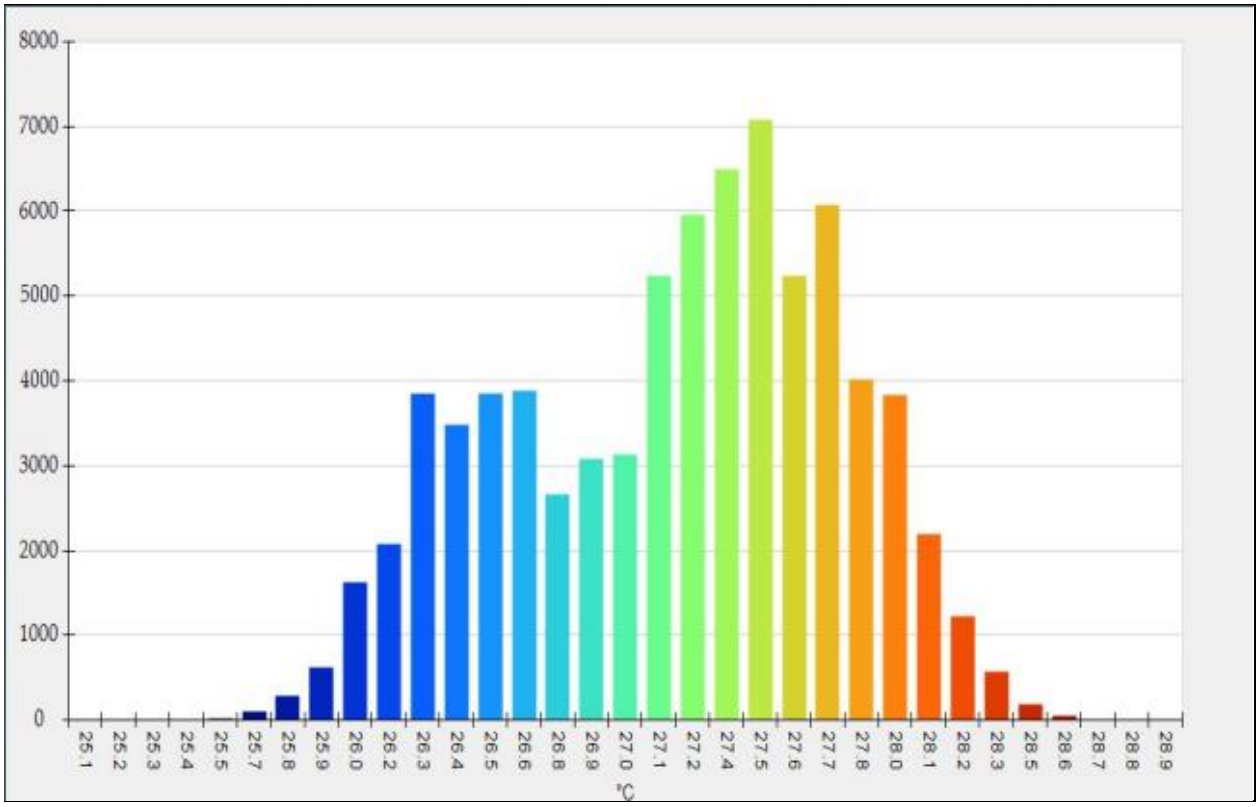



Figure 2.14: Histogram showing pixel distribution of PV Cell #1 (non-defective)

Figure 2.14 shows the distribution of pixels of the same PV cell that was shown in Figure 2.5. For this cell the maximum surface temperature Tm is 28.8°C and the number of pixels at this temperature Pm is 1. This is the lowest recorded observation among all the 15 cells. Since there are 76,800 pixels in total, the proportion of Pm in the IR image in percentage is (1/76800) x100 = 0.001302%

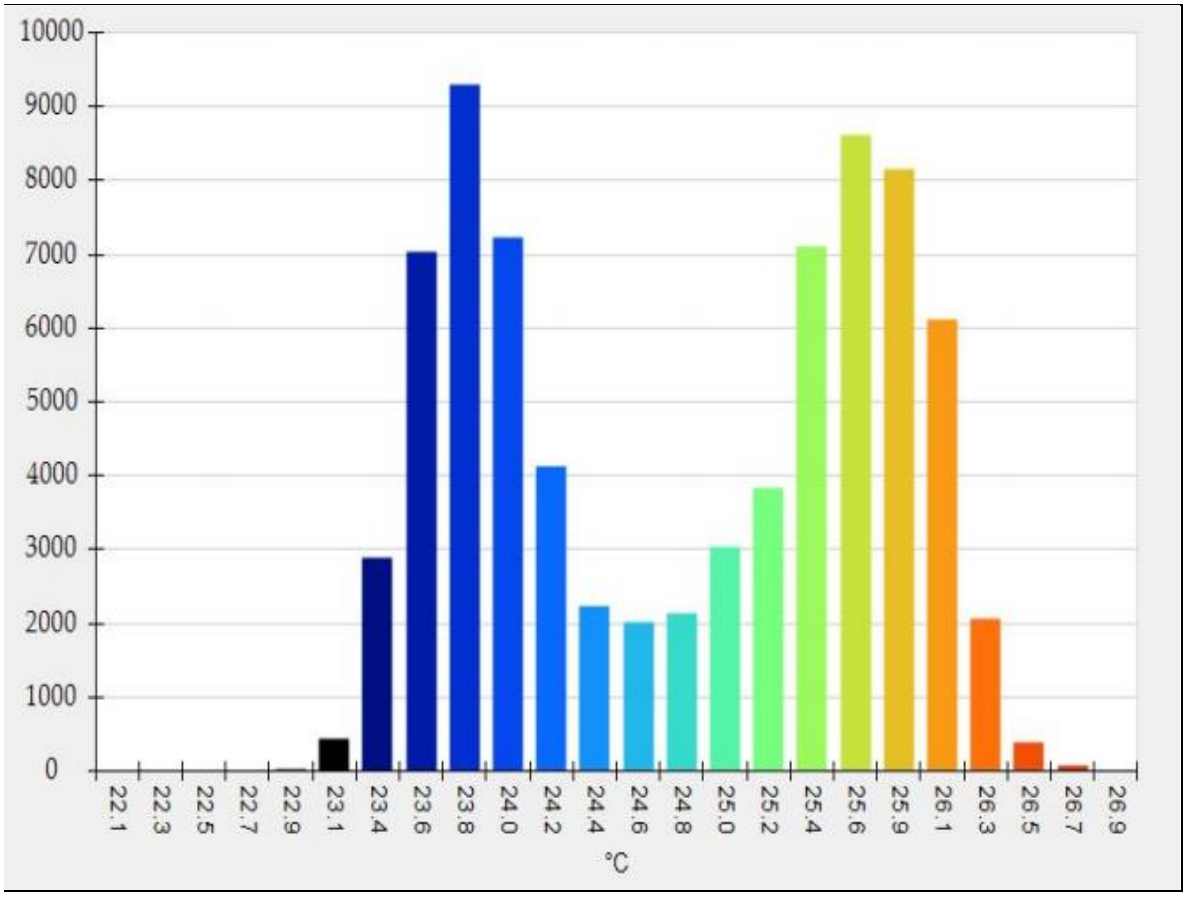



Figure 2.15 Histogram showing pixel distribution of PV Cell #7 (non-defective)

Figure 2.15 shows the distribution of pixels of the same PV cell that was shown in Figure 2.6. For this cell the maximum surface temperature Tm is 26.7°C and the number of pixels at this temperature Pm is 61. This is the highest recorded observation among all the 15 cells. Since there are 76,800 pixels in total, the proportion of Pm in the IR image in percentage is (61/76800) x100 = 0.07943%

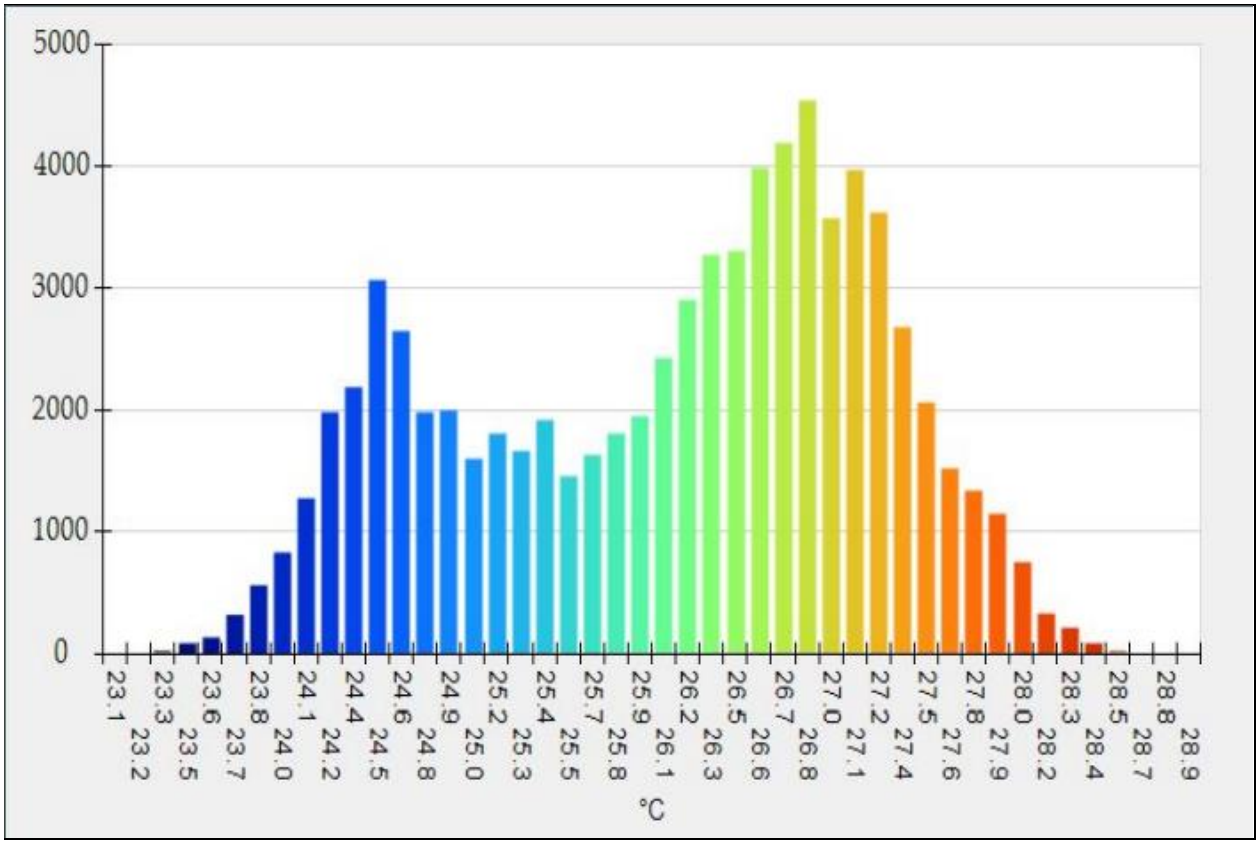



Figure 2.16 Histogram showing pixel distribution of PV Cell #4 (defective)

Figure 2.16 shows the distribution of pixels of the same PV cell that was shown in Figure 2.7. For this cell the maximum surface temperature Tm is 28.7°C and the number of pixels at this temperature Pm is 5. Since there are 76,800 pixels in total, the proportion of Pm in the IR image in percentage is (5/76800) x100 = 0.006510%

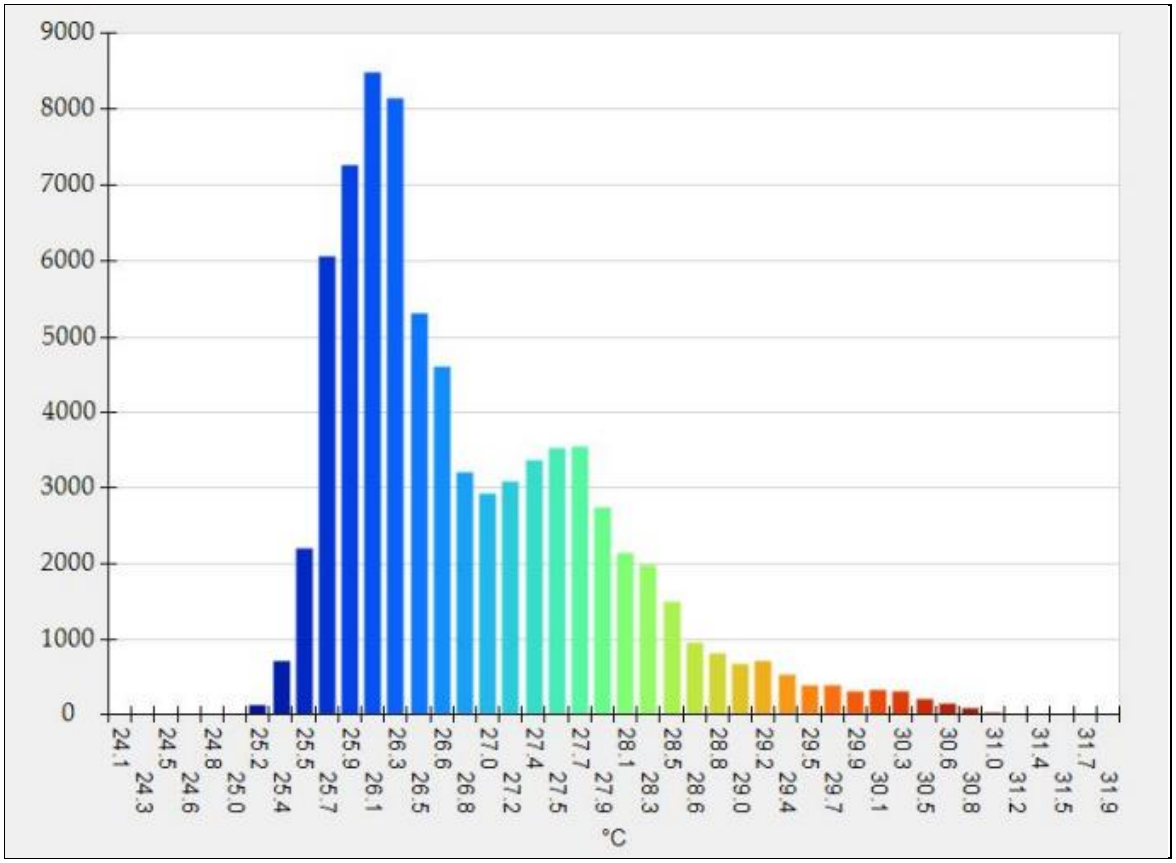



Figure 2.17 Histogram showing pixel distribution of PV Cell #6 (defective)

Figure 2.17 shows the distribution of pixels of the same PV cell that was shown in Figure 2.8. For this cell the maximum surface temperature Tm is 31.0°C and the number of pixels at this temperature Pm is 18. Since there are 76,800 pixels in total, the proportion of Pm in the IR image in percentage is (18/76800) x100 = 0.02344%

Table 2.3 summarizes the results of the pixel distribution analysis of all the cells, which shows that Cell #7 had the highest percentage of pixels at a maximum surface temperature of 26.7°C while Cell #1 had the lowest percentage of pixels at a maximum surface temperature of 28.8°C.

Table 2.3 Summary of Pixel Distribution Analysis of all 15 PV Cells

| Cell No. | Total Pixel Count, $P_T$ (Pixels) | Maximum Temperature, $T_m$ (°C) | Pixel Count at Maximum Temperature, $P_m$ (Pixels) | Percentage of Pm in the IR image (%) | Condition of the Panel |
|---|---|---|---|---|---|
| 1 | 76,800 | 28.8 | 1 | 0.001302 | Non-Defective |
| 2 | 76,800 | 29.1 | 6 | 0.007813 | Non-Defective |
| 3 | 76,800 | 30.2 | 2 | 0.002604 | Non-Defective |
| 4 | 76,800 | 28.7 | 5 | 0.006510 | Defective |
| 5 | 76,800 | 29.6 | 5 | 0.006510 | Non-Defective |
| 6 | 76,800 | 31.0 | 18 | 0.02344 | Defective |
| 7 | 76,800 | 26.7 | 61 | 0.07943 | Non-Defective |
| 8 | 76,800 | 28.1 | 4 | 0.005208 | Non-Defective |
| 9 | 76,800 | 31.6 | 8 | 0.01042 | Defective |
| 10 | 76,800 | 28.7 | 5 | 0.006510 | Non-Defective |
| 11 | 76,800 | 29.7 | 7 | 0.009115 | Non-Defective |
| 12 | 76,800 | 28.6 | 25 | 0.03255 | Non-Defective |
| 13 | 76,800 | 28.0 | 4 | 0.005208 | Non-Defective |
| 14 | 76,800 | 28.2 | 7 | 0.009115 | Non-Defective |
| 15 | 76,800 | 30.1 | 9 | 0.01172 | Non-Defective |

The primary purpose of this analysis was to determine whether there is a clear relationship between the maximum surface temperature of a cell and the pixel count at that temperature; a scatter diagram can further help us understand the relationship between the two. Therefore, a scatter diagram is generated using the recorded observations for $T_m$ and $P_m$, as shown in Figure 2.18.

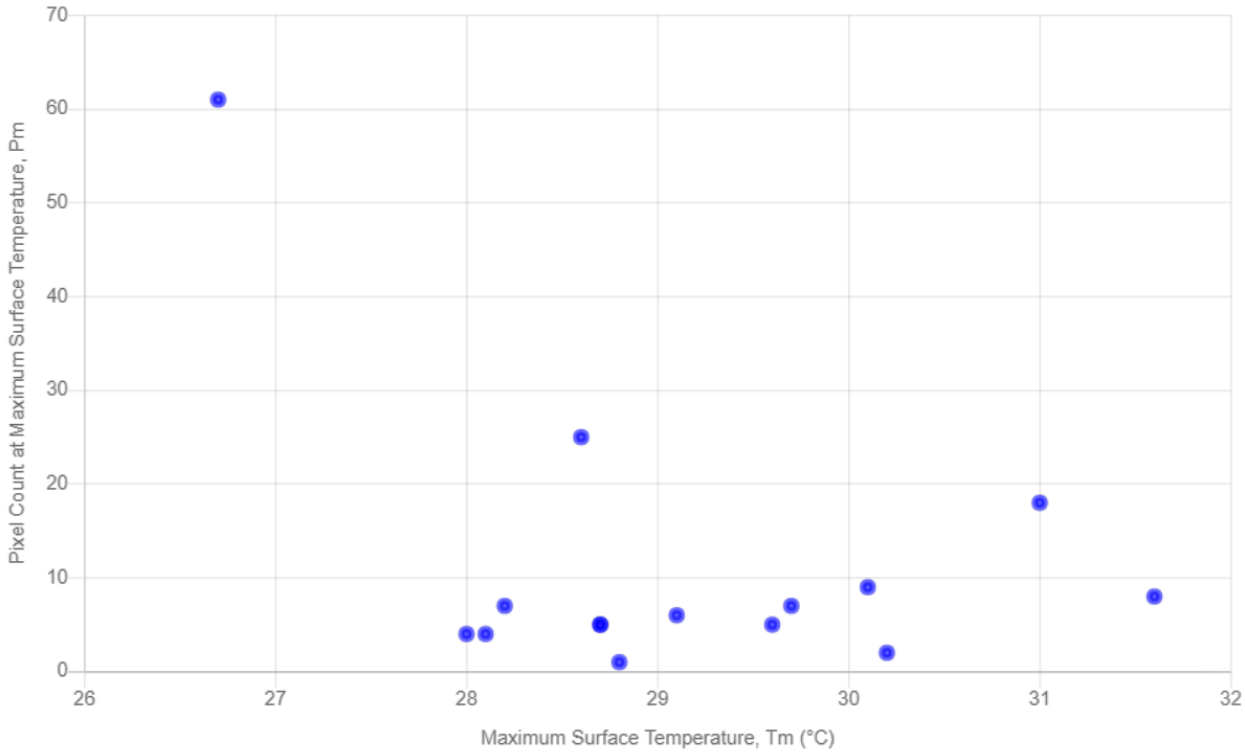



Figure 2.18 Scatterplot of Maximum Surface Temperature, $T_m$ (°C) versus Pixel Count at Maximum Surface Temperature, $P_m$

In order to interpret the above scatterplot properly, we also evaluated the Pearson's correlation coefficient, which is given by: -

$$r_{xy} = \frac{\sum(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum(x_i - \bar{x})^2 \sum(y_i - \bar{y})^2}} \quad (4)$$

| **X Values (Tm)** | **Y Values (Pm)** |
|---|---|
| $\sum x = 437.1$ | $\sum y = 167$ |
| Mean, $\bar{x} = 29.14$ | Mean, $\bar{y} = 11.133$ |
| $\sum (x - \bar{x})2 = 22.096$ | $\sum (y - \bar{y})2 = 3201.733$ |

**X and Y Combined**

N = 15

$\sum (x - \bar{x}) (y - \bar{y}) = -112.48$

rxy calculation

$rxy = [\sum (x - \bar{x}) (y - \bar{y})] / [\sqrt{} (\sum (x - \bar{x})2 \sum (y - \bar{y})2)]$

$rxy = [-112.48] / [\sqrt{} ((22.096) (3201.733))] = -0.4229$

### 2.7.1 Conclusion of Pixel Distribution Analysis

A negative correlation coefficient value ($r_{xy}$) closer to zero shows a negative correlation and a weak relationship between the x and y variables i.e., between Pixel Count at Maximum Surface Temperature ($P_m$) and Maximum Surface Temperature ($T_m$). Therefore, it can be concluded that there is very little or no clear relationship between the maximum surface temperature of a cell and the pixel count at that temperature. As such, we didn't incorporate these results into our training phase of the networks.

## 2.8 Data Augmentation

Data size is an important factor for the effective training of deep neural networks. Deep neural networks consist of a vast number of parameters to learn and a lot of computational power is required to optimize all the parameters. When the dataset is considerably small, the network cannot train well and can result in overfitting.

Overfitting occurs when the network models the training data too well. It learns the details, features and noise in the training data in such a way that it negatively impacts the performance of the model on the testing dataset. A larger dataset can help improve the optimization process

of the network and thus avoid overfitting. The network model can further generalize better as it can capture the inherent data distribution more effectively.

In this study, we are provided with only 15 infrared images of different photo-voltaic cells that were taken in the 1st minute of their operation. This dataset is significantly small and it is the one of the main setbacks in our work. As expected, the proposed neural network models fail to train properly and often results in overfitting.

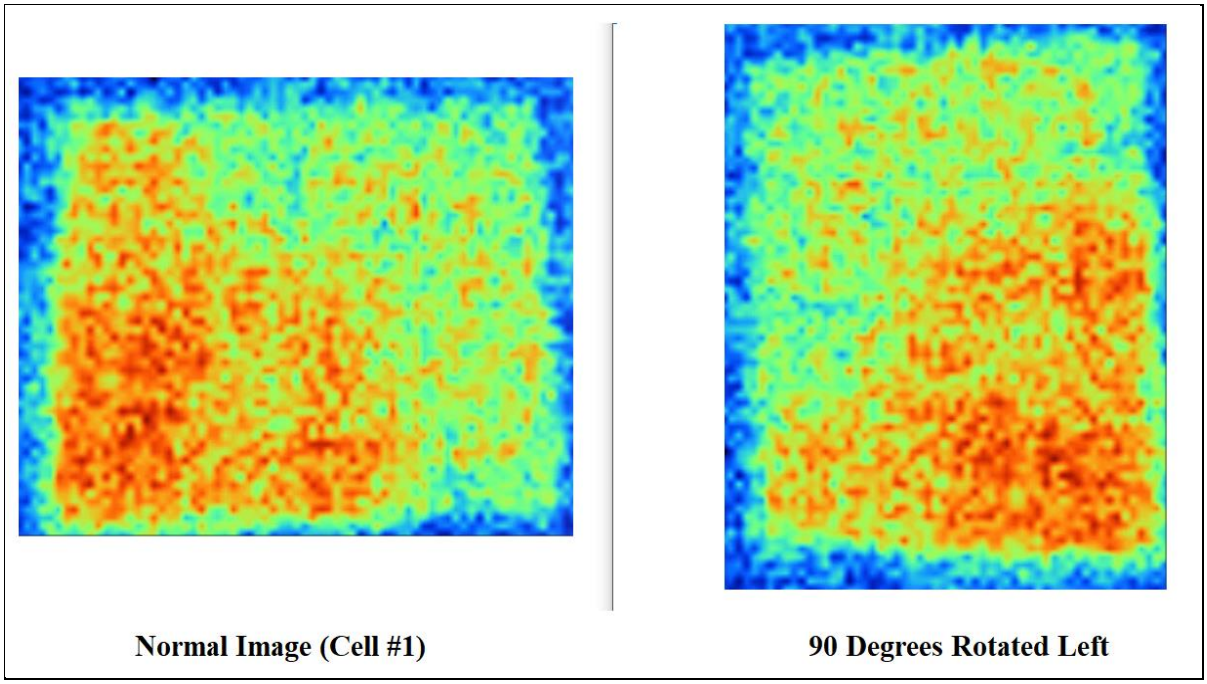
### 2.8.1 Implementation

Data Augmentation is a useful technique that can increase the number of images present in the dataset. Thus, data augmentation can reduce overfitting and allow the network model to generalize better. The following data augmentation techniques have been implemented in our work: -

- Rotation
- Flipping
- Scaling
- Histogram Equalization
- Addition of Gaussian Noise

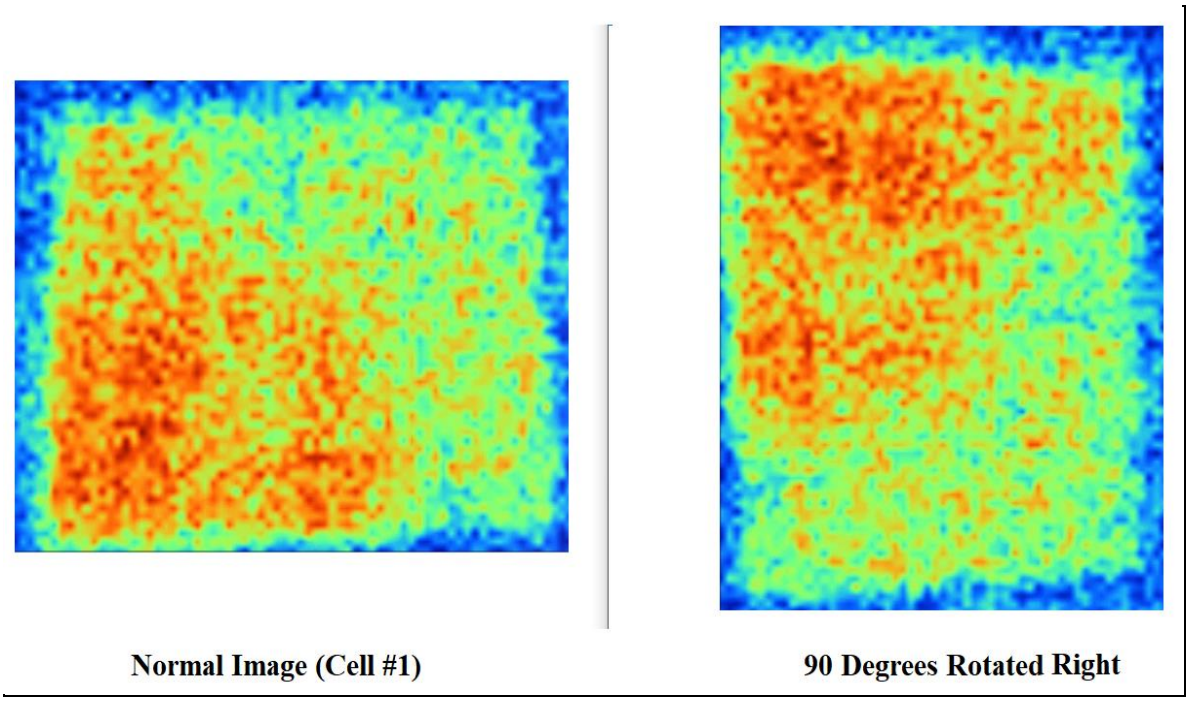
1. Rotation

(i) 90 Degrees Left



**Normal Image (Cell #1)** **90 Degrees Rotated Left**

(ii) 90 Degrees Right



**Normal Image (Cell #1)** **90 Degrees Rotated Right**

(iii) 180 Degrees

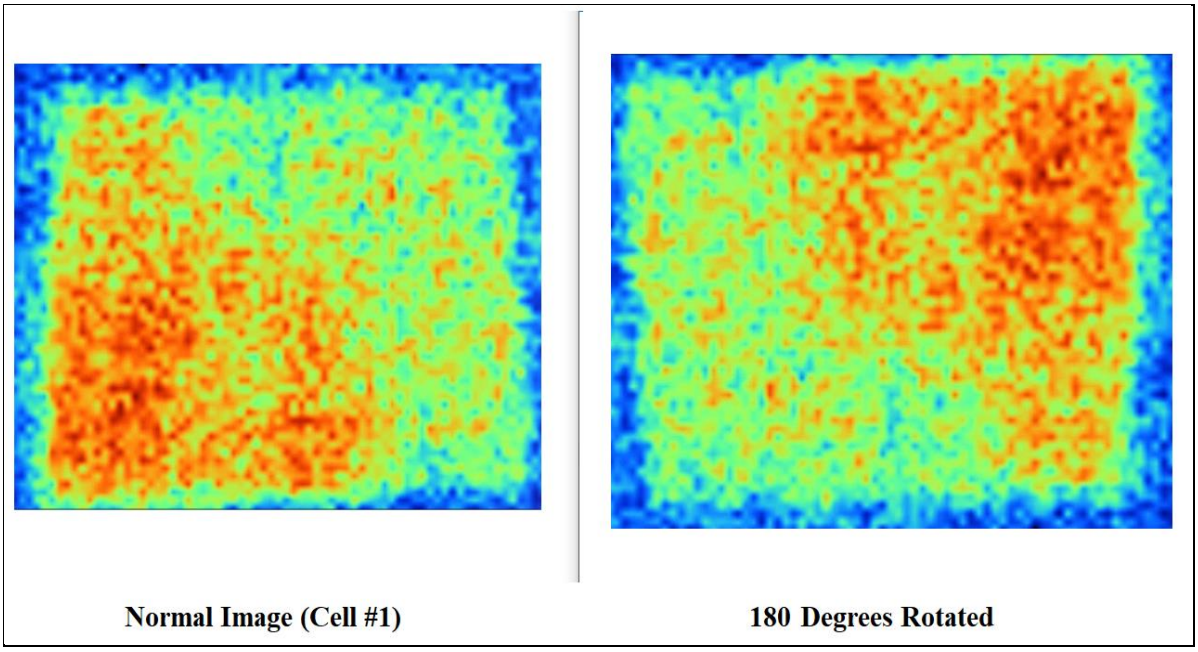
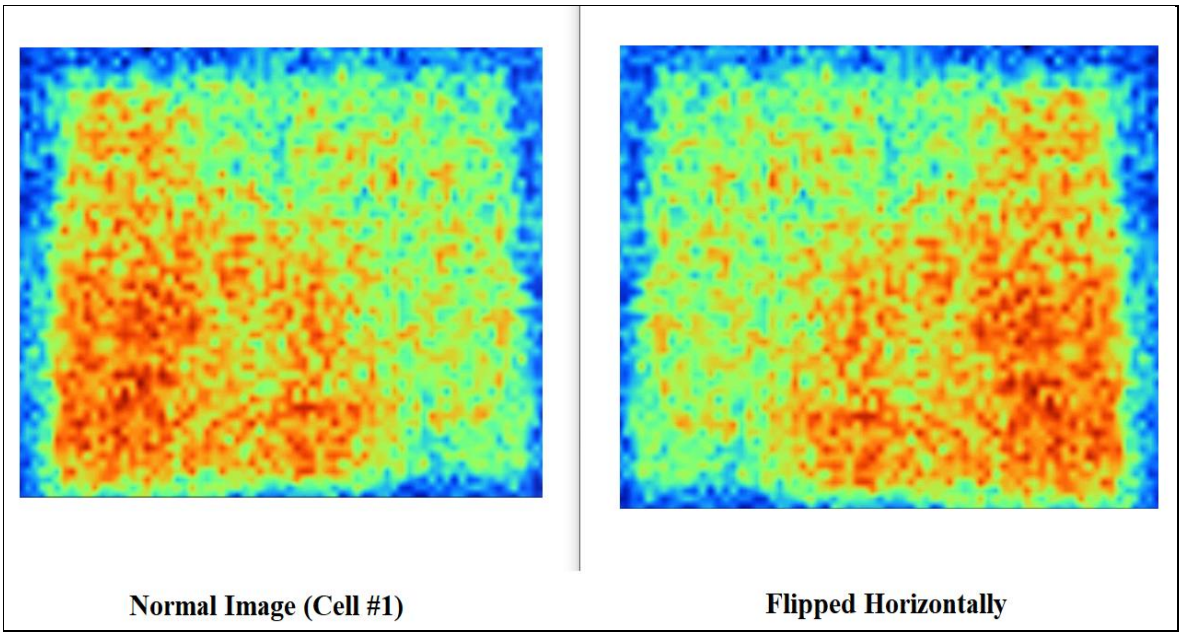



Figure 2.19 Rotation of the IR images; (i) 90 Degree Rotation (ii) 90 Degrees Left Rotation (iii) 180 Degrees Rotation

2. Flipping

(i) Horizonal



(ii) Vertical

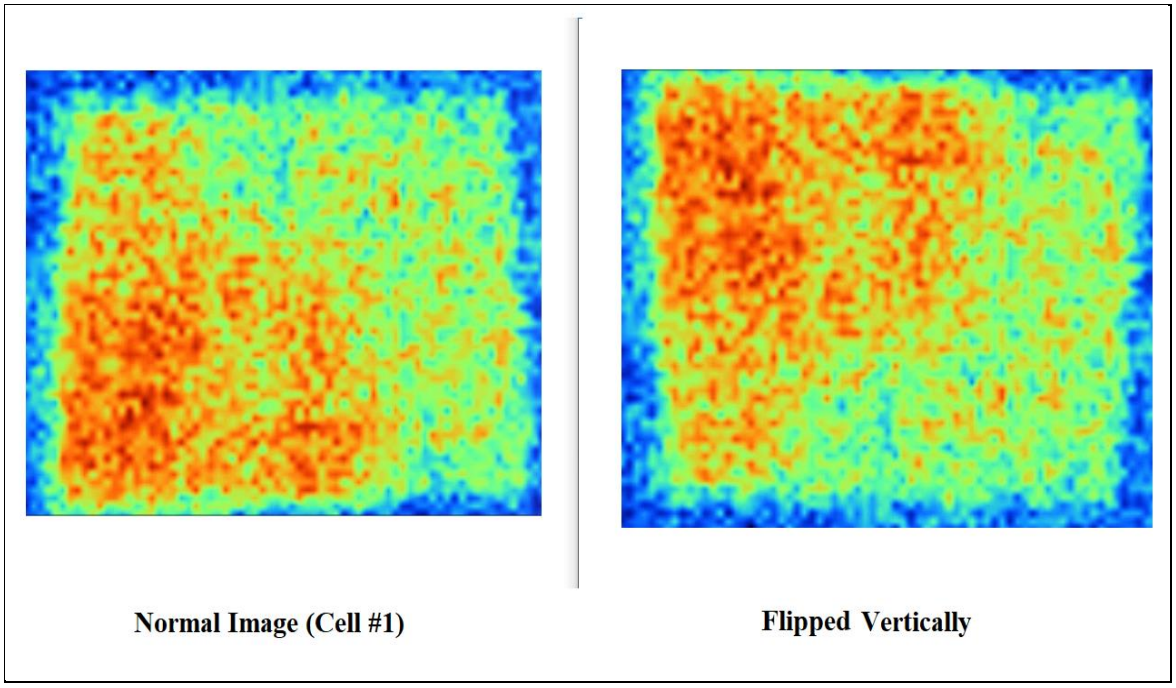



Figure 2.20 Flipping of the IR images; (i) Horizontal Flip (ii) Vertical Flip

3. Scaling (Scale factor 1.25 – Enlargement)

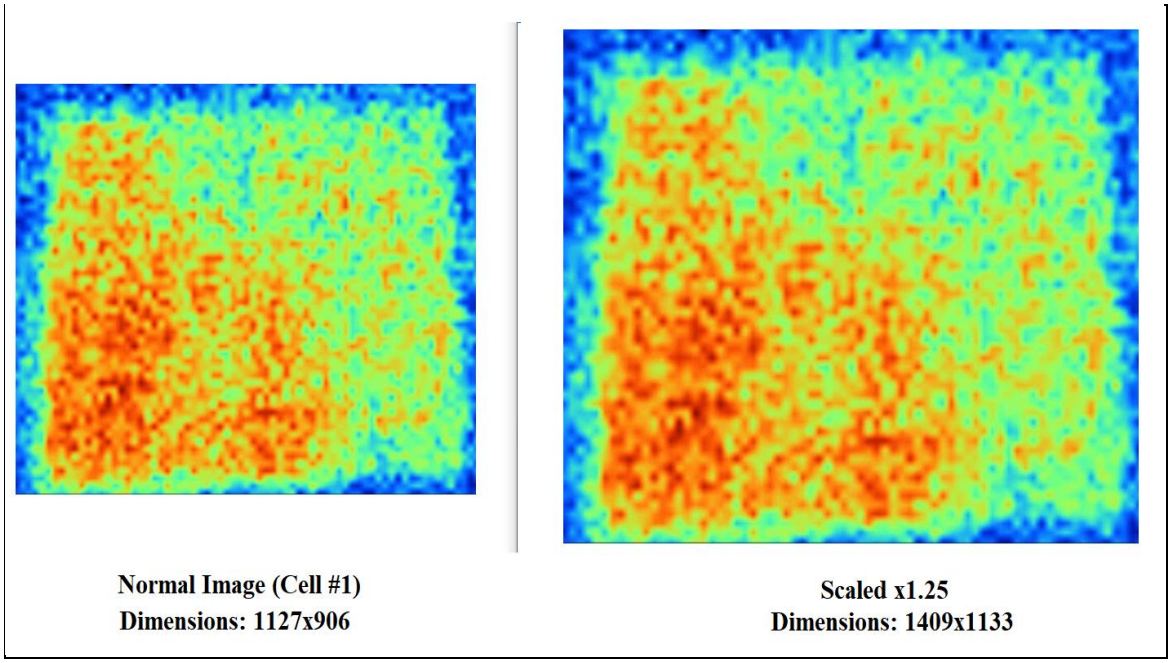



Figure 2.21 Scaling of the IR images; (i) Normal Dimensions (ii) Scaled Dimensions

4. Histogram Equalization (0.3% Saturated Pixels)

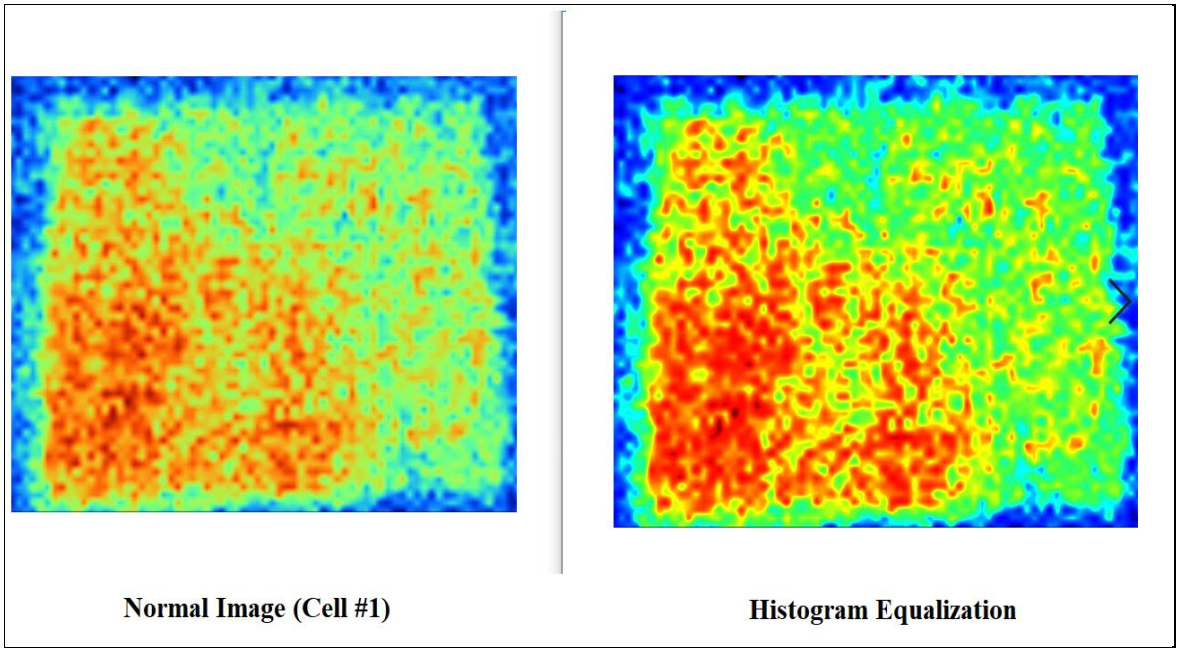



Figure 2.22 Histogram Equalization of the IR images; (i) Normal Image (ii) Histogram Equalization Applied Image

5. Addition of Gaussian Noise (Mean distribution – 0, Standard deviation - 25)

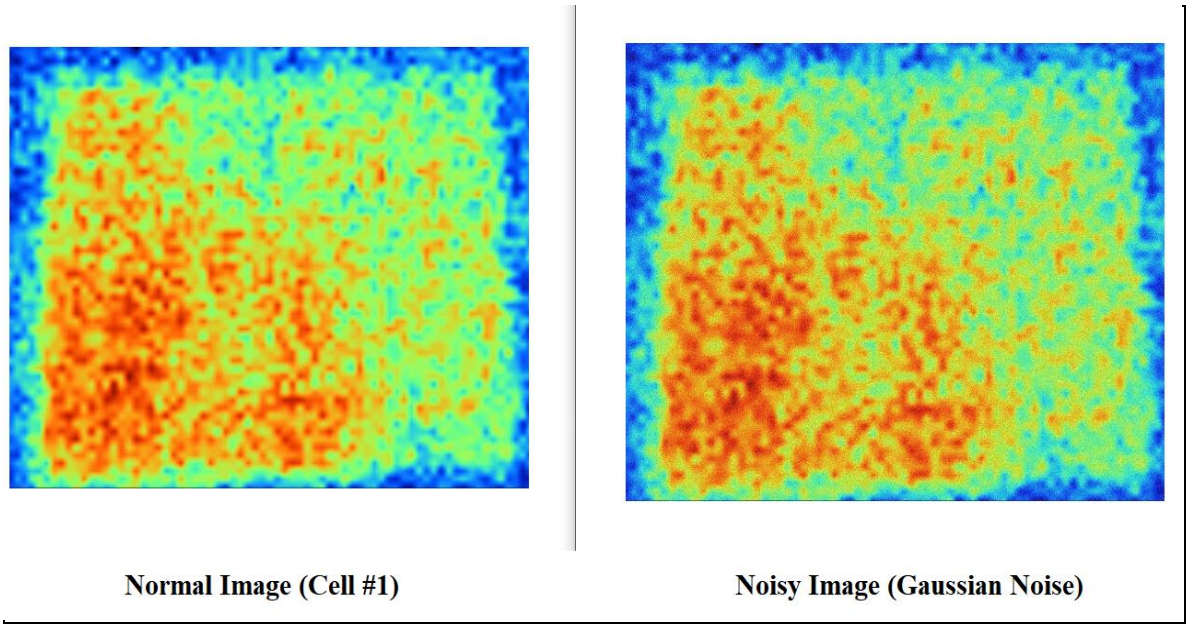



Figure 2.23 Addition of Gaussian Noise; (i) Image without Gaussian Noise (ii) Image with Gaussian Noise

### 2.8.2 Summary

By applying each augmentation technique to 15 of the IR images, the size of the dataset has increased up to 120 images (24 defective and 96 non-defective – based on temperature difference). This may still be fairly small for training a deep neural network, but it is larger than the original dataset and will surely result in an improved training performance of the proposed models. The augmented images were then labelled (for the training of the algorithms) based on the ‘Temperature Difference’ technique and the ‘Thresholding & Contour Detection’ technique and then accordingly split into training, testing and validation sets.

# Chapter 3

# Deep Learning Frameworks

## 3.1 Convolutional Neural Networks (CNN)

Convolutional Neural Network (CNN) is an Artificial Neural Network that is primarily used for image processing or analyzing images. CNN detects an image pattern and uses it to analyze the image. It detects the input image by detecting objects, edges and shapes of that image and differentiate it from one another. CNN is constructed mathematically with 3 types of layers: convolutional, pooling and fully constructed layers. One convolutional layer takes input, transforms the input and sends it to the next layer. The layers have filters which are also known as kernel, which detect various patterns. The filters primarily will detect the more obvious pattern but when the filters are deeper into the layers, the more sophisticated they get so they can detect more detailed and sophisticated parts of the image.

The computer sees an image as an array of pixels and numbers. When an image is given as input, the image is divided into matrix where each element of the row and column consist of one pixel. CNN filters start to process the image from the top left corner using the corresponding matrix format (3*3, 4*4, 9*9 etc.) and makes it into one single pixel value. This process will continue until every single set of pixels are transformed from the given input picture.

### 3.1.1 Convolution Process

CNN is a great algorithm for image classification and data analysis. With lots of training and some pre-processing, this algorithm can classify images with optimum accuracy. It is vastly used in the field of various technological fields and the usage of this is growing exponentially.

It is a very complex yet very effective algorithm. The most prominent CNN algorithms are VGGNet16, MobileNet, ResNets, GoogleNet, LeNet, AlexNet etc. All of these are very successful in image classification and they are being implemented in various tasks and fields successfully.

### 3.1.2 General Architecure of CNN

The CNN architecture is made of number of layers and each layer has its significance. The CNN architecture can be described as follows:

Input - The Input is a matrix of values with a set of height, width and number of channels.

Convolution - This layer is responsible for feature detection. In the layer at first low-level filters are used. When we go deeper into the layer, high level filters are used and high-level feature extraction is performed.

Pooling layer - The pooling layer reduces the size of each feature map which came from the convolution layer. The pooling layer operates on each feature map and creates a new set of maps with the same numbers of pooled feature maps. The size of the pooling filter is much smaller than the size of the feature maps. It is almost always 2x2 pixels applied with a stride of 2 pixels. So that means the pooling layer will always reduce the feature map by a factor of 2.

The pooling operation is of 2 types:

1. Average pooling: Calculating the average value of each patch of the feature maps.
2. Maximum Pooling: Calculating the maximum value of each patch of the feature maps.

Activation Function - This layer returns 0 for every negative value in the input image and returns the same value for every positive image. When the value is zero, the neuron does not

get activated. For example, The ReLu and Softmax are very popular activation functions, which we have employed for our algorithms:

- Rectified Linear Unit (ReLu):

This is the most popular activation function which is used in the hidden layer of the neural network. It is a nonlinear function and performs better than the sigmoid activation function. The first advantage that this activation function has is that it does not activate all the neurons at a time. This layer returns 0 for every negative value in the input image and returns the same value for every positive image. When the value is zero, the neuron does not get activated. It makes the computation process very efficient. Secondly, it is not prone to the vanishing gradient problem unlike Sigmoid and Tanh.

There are some drawbacks for this activation function. Sometimes there might be a negatively saturated region in the ReLU which means that the gradient at that region is zero. It can lead to dead neurons hence; those neurons won't respond to the variations in error or the inputs.

The ReLU Activation Function is illustrated in Figure 3.1.

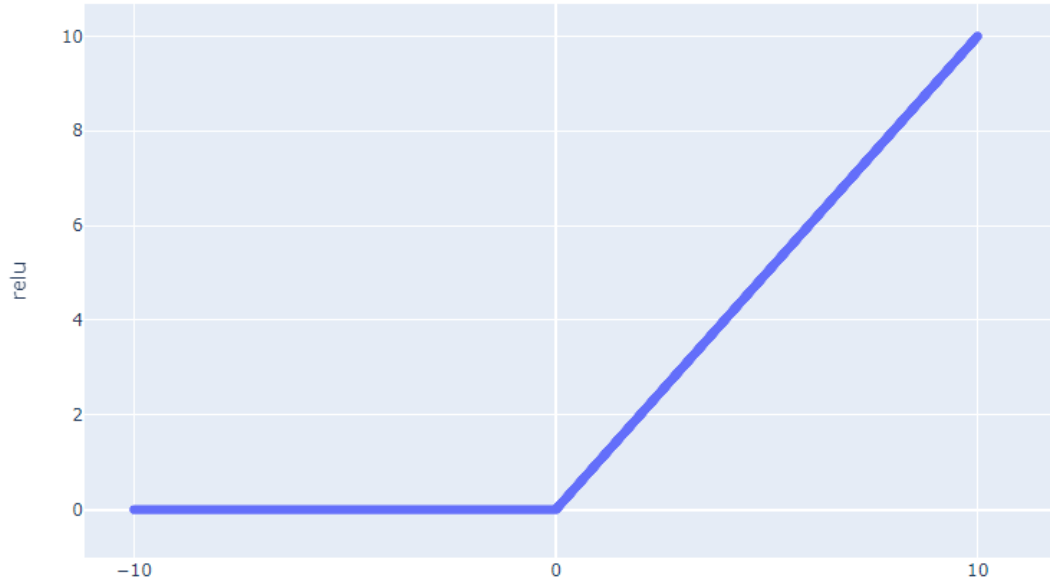



Figure 3.1 – Input vs Output Plot of ReLu Activation Function

- Softmax Output Activation Function:

This function is usually used in the last layer of the neural network which calculates the probability distribution of n number of events. The outputs of this function are vector values that sum up to 1 that also can be interpreted as class membership. This is closely related to the argmax function, which outputs a 1 for the chosen option and a 0 for the options other than the chosen options. It is able to handle multiple classes. The Softmax Activation Function is illustrated in Figure 3.2.

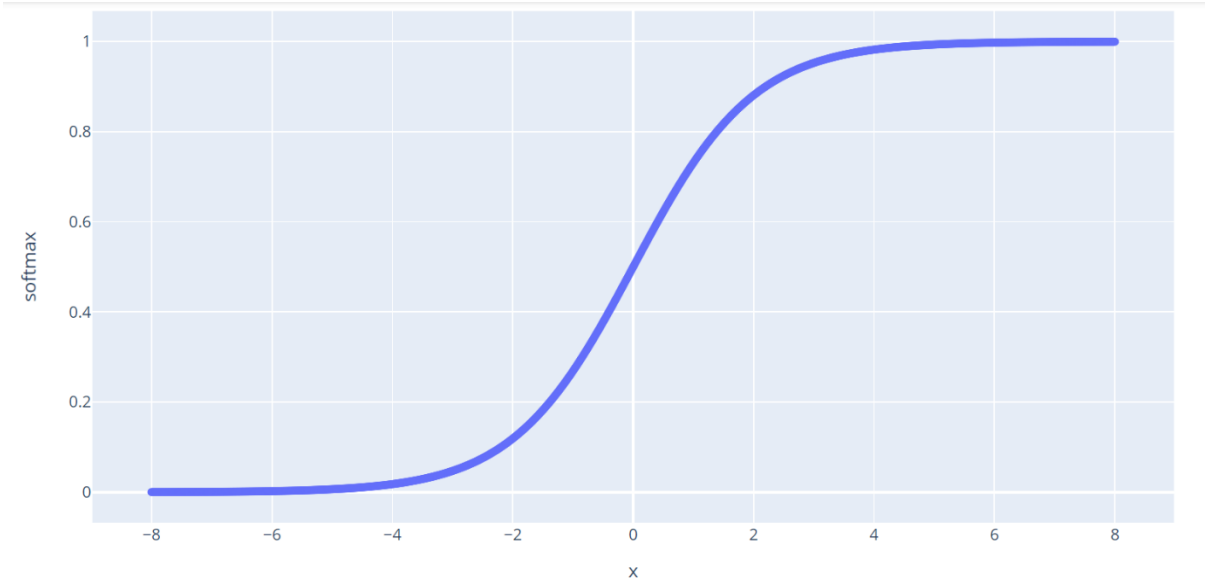



Figure 3.2 – Input vs Output Plot of Softmax Activation Function

Fully Connected layers (FC layer) - In this layer, the neurons establish all the connections with the previous layers. Also, we flatten the output of the last layer and connect them with the next layer. This is the final phase of the CNN network.

### 3.1.3 Learning Parameter Update

The backpropagation algorithm is used to update learnable parameters of the neural network. The loss of model decreases by the backpropagation algorithm, making small changes to each learnable parameter. It updates each parameter in a way which enables it to take a step in the direction along which the loss decreases. The direction is the gradient of the weight. As the gradient keeps moving backwards towards the initial layers, this value keeps getting manipulated by each local gradient. The gradient keeps shrinking which makes the initial update very small which also increases the training period.

## 3.2 Convolutional Neural Networks: VGG16

VGG is the acronym for the lab at Oxford University (Visual Geometry Group) and 16 is the number of layers in the model with trainable parameters. In our research, we have used VGG16

which is a convolutional neural network model proposed by K. Simonyan and A. Zisserman from the University of Oxford in the paper “Very Deep Convolutional Networks for Large-Scale Image Recognition” [44]. VGG16 is a 16-convolution layer architecture which consist of 138 million parameters. The architecture of VGG-16 explains some of the fundamental building blocks of the convolutional network - pooling layers and conv layers.

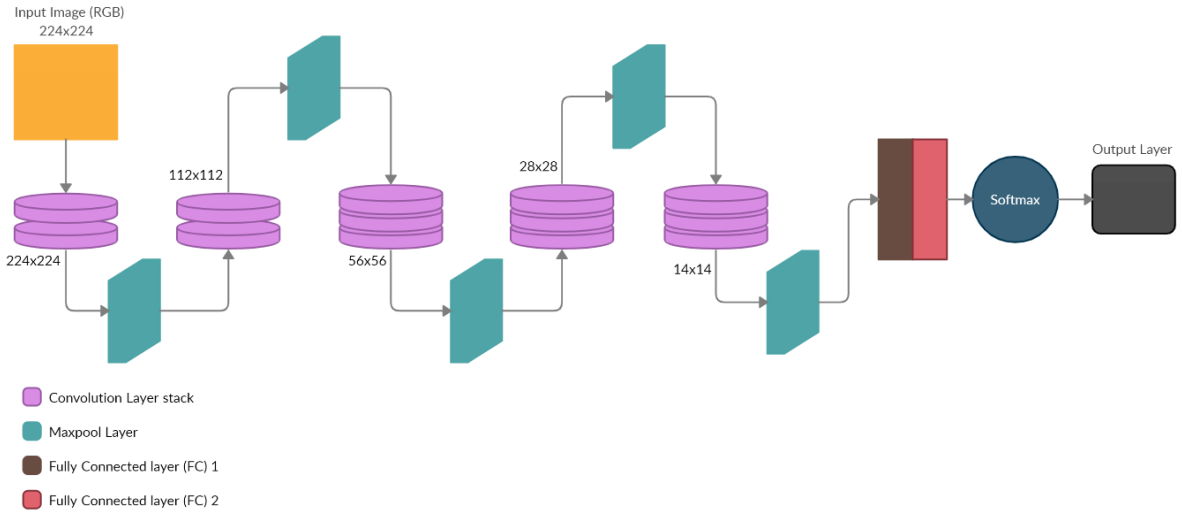



Figure 3.3 Diagram of the architecture of VGG-16 with example Conv Layer and preceding input layer highlighted.

### 3.2.1 VGG16 Architecture

In the algorithm the input image has to be of fixed size of 224x224 RGB image. The image is passed through several stack of convolutional (conv.) layers and filtered within a very small size of 3×3 for the receptive field: (which is the smallest size to capture the notion of left/right, up/down, center for CNN algorithm). In the “Flatten” and “Dense” configuration it utilizes 1×1 convolution filter as a linear transformation of the input channels followed by non-linearity configuration. The convolution stride is fixed to 1 pixel and the spatial padding of conv. layer input is set in such that it will preserve the spatial resolution even after convolution. In our case if the padding is 1-pixel for 3×3 conv. Layers then the spatial pooling is carried out by five

max-pooling layers, which follow some of the conv. Layers (shown in the above figure). The max-pooling is performed over a 2×2 pixel window, with stride 2.

"Flatten" layer is used to convert all the resultant 2D arrays as single long continuous linear vector.

Two "Dense" layers and one "Output Classes" are the three Fully-Connected (FC) layers which follows a stack of convolutional layers. The first two FC has 4096 channels each and the third performs 1000-way ILSVRC classification, thus contains total of 1000 channels where one is for each class. The final layer is the soft-max layer which is used for classification of images. The configuration of the fully connected layers is the same in all networks.

## 3.3 Convolutional Neural Networks: VGG19

Similar to VGG16, VGG19 has the same baseline network, but only with additional layers (19) in the model with trainable parameters. It's a simple model that shares most of the same basic architecture and algorithms as LeNet5, one of the first ConvNets from the 90s. The main difference is the addition of several more layers, which seems to validate the idea that deeper networks are able to learn better representations. VGG-19 is significantly more accurate ConvNet architectures, which not only achieve the great accuracy on ILSVRC classification tasks, but are also applicable to other image recognition datasets, where they achieve excellent performance even when used as a part of a relatively simple pipelines.

### 3.3.1 Architecture of VGG-19

VGG-19 is a 19 layers variation of VGGNet architecture consisting of 16 convolutional layers with 3 fully connected layers and the 5 pooling layers as VGG-16. The VGG-19 consists of two fully connected layers with 4096 channels each and followed by another fully connected layer with 1000 channels for a 1000-way ILSRVC prediction. The last fully connected layer

uses Softmax layer for classification. Figure 3.6 depicts the typical architecture of VGG-19 network.

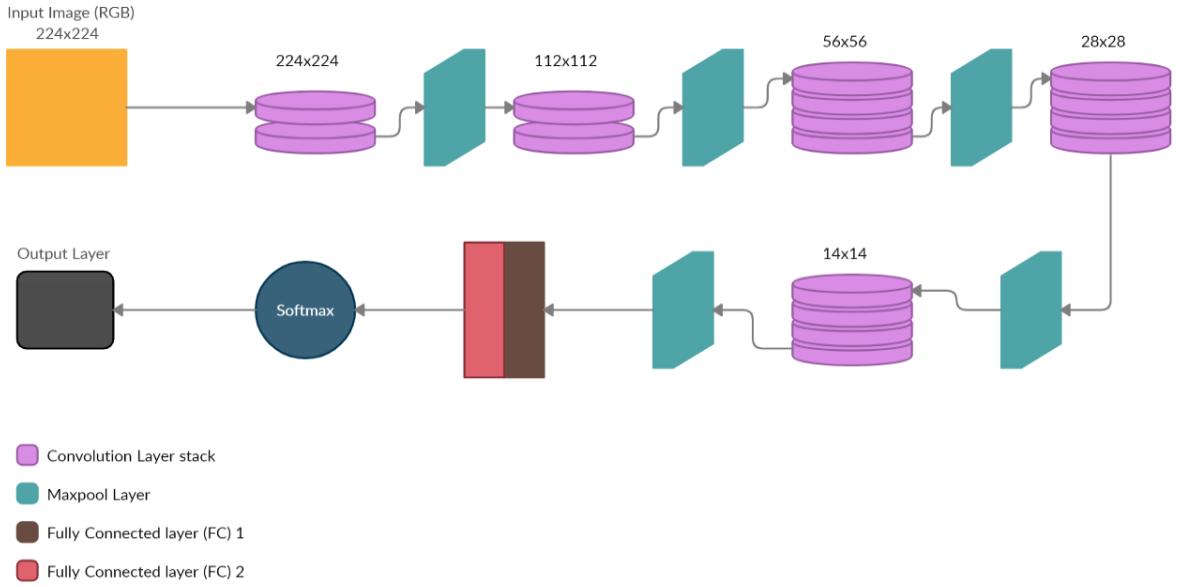



Figure 3.4 Architecture of a typical VGG-19 network

### 3.3.2 Layers and their Functions

VGGNet architecture uses the fix size RGB Image for 224*224 resolution with 64 filter channels. The matrix shape for the image will be (224, 224, 3) where 3 means the color of the image will be RGB. In this layer the mean RGB value is subtracted from each pixel and computed over the training set. The filters in this layer are sized 3*3 with stride of 1 which enables the filters to cover the whole image. To preserve the spatial resolution of the image spatial padding was used. After performing a max pooling over a 2*2 pixels windows with stride 2, the image resolution can be reduced to 112*112 with 64 filters.

After this, the image is followed by 2 more convolution layers consisting of 128 filters and formats the dimension of the image to 112*112*128 from 112*112*64. Then after next pooling layer the image is reduced to 56*56*128. After two more convolution layers with 256 filters and followed by pooling layer the size of the image is again reduced to 28*28*256. After more

stack each with 3 convolution layers are separated by a max-pool layer and followed by a final pooling layer is used the image size converts to 7*7*512.

Table 3.1 Summary of VGG-16 and VGG-19 Architecture Configurations

| **VGG-16** | **VGG-19** |
|---|---|
| 16-weighted layers | 19-weighted layers |
| Input Image (224 x 224 Resolution) | |
| 3 x 3 convolution, depth-64<br>3 x 3 convolution, depth-64 | 3 x 3 convolution, depth-64<br>3 x 3 convolution, depth-64 |
| Maxpool Layer | |
| 3 x 3 convolution, depth-128<br>3 x 3 convolution, depth-128 | 3 x 3 convolution, depth-128<br>3 x 3 convolution, depth-128 |
| Maxpool Layer | |
| 3 x 3 convolution, depth-256<br>3 x 3 convolution, depth-256<br>3 x 3 convolution, depth-256 | 3 x 3 convolution, depth-256<br>3 x 3 convolution, depth-256<br>3 x 3 convolution, depth-256<br>3 x 3 convolution, depth-256 |
| Maxpool Layer | |
| 3 x 3 convolution, depth-512<br>3 x 3 convolution, depth-512<br>3 x 3 convolution, depth-512 | 3 x 3 convolution, depth-512<br>3 x 3 convolution, depth-512<br>3 x 3 convolution, depth-512<br>3 x 3 convolution, depth-512 |
| Maxpool Layer | |
| 3 x 3 convolution, depth-512<br>3 x 3 convolution, depth-512<br>3 x 3 convolution, depth-512 | 3 x 3 convolution, depth-512<br>3 x 3 convolution, depth-512<br>3 x 3 convolution, depth-512<br>3 x 3 convolution, depth-512 |
| Maxpool Layer | |
| FC-4096 Fully-Connected Layer | |
| FC-4096 Fully-Connected Layer | |
| FC-2 Fully-Connected Layer | |
| Soft-max Output | |

After this, the image is followed by 2 more convolution layers consisting of 128 filters and formats the dimension of the image to 112*112*128 from 112*112*64. Then after next pooling layer the image is reduced to 56*56*128. After two more convolution layers with 256 filters

and followed by pooling layer the size of the image is again reduced to 28*28*256. After more stack each with 3 convolution layers are separated by a max-pool layer and followed by a final pooling layer is used the image size converts to 7*7*512. Then three fully connected layers are implemented from which first two with 4096 channels and followed by a layer with 1000 channels for 1000-way ILSVRC classification and the final layer is a Softmax function which is used for multi-classification. This is represented in Table 3.1.

In our study, we have employed VGG-16 and VGG-19 typical networks for a binary image classification task.

### 3.3.3 Modern Applications

VGG 19 is applied in object detection or fault classification of various images. It can go through thousands of images at a time and can focus on the specific parts in them. VGG19 also can be pre-trained so that means it has sets of data for making the detection process smoother. It is used to detect objects, human faces, body parts, animals: almost anything and everything if its programmed or trained in the proper way. it is capable of distinguishing various objects if it is trained with the proper sets of data. The VGG series is vast and each of them has its specialties, so the application of VGG also depends on the type of VGG algorithm. Some of them are as follows:

*Human face detection* - the use of VGG in detecting faces through cameras, especially security cameras, is a very widely popular idea at the moment. It becomes very efficient to detect suspicious activity or events. It is also useful in detecting faces while wearing a mask and now it is much needed since now, we are in the middle of a global pandemic. It is extremely efficient as it can distinguish between human and other objects and thus, making face detection really easy.

*Fault detection in PV panels* - VGG can be used in fault detection in PV panels. Since the human eye cannot detect a faulty PC cell, using deep learning algorithms make the detection process more accurate. VGG is a great algorithm for that purpose. A faulty solar cell will be generating different types of heat spots and thus the VGG can detect those faulty cells. This can mainly be achieved by properly training the VGG model with lots of cell images and data sets regarding the images.

*Application of VGG in the medical field* - The concept of detecting cancer cells or the detection of eye diseases by using VGG algorithm was being introduced in recent years. It has shown quite promising results and now it is going to be implemented very soon in detecting cells. This will be revolutionary if it can be implemented successfully.

## 3.4 Convolutional Neural Networks: EfficientNet

EfficientNet is a state-of-the-art convolutional neural network (CNN) model, initially introduced by Tan and Le in [45], which employs a scaling method in order to efficiently improve dimensional depth, width and resolution of the neural network using a compound scaling method using a parameter called *compound coefficient ($\phi$)*. Standard implementations arbitrarily adjust and scales the aforementioned factors, while EfficientNet emphasizes on uniform scaling of the network depth, width and resolution in accordance with a set of pre-defined scaling coefficients. The model has manifested outstanding accuracy in classification tasks and has proven to be superior to many state-of-the-art CNNs. The model adopts a computation method that incorporates a versatile improvement of the efficiency of the model where the parameters and FLOPS (Floating Point Operations Per Second) are significantly minimized in contrast to other renowned CNN models.

### 3.4.1 Model Scaling

Model Scaling refers to the determination of an ideal model size for a particular classification task, one that would provide the best possible accuracy for the designated task. To be more specific, it involves the adjustment of one of the following dimensions - the model's depth, width or the input image resolution, in order to determine the ideal set of dimensions for the network to perform efficiently.

EfficientNet employs a standard baseline network and implements compound scaling method in order to strengthen the networks predictive capability by working with the baseline network's basal convolutional operations and network structure. Some of the mainstream CNN architectures that are employed as a baseline network or model includes MobileNet, GoogleNet and ResNet, but not limited to these. One of the most fundamental architectures of EfficientNet is termed EfficientNet-B0, that constitutes MBConv as the primary building block of the network. The general principle is to employ such a baseline network and then modify the different dimensions of the network in an attempt to improve the accuracy and efficiency of the network. Dimensions which can revamped are elaborated below –

Adjusting *Model Depth*: The depth of a network signifies the number of layers in a network. A model with a large number of convolutional layers signifies a much deeper network, one that is capable of learning and retaining more finer details of the image and is therefore more accurate, as opposed to a model that constitutes a comparatively smaller number of layers. Therefore, by increasing the depth, by incorporating more convolutional layer, more sophisticated features of an image can be studied and learned. However, a deeper network involves more time-consuming computations and is therefore slower than a shallow network (a smaller number of layers).

A deeper network usually comes with few constraints: -

- They can be difficult to train due to vanishing gradient problem when training with gradient-based learning methods. In such instances, gradient value becomes small and insignificant, inhibiting the weight update. And in many severe instances, it could lead to the stoppage of further training.
- As depth is significantly increased, the increase in accuracy saturates at a certain point.

Considering these constraints, the depth of the model scaled up and adjusted accordingly.

Adjusting *Model Width*: The width of a network corresponds to the number of filters or channels within a convolutional layer. Similar to the different channels (R, G, B) that a typical image constitutes, convolutional neural network layers also constitute multiple channels. A wider network is one that has a larger number of channels in each layer. The width of the networks can be scaled accordingly in order to capture higher-level features. A wider and shallow network are quite easy to train, however, they come with few constraints as-well: -

- Networks that are made to be immensely wide and shallow can be quite inefficient in terms of studying high-level features.
- As width is significantly increased, the increase in accuracy saturates at a certain point.

Hence, considering such constraints, the model width can be scaled up and adjusted accordingly in order to construct a well performing model.

Adjusting *Input Image Resolution*: The resolution of an image is its corresponding height and width. Generally, convolutional neural networks employ input images that are usually fixed. An image with a resolution of 512x512 constitute more information as opposed to a 224x224 image. Consequently, the architecture of a model can be changed accordingly in order to feed it with an image that has a high resolution, which would yield better performance and accuracy.

Needless to say, an increase in accuracy signifies a required increase in processing power due to the of large amount of information of a larger image that needs to be processed. Similar to the other parameters, the increase in accuracy tends to saturate after a certain image resolution.

All things considered, it was observed that scaling up the network width, depth, or resolution of the network can significantly increase accuracy, but the increase in accuracy starts to saturate for larger values of the coefficients. Henceforth, in order construct a well performing model, in terms of efficiency and accuracy, balancing and uniform scaling of all dimensions of network width, depth, and resolution during scaling is essential. However, deciding on the depth, width, and resolution for a network is not very apparent and requires searching over all the 3 dimensions to have them scaled up uniformly. Henceforth, it can be concluded that, in order to improve the accuracy of the network, contributions from all three dimensions must be integrated.

### 3.4.2 Compound Scaling

The compound scaling method is a scaling technique put forward by the authors of the introductory paper, a technique that uses a *compound coefficient ($\phi$)* in order to uniformly scale the network width, depth and resolution and also maintain a balance between the dimensions.

*Compound coefficient ($\phi$)* is a user-defined parameter, a global scaling factor, which is used to determine the size of resources that can be allocated for the network width, depth and resolution. The parameters $\alpha$, $\beta$, and $\gamma$ determines how the allocated resources can be distributed to the width, depth and resolution of the network. Doubling the depth doubles the FLOPS, doubling width or resolution will increase FLOPS by a times four; which indicates that FLOPS of a convolutional operation is proportional to d, $w^2$ and $r^2$. Hence, using a compound coefficient to scale the dimensions will increase the total FLOPS by $(\alpha * \beta^2 * \gamma^2)^{\phi}$. As such, in order ensure that FLOPS do not exceed $2^{\phi}$, the condition $(\alpha * \beta^2 * \gamma^2) \approx 2$ is applied. Hence, if

compound coefficient is set to 1 and we have twice the resources available, FLOPS get scaled by a factor of $2^1$.

The determination of the parameters $\alpha$, $\beta$, and $\gamma$ is done by employing grid search with $\phi$ set to 1, which then determines the best set of values that results in highest accuracy. Once the ideal values for these parameters pertaining to the network is determined, they are then fixed, and the value of $\phi$ can be further increased to build larger but more accurate models. The values of the parameters $\alpha$, $\beta$, and $\gamma$ are not re-evaluated after every scale up of $\phi$ as it is computationally expensive to carry out repeated grid search. In this manner, many different versions of the EfficientNet model have been constructed, EfficientNet-B1 to EfficientNet-B7, where the integer in the name signifies the value of the compound coefficient $\phi$. In these models, EfficienNet-B0 is set as the baseline network, using which the values of $\alpha$, $\beta$, and $\gamma$ were determined and fixed for other models of the family. This technique of model scaling can be applied to any CNN architecture in order to attain an improved accuracy.

In our work, we have employed models B5-B7 within the EfficientNet family in order to analyze their predictive capability on our dataset.

# Chapter 4

# Result Analysis and Discussion

In our study, we have employed the use of solar cell images obtained through Infrared Thermography in order to inspect their respective thermal mappings with the help of Convolutional Neural Networks (CNN) and use these algorithms to make prediction between defective and normal modules. Prominent CNN models such as VGG16, VGG19 and EfficientNet were employed in order to be trained and tested on these Infrared Images.

## 4.1 Performance Analysis – VGG-16 and VGG-19

Since the IR images have been classified according to two different image processing approaches, temperature difference (ΔT) and thresholding; both the VGG models have been trained and validated in terms of both type of approaches. As mentioned before, our research originally involves only 15 infrared images of solar cells and according to the ΔT analysis, 12 of the images are classified as normal and the other 3 as defective. On the other hand, according to the thresholding approach, 10 of the images are classified as normal and the other 5 are classified as defective.

In order to train the VGG models using the temperature difference appraoch, these 15 images are divided into 3 sets of data: training set (5 images), validation set (5 images) and testing set (5 images); with each set containing the same number of defective and normal images (4 normal and 1 defective). Both VGG16 and VGG19 models were able to achieve an overall accuracy of 80 percent, which is reasonable considering the small size of the dataset. However, VGG19 achieved the performance with over half the number of epochs that was required for VGG16.

Moreover, in order to increase the size of the dataset and improve the performance of the models, data augmentation techniques such as rotation, flipping, histogram equalization, noise

and scaling are applied on the images. As a result, 120 additional IR images are obtained, which consists of 24 defective cells and 96 normal cells. For training, the total of 135 IR images are divided into 3 sets of data: training set (45 images), validation set (45 images) and testing set (45 images); with each set containing the same number of defective and normal images (36 normal and 9 defective). Comparing the performances of the VGG16 and VGG19 models on the augmented data sets, the VGG16 model had an improved overall accuracy of 93.75% while the VGG19 model had no changes in the overall accuracy.

Subsequently, for training the VGG models using the thresholding and contour detection appraoch, the original 15 images are also divided into 3 sets of data: training set containing 6 images (4 normal and 2 defective), validation set containing 5 images (3 normal and 2 defective) and testing set containg 4 images (3 normal and 1 defective). In this case, the VGG16 model outperformed the VGG19 model with the highest overall accuracy of 100%, which is also greater than the accuracy values in the temperature difference approach; the VGG19 model had the lowest overall accuracy of 60% in this particular approach. A 100 percent accuracy cannot be proclaimed reasonable here, since the validation dataset is too small. Hence, correctly assessing the predictive accuracy was not viable. The VGG16 network might be coincidentally achieving such a high overall accuracy and hence to reach an authentic conclusion, data augmentation techniques are applied to increase the dataset and compute the new overall accuracy. However, we can conclude that the model does depict a good performance.

Therefore, data augmentation techniques similar to the ones in the first approach were applied to increase the size of the dataset and improve the performance of the models. Again, 120 additional IR images are obtained, consisting of 24 defective cells and 96 normal cells. Similarly, the entire dataset of 135 IR images are again divided into 3 sets of data: training set containing 54 images (36 normal and 18 defective), validation set containing 45 images (27 normal and 18 defective) and testing set containing 36 images (27 normal and 9 defective).

From the results of the training, it can be seen that the overall accuracy of VGG16 came out to be 100% again while the overall accuracy of VGG19 improved to 84.44 percent. Hence, we can conclude that the VGG16 model performed better than VGG19 model in all the approaches while having achieved the highest accuracy of 100% with the thresholding and contour detection approach.

Table 4.1 summarizes the learning performances of both the VGG models on the original dataset and the augmented dataset.

Table 4.1 Summary of performances of the VGG models

| ***Model*** | ***Learning Rate*** | ***Epochs*** | ***Performance accuracy with Temperature Difference approach*** | ***Performance Accuracy with Thresholding Approach*** | ***Performance accuracy with Temperature Difference approach (with Data Augmentation)*** | ***Performance accuracy with Thresholding approach (with Data Augmentation*** |
|---|---|---|---|---|---|---|
| *VGG16* | 0.0001 | 100 | 80% | 100% | 93.75% | 100% |
| *VGG19* | 0.0001 | 50 | 80% | 60% | 80% | 84.44% |

## 4.2 Performance Analysis – EfficientNet

In addition to the VGG architectures, we have also utilized 3 different EfficientNet model architectures, EfficientNet B5, B6, B7. Some of the optimized parameters such as the number of width, depth and resolution has been shown in Table 4.2. These three models have been trained using both the classification approaches, temperature difference analysis and thresholding. The same dataset that was used for the VGG models has also been used for

training the EfficientNet models and both the original and augmented datasets have been split in the exact same manner. In temperature difference approach, 12 IR images are classified as normal and 3 are classified as defective. In thresholding and contour detection approach, 10 IR images are classified as normal and 5 are classified as defective.

Initially, using the temperature difference approach, all the three models achieved the same performance accuracy of 80%. Moreover, incorporating augmented datasets that were used for the VGG models, the accuracy remained at 80% for all the three models. Hence for this approach, data augmentation did not provide any consistent improvement in the learning of the network models.

Next, using the thresholding and contour detection approach, EfficientNet models B5 and B6 had overall accuracy values of only 40% while EfficientNetB7 had a higher overall accuracy of 60%. This is much lower when compared with the learning performance in the temperature difference approach. With the augmented images (135), the performance of the EfficientNet B5 and B6 improved up to 60% while no improvement was recorded for EfficientNetB7. This is still much lower than the learning performances observed in the temperature difference approach. Therefore, it can be concluded that the EfficientNet models performed much better with the temperature difference approach and data augmentation had very little improvement in their learning performances. A much larger dataset would have helped us assess the performances better. Table 4.2 summarizes the learning performances of the EfficientNet models.

Table 4.2 Summary of the learning performances of EfficientNet-B5, B6 and B7

| ***Efficient Net Model*** | ***Width, Depth, Resol-ution*** | ***Epochs*** | ***Performance accuracy with Temperature Difference approach*** | ***Performance Accuracy with Thresholdi-ng Approach*** | ***Performance accuracy with Temperature Difference approach (Data Augmentati-on)*** | ***Performance accuracy with Thresholdi-ng approach (Data Augmentati-on)*** |
|---|---|---|---|---|---|---|
| *Efficient Net-B5* | (1.6, 2.2, 456) | 50 | 80% | 40% | 80% | 60% |
| *Efficient Net-B6* | (1.8, 2.6, 528) | 50 | 80% | 40% | 80% | 60% |
| *Efficient Net-B7* | (2.0, 3.1, 600) | 50 | 80% | 60% | 80% | 60% |

## 4.3 Performance Comparison of best performing models

Finally, Table 4.3.1 sums up the performances of the deep learning models in terms of both the approach of the classification and the size of the dataset.

Table 4.3 Overall performance comparison of VGG-16, VGG-19 and EfficientNet models

| Deep Learning Model | VGG16 | VGG19 | EfficientNetB5 | EfficientNetB6 | EfficientNetB7 |
|---|---|---|---|---|---|
| Highest Accuracy using Temperature Difference Approach | 93.75% | 80.0% | 80.0% | 80.0% | 80.0% |
| Highest Accuracy using Thresholding Approach | 100% | 84.44% | 60% | 60% | 60% |
| Highest Accuracy with normal dataset (15 IR images) | 100% | 80% | 80% | 80% | 80% |
| Highest Accuracy with augmented dataset (135 images) | 100% | 84.44% | 80% | 80% | 80% |

The most obvious conclusion we can draw from the table is that VGG16 had the best overall accuracy of 100%, while VGG19 had the second-best overall accuracy of 84.44% and EfficientNet had the lowest overall accuracy of 80%. This is also illustrated in Figure 4.1.

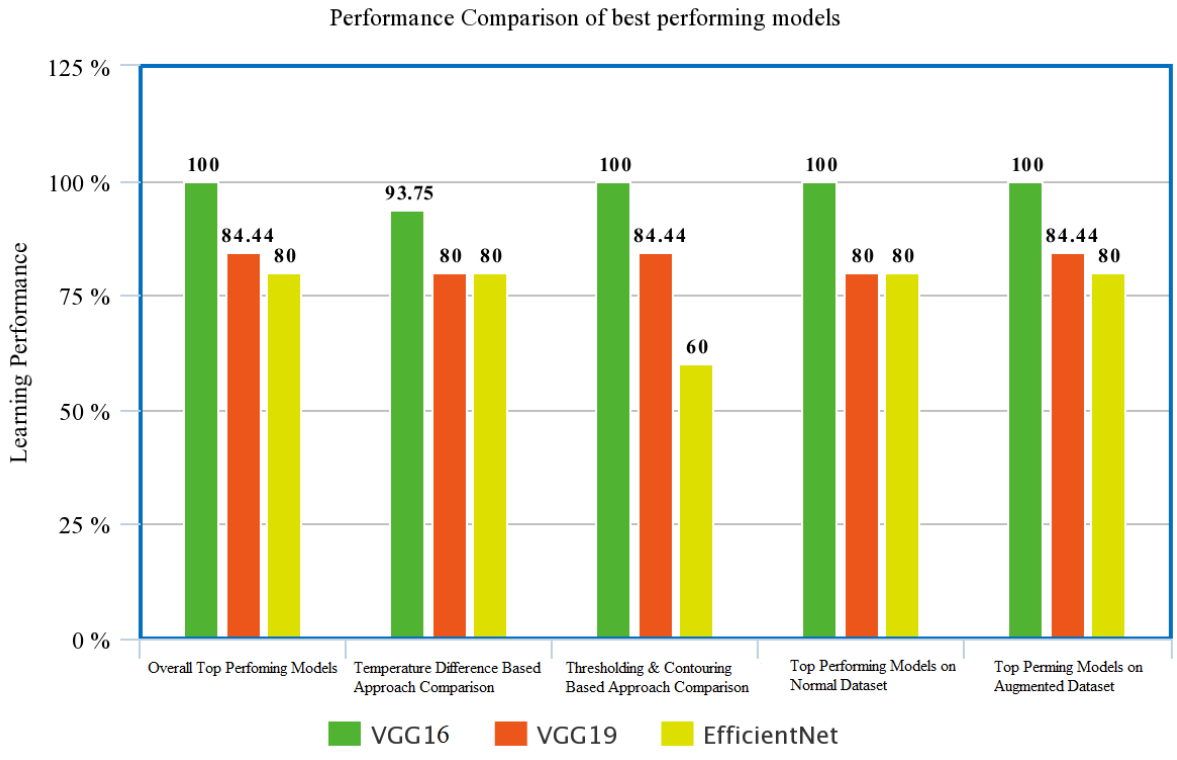



Figure 4.1 Performance comparison of best performing models

Moreover, we can also deduce that the VGG models performed better using thresholding approach when compared with the temperature difference approach while the EfficientNet models performed better using the temperature difference approach when compared with the thresholding approach.

Lastly, data augmentation helped improved the performances of the VGG models to a small extent but had very small to no impact on the performances of the EfficientNet models.

The inconsistent performance results in some of the models can be justified by pointing out the lack of dataset, which created an imbalance during the splitting of the dataset into training, validating and testing dataset. It was observed that in some instances, the model had only 1 defective image and 4 normal images to train on. Similar issue was experienced in the validation datasets as-well. Thus, the models had very small number of images to train and validate on.

## 4.4 Training Performance Overview

This section provides an illustration of some of the training evaluations pertaining to VGG-16 VGG-19 and EfficientNet, with the help of plots. The accuracy of a model signifies the extent to which a model was able to make correct predicts. In addition, the loss of model signifies how deviated the model is away from the true labels. A greater loss indicates that the model is experiencing difficulties in making the right set of predictions (with reference to the training and validation dataset) and a reduction in loss indicates that the model is gradually progressing closer to making more accurate predictions on the datasets. Moreover, the number of epochs within the plots refer to the number of times the model has entirely processed the training dataset (one epoch = one forward pass and one backward pass).

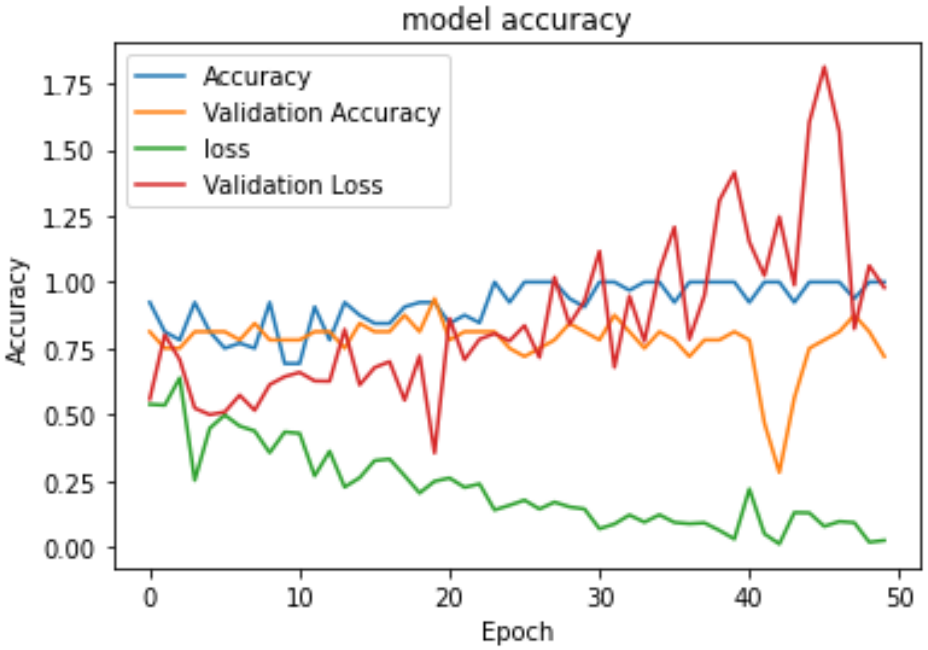



Figure 4.2 Training and Validation Performance of VGG-16 (Temperature Difference approach with Data Augmentation)

Figure 4.2 shows that training accuracy and loss has increased and decreased respectively over an increase in epochs, for VGG-16 model trained on the IR augmneted dataset, that was

labelled using Temperature Difference approach. The training was halted early due to no further increase in validation accuracy. Due to lack of extensive validation dataset, a significant amount of validation loss was observed due to overfitting. A validation accuracy of 93.75 percent was achived in this case as opposed to the model that trained on unaugmented dataset, which had delivered a validation accuracy of 80 percent.

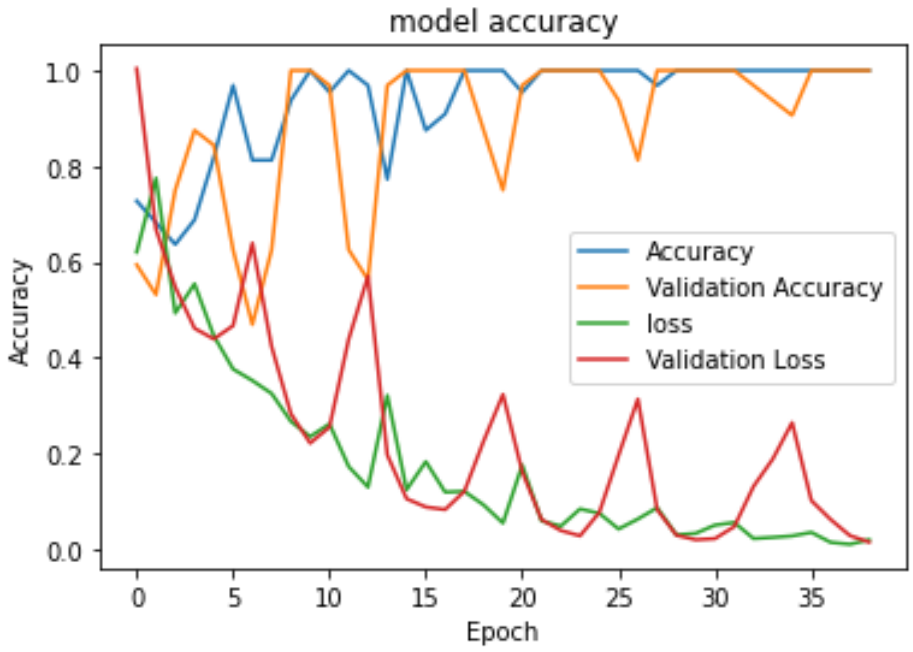



Figure 4.3 Training and Validation Performance of VGG-16 (Thresholding and Contouring approach with Data Augmentation)

Figure 4.3 shows the extensive decrease in validation loss and training loss (over an increase in epochs) on the IR Image augmented dataset that was labelled using Thresholding and Contouring. We can also observe a very high validation and training accuracy. A validation accracy of 100 percent was achived in this case, which is not an optimum conclusion considering the small size of the validation dataset. However, we can conclude that the model performed really well considering an extensive decrease in both validation and training loss over an increase in number of epochs.

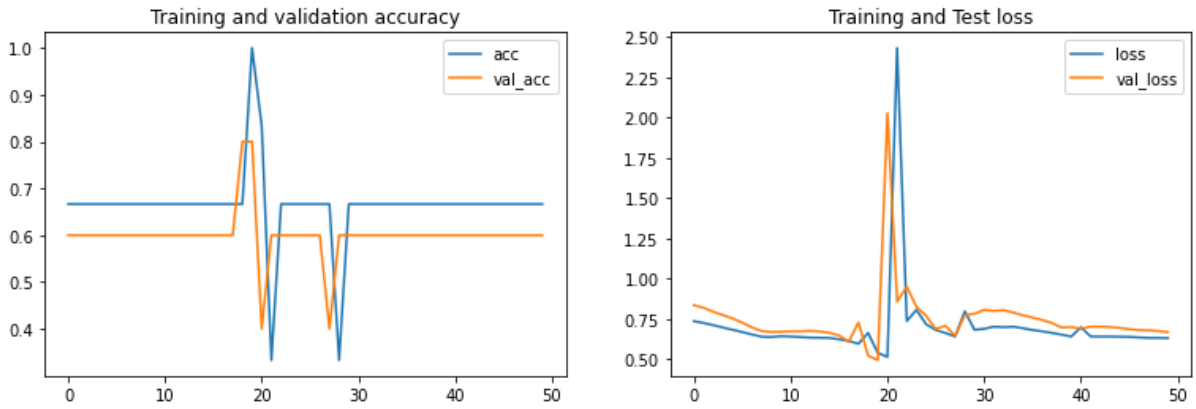



Figure 4.4 Training and Validation Performance of VGG-19 (Thresholding and Contouring Approach)

Figure 4.4 shows the accuracy and loss performance of the VGG-19 model over number of epochs, which was trained and validated on the IR image dataset that was labelled using Thresholding and Contouring approach. It can be noted that there was a signifcant decrease in training and validation (test) loss as number of epochs increased. It also had a good training and validation accuracy. A validation accuracy of 60 percent was achieved for the model on the unaugmented IR image dataset.

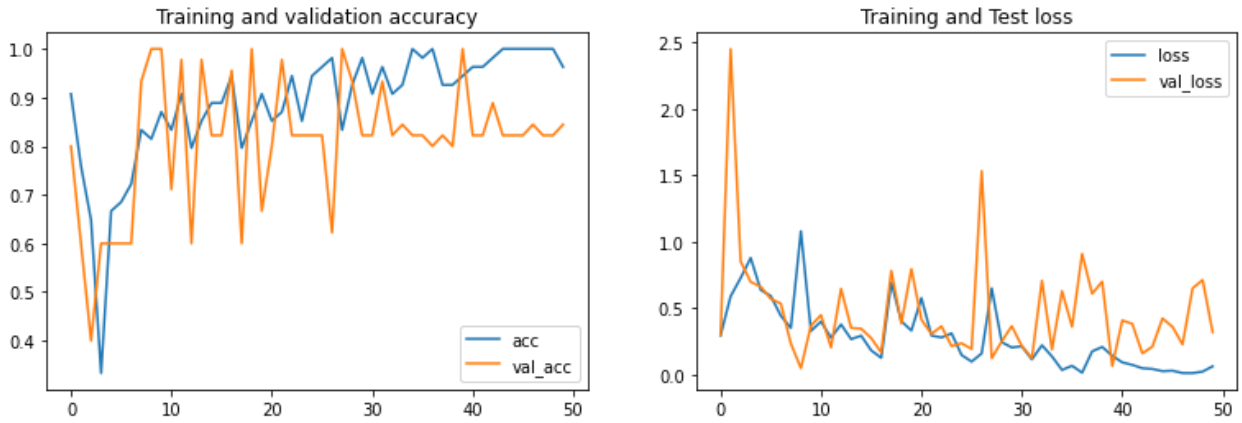



Fig 4.5 Training and Validation Performance of VGG-19 (Thresholding and Contouring Approach with Data Augmentation)

Figure 4.5 shows the accuracy and loss performance of the VGG-19 model over number of epochs, which was trained and validated on the augmented dataset that was labelled using Thresholding and Contouring approach. It can be noted that there was a signifcant decrease in training and validation (test) loss as number of epochs increased. It also had a good training and validation accuracy. A validation accuracy of 80 percent was achieved using this method, which is a better result as opposed to the model that trained on unaugmented dataset that was labelled using Thresholding and Contouring.

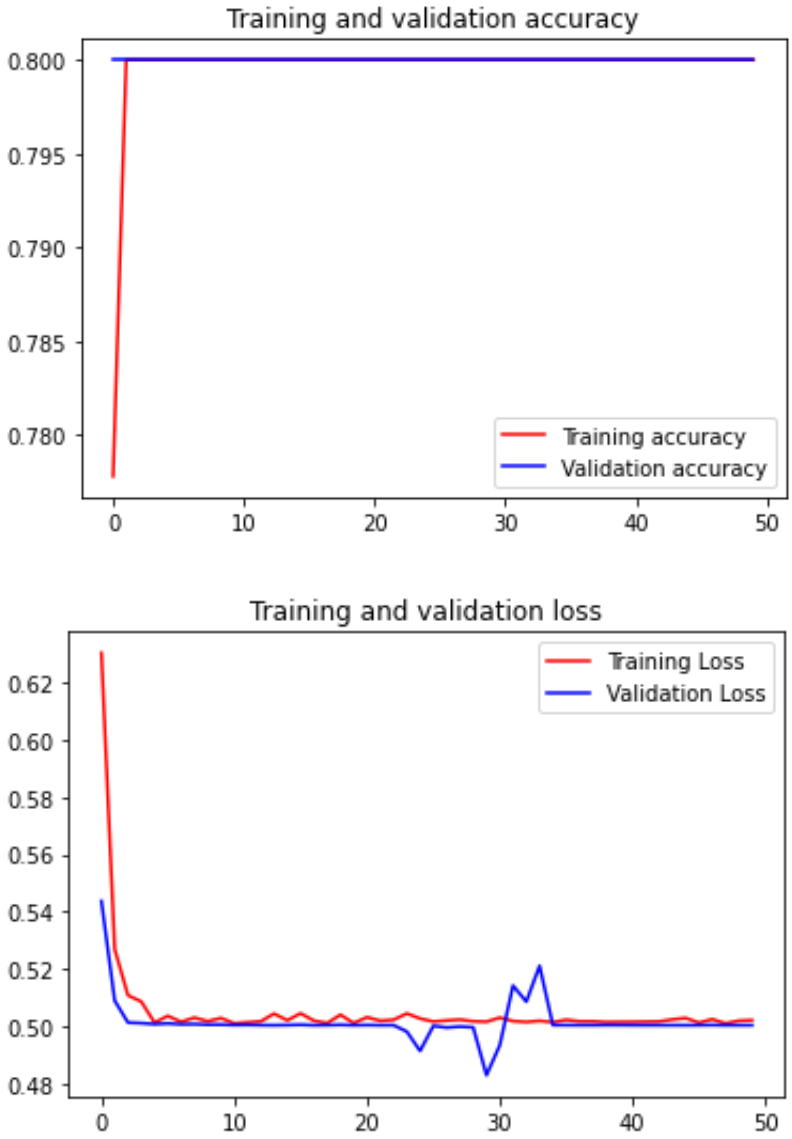



Figure 4.6 Training and Validation Performance of EfficientNet-B7 (Temperature Difference Approach with Data Augmentation)

Figure 4.6 illustrates the accuracy and loss model of EfficientNet-B7 on the IR augmented image dataset. It can be observed that the model achieved a validation accuracy of 80 performance. However, a significant amount of loss in training and validation dataset also persisted.

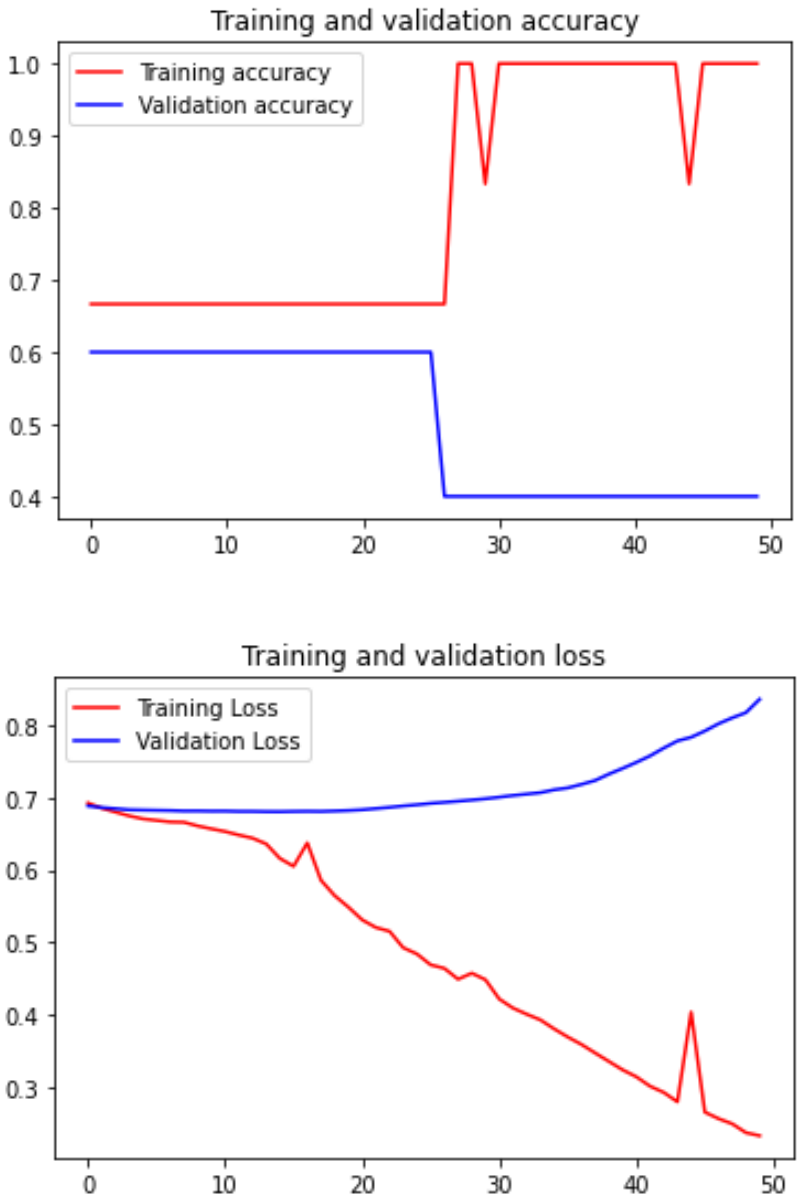



Figure 4.7 Training and Validation Performance of EfficientNet-B6 (Thresholding and Contouring Approach)

Figure 4.7 illustrates the accuracy and loss model of EfficientNet-B6 on the IR image dataset, labelled using Thresholding and Contouring. We can observe a significant decrease in training loss over an increase in number of epochs. However, on the validation dataset, the model performed poorly, with only a validation accuracy of 40 percent.

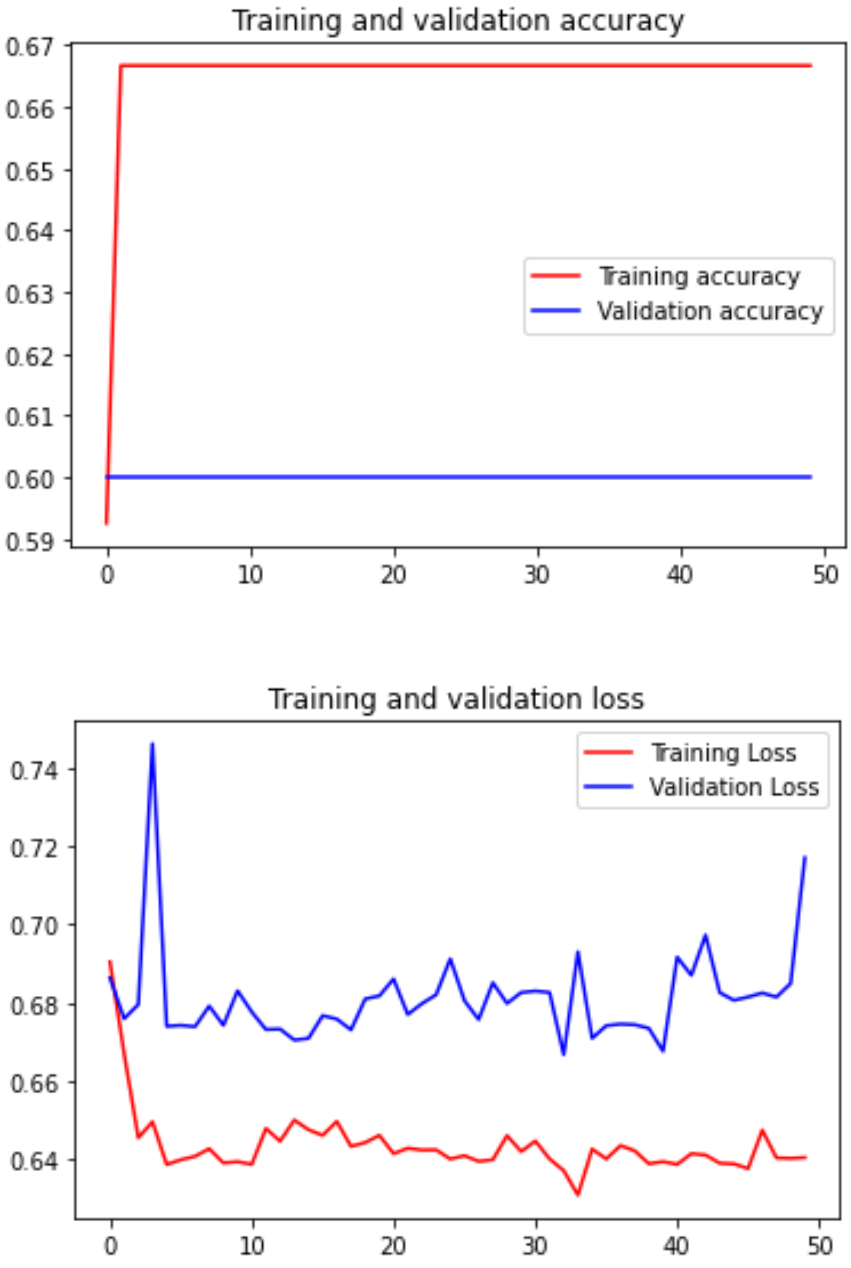



Figure 4.8 Training and Validation Performance of EfficientNet-B6 (Thresholding and Contouring Approach with Data Augmentation)

Figure 4.8 illustrates the accuracy and loss model of EfficientNet-B6 on the IR augmented image dataset, labelled using Thresholding and Contouring. It can be observed that the B6 model's validation has increased from 40 percent to 60 percent when the augmented dataset was used to train the model.

# Chapter 5

# Conclusion and Future Work

With increase in prominence of deep learning networks, more and more fields are incorporating deep learning frameworks in order to improve the efficiency of a particular task. As such, the need for autonomous monitoring and fault diagnostic techniques in electrical equipment and modules has become progressively essential over the years, particularly pressing more emphasis on image classification tasks. Early fault diagnosis of electrical modules can provision a reduction in manual labor, minimize cost and potential damage. In our study, we have put emphasis on solar cell modules whose heat emission can be analyzed and realized with the help of Infrared images. In addition, in order to develop an estimated difference between a potential defective and normal module, we have employed several different image processing techniques. These techniques were then used to label the images in order for the different CNN models to train on. Experimental results showed that for both normal and augmented datasets, the VGG-16 outperformed other models. Moreover, in order to improve the efficiency of the models, we have applied data augmentation, which has shown significant improvements in VGG-16 and VGG-19 models, while it has had little to no effect in EfficientNet models. One of the constraints that is imperative to mention is the outstanding lack of dataset- 15 images (due to Covid restrictions), which is an outstandingly small figure for a dataset that is to be used to train a CNN model. Due to the lack of dataset, the splitting of the dataset into training, validation and testing dataset was quite imbalanced as-well. As such, the results obtained were significant but not up to par. Our objective was to use the available dataset in order to assess the performances of the different CNN on the IR image dataset. All in all, we can conclude from our work, that among the 3 different CNN, VGG-16 has demonstrated the best performance in all datasets.

Fault diagnosis using image classifiers has witnessed progressive advancement over the years, especially with regards to deep learning frameworks. Our work can be further improved in the future by integrating a larger dataset that would help better assess the performances of the different models. Moreover, other state-of-the-art models, that depend on larger datasets, can also be employed in order to study their predictive capability on Infrared Images of solar cell modules. Last but not the least, other image processing techniques (such as Histogram-Profile Analysis or Line Profile Analysis) can also be used to potentially provide better labelling of the dataset. The ultimatum is to determine the most optimum CNN model that would have the best predictive capability in differentiating between IR images of defective and normal solar cell modules.

# References

[1] Ram, M.; Bogdanov, D.; Aghahosseini, A.; Gulagi, A., Oyewo, A., Child, M., Caldera, U., Sadovskaia, K., Farfan, J., Barbosa, L., 2019. Global energy system based on 100% renewable energy–power, heat, transport and desalination sectors. Study by Lappeenranta University of Technology and Energy Watch Group, Lappeenranta, Berlin.

[2] Kaizuka, I., J¨ager-Waldau, A., Donoso, J., 2020. Snapshot of Global PV Markets – 2020. In: Masson, G. (Ed.). International Energy Agency Photovoltaic Power System Programme.

[3] Islam, M.R., Rabbani, M.G., Hossain, M.D., Rahman, M.F., 2009. Effective Solar Photovoltaic Activities in Bangladesh. International Energy Journal 10 (2009) 29-36.

[4] Saleem, A., Rashid, F., Mehmood, K., 2016. The Efficiency of Solar PV System. Proceedings of 2nd International Multi-Disciplinary Conference 19-20 December 2016, Gujrat.

[5] AbdulMawjood, K., Refaat, S.S., Morsi, W.G., 2018. Detection and prediction of faults in photovoltaic arrays: A review. IEEE 12th International Conference on Compatibility, Power Electronics and Power Engineering (CPE-POWERENG).

[6] Z. Ye, Y. Ling, B. Lehman, J. F. de Palma, J. Mosesian, and R. Lyons, "Decision tree-based fault detection and classification in solar photovoltaic arrays," in The 27th Annual IEEE Applied Power Electronics Conference and Exposition (APEC), Orlando, Florida, USA, 2012, pp. 93-99.

[7] K. II-Song, "Fault detection algorithm of the photovoltaic system using wavelet transform," in The IEEE India International Conference on Power Electronics (HCPE), New Delhi, India, 2010, pp. 1-6.

[8] M. Tadj, K. Benmouiza, A. Cheknane, and S. Silvestre, "Improving the performance of PV systems by faults detection using GISTEL approach," Energy Conversion and Management, vol. 80, pp. 298-304, 2014.

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

[10] S. Syafaruddin, E. Karatepe, and T. Hiyama, "Controlling of artificial neural network for fault diagnosis of photovoltaic array," in The 16th International Conference on Intelligent System Application to Power Systems (ISAP), Hersonissos, Crete, 20 11, pp. 1-6.

[11] Deitsch, S., Christlein, V., Berger, S., Buerhop-Lutz, C., Maier, A., Gallwitz, F., Riess, C., 2019. Automatic classification of defective photovoltaic module cells in electroluminescence images. Sol. Energy 185, 455–4

[12] Dunderdale, C., Brettenny, W., Clohessy, C., Dyk, E.E., 2019. Photovoltaic defect classification through thermal infrared imaging using a machine learning approach. Prog. Photovolt. 28 (3), 177–188.

[13] S. R. Madeti and S.N. Singh, “A comprehensive study on different types of faults and detection techniques for solar photovoltaic system,” Solar Energy, vol. 158, Dec., 2017, pp. 161-185.

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

[15] J. Haney and A. Burstein, “PV System Operations and Maintenance Fundamentals,” Solar America Board for Codes and Standards, USA, DE-FC36-07GO17034, Aug., 2013. Accessed

on:Jan.26,2021.[Online].Available:http://www.solarabcs.org/about/publications/reports/operations- maintenance/pdfs/Solar%20ABCs-35-1-pager.pdf

[16] J. A. Silva, J. M. Serra, A. M. Vallêra and K. Lobato, “Luminescence in Photovoltaics,” in Fluorescence in Industry. Springer Series on Fluorescence (Methods and Applications), 1st ed., vol. 18. B. Pedras, Ed. Cham: Springer International Publishing, 2019, pp. 173-211. Accessed on: Feb. 2, 2021. [Online]. doi: 10.1007/4243_2018_7

[17] S. Spataru, P. Hacke, D. Sera, S. Glick, T. Kerekes and R. Teodorescu, "Quantifying Solar Cell Cracks in Photovoltaic Modules by Electroluminescence Imaging," in 2015 IEEE 42nd Photovoltaic Specialist Conf. (PVSC), New Orleans, LA, 2015, pp. 1-6. Accessed on: Feb. 2, 2021. [Online]. doi: 10.1109/PVSC.2015.7356124

[18] A. Bartler, L. Mauch, B. Yang, M. Reuter and L. Stoicescu, “Automated Detection of Solar Cell Defects with Deep Learning,” in 2018 26th European Signal Processing Conf. (EUSIPCO), Rome, Italy, 2018, pp. 2035-2039.

[19] J. A. Tsanakas, G. Vannier, A. Plissonnier, D. L. Ha and F. Barruel, “Fault Diagnosis and Classification of Large-Scale Photovoltaic Plants through Aerial Orthophoto Thermal Mapping,” in 31st European Photovoltaic Solar Energy Conf. and Exhibition (EUPVSEC), Hamburg, 2015, pp. 1783-1788.

[20] R. Moretón, E. Lorenzo, J. Leloux and J. M. Carrillo, “Dealing in practice with hot- spots,” in 29th European Photovoltaic Solar Energy Conf. and Exhibition (EUPVSEC), Amsterdam, 2014, pp. 2722 – 2727.

[21] C. Buerhop, D. Schlegel, M. Niess, C. Vodermayer, R. Weißmann and C. J. Brabec, “Reliability of IR-imaging of PV-plants under operating conditions,” Solar Energy Materials and Solar Cells, vol. 107, pp. 154-164, Dec., 2012.

[22] J. A. Tsanakas and P. N. Botsaris, “Infrared Thermography as an Estimator Technique of a Photovoltaic Module Performance via operating temperature measurements,” in 10th European Conference on NDT, Moscow, 2010.

[23] J. A. Tsanakas, D. Chrysostomou, P. N. Botsaris and A. Gasteratos, “Fault Diagnosis of Photovoltaic Modules through Image Processing and Canny Edge Detection on field thermographic measurements,” International Journal of Sustainable Energy, vol. 34, no. 6, pp. 351-372, July, 2015.

[24] M. S. Jadin and S. Taib, “Recent progress in diagnosing the reliability of electrical equipment by using infrared thermography,” Infrared Physics & Technology, vol. 55, no. 4, pp. 236-245, July, 2012.

[25] M. Aghaei, A. Gandelli, F. Grimaccia, S. Leva and R. E. Zich, “IR real-time Analyses for PV system monitoring by Digital Image Processing Techniques,” in 2015 Int. Conf. Event-based Control, Communication, and Signal Processing (EBCCSP), Krakow, 2015, pp. 1-6.

[26] C. Ibarra-Castanedo, D. Gonzalez, M. Klein, M. Pilla, S. Vallerand and X. Maldague, “Infrared Image Processing and Data Analysis,” Infrared Physics & Technology, vol. 46, no. 1, pp. 75-83, Dec., 2004.

[27] C. A. L. Almeida et al., “Intelligent Thermographic Diagnostic Applied to Surge Arresters: A New Approach,” IEEE Transactions on Power Delivery, vol. 24, no. 2, pp. 751-757, Apr., 2009.

[28] S. Rogotis, D. Ioannidis, A. Tsolakis, D. Tzovaras and S. Likothanassis, “Early defect diagnosis in installed PV modules exploiting spatio-temporal information from thermal images,” in 12th Int. Conf. Quantitative Infrared Thermography, Bordeaux, 2014. Accessed on: Feb. 13, 2021. [Online]. doi: 10.21611/qirt.2014.038

[29] V. S. B. Kurukuru, A. Haque, M. A. Khan and A. K. Tripathy, “Fault classification for Photovoltaic Modules Using Thermography and Machine Learning Techniques,” in Proc. 2019 Int. Conf. Computer and Information Sciences (ICCIS), Sakaka, Saudi Arabia, 2019, pp. 1-6.

[30] J. Antonanzas, N. Osorio, R. Escobar, R. Urraca, F. J. Martinez-de-Pison and F. Antonanzas-Torres, “Review of photovoltaic power forecasting,” Solar Energy, vol. 136, pp. 78-111, Oct., 2016.

[31] S. Rodrigues, H. G. Ramos and F. Morgado-Dias, “Machine Learning in PV Fault Detection, Diagnostics and Prognostics: A Review,” in 2017 IEEE PVSC-44, Washington, DC, 2017, pp. 3178-3183.

[32] L. L. Jiang and D. L. Maskell, “Automatic fault detection and diagnosis for photovoltaic systems using combined artificial neural network and analytical based methods,” in 2015 International Joint Conference on Neural Networks (IJCNN), Killarney, Ireland, 2015, pp. 1-8.

[33] Syafaruddin, E. Karatepe and T. Hiyama, “Controlling of artificial neural network for fault diagnosis of photovoltaic array,” in 2011 16th Int. Conf. Intelligent System Applications to Power Systems (ISAP), Hersonissos, Greece, 2011, pp. 1-6.

[34] C. Dunderdale, W. Brettenny, C. Clohessy and E. E. V. Dyk, “Photovoltaic defect classification through thermal infrared imaging using a machine learning approach,” Progress in Photovoltaics Research and Applications, vol. 28, no. 3, pp. 177-188, Dec., 2019.

[35] R. Pierdicca, E. S. Malinverni, F. Piccinini, M. Paolanti, A. Felicetti and P. Zingaretti, “DEEP CONVOLUTIONAL NEURAL NETWORK FOR AUTOMATIC DETECTION OF DAMAGED PHOTOVOLTAIC CELLS,” International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, vol. 422, pp. 893- 900, May, 2018.

[36] H. T. C. Pedro and C. F. M. Coimbra, “Assessment of forecasting techniques for solar power production with no exogenous inputs,” Solar Energy, vol. 86, no. 7, pp. 2017- 2028, July, 2012.

[37] Z. Chen, L. Wu, S. Cheng, P. Lin, Y. Wu and W. Lin, “Intelligent fault diagnosis of photovoltaic arrays based on optimized kernel extreme learning machine and I-V characteristics,” Applied Energy, vol. 204, pp. 912-931, Oct., 2017.

[38] A. Pai, CNN vs. RNN vs. ANN – Analyzing 3 Types of Neural Networks in Deep Learning, Analytics Vidhya, Feb. 17, 2020. Accessed on: Feb. 25, 2021. [Online]. Available: https://www.analyticsvidhya.com/blog/2020/02/cnn-vs-rnn-vs-mlp-analyzing- 3-types-of-neural-networks-in-deep-learning/

[39] I. Gogul and V. S. Kumar, “Flower species recognition system using convolution neural networks and transfer learning,” in 2017 4th Int. Conf. on Signal Processing, Communications and Networking (ICSCN), Chennai, India, 2017, pp. 1-6. Accessed on: Feb. 25, 2021. [Online]. doi: 10.1109/ICSCN.2017.8085675

[40] G. Huang, Q. Zhu and C. Siew, “Extreme learning machine: a new learning scheme of feedforward neural networks,” in 2004 IEEE Int. Joint Conf. on Neural Networks, Budapest, Hungary, vol. 2, 2004, pp. 985-990.

[41] G. Huang, GB. Huang, S. Song and K. You, “Trends in extreme learning machines: A review,” Neural Networks, vol. 61, pp. 32-48, Jan., 2015.

[42] Luo, S., Zhang, Q., Luo, F., Wang, Y., Chen, Z., 2004. An improved moment-preserving auto threshold image segmentation algorithm. International Conference on Information Acquisition, 2004.

[43] Amin, M., Hasan, G., Ahmed, I., Dewan, S., 2019. Infrared Thermography Based Performance Analysis of Photovoltaic Modules. Brac University. Retrieved from http://dspace.bracu.ac.bd/xmlui/handle/10361/12070.

[44] Simonyan, K., Zisserman, A., 2014. Very Deep Convolutional Networks for Large-Scale Image Recognition. Retrieved from https://arxiv.org/abs/1409.1556.

[45] Tan, M., Le, Q. V., 2019. EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. International Conference on Machine Learning, 2019.