SOILING TYPE CLASSIFICATION AND PREDICTION OF POWER LOSS OF A PV PANEL USING CNN

By
Alvi Ahmed
14321028
Tahseen Md. Nur
19321049
Omar Tanzim Zahid
16121068
Elma Minaz Ali
16121096

A thesis submitted to the Department of Electrical and Electronic Engineering in partial fulfillment of the requirements for the degree of
Bachelor of Science in Electrical and Electronic Engineering

Electrical and Electronic Engineering
BRAC UNIVERSITY
October 2020

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:

Alvi Ahmed 14321028	**Tahseen Md. Nur** 19321049
Omar Tanzim Zahid 16121068	**Elma Minaz Ali** 16121096

Approval

The thesis/project titled "Soiling Type Classification and Prediction of Power Loss of a PV Panel Using CNN" submitted by

1. Alvi Ahmed (14321028)
2. Tahseen Md. Nur (19321049)
3. Omar Tanzim Zahid (16121068)
4. Elma Minaz Ali (16121096)

of Summer, 2020 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of Bachelor of Science in Electrical and Electronic Engineering on 05.10.20

Examining Committee:

Supervisor:
(Member)

____
Dr. Mohammed Belal Hossain Bhuian
Associate Professor, EEE
BRAC UNIVERSITY

Program Coordinator:
(Member)

____
Abu S.M. Mohsin
Assistant Professor, EEE
BRAC UNIVERSITY

Departmental Head:
(Chair)

____
Dr. Shahidul Islam Khan
Chairperson & Professor, EEE
BRAC UNIVERSITY

Ethics Statement

We, hereby, declare that this thesis is based on results we have experimented. The contents, materials and datasets of work conducted by other researchers and analysts are provided in Reference. We confirm that this thesis report is submitted by the names listed on the book for the degree of Bachelor of Science in Electrical and Electronic Engineering of the Department of Electrical and Electronic Engineering under the School of Engineering and Computer Science, BRAC UNIVERSITY. We, hereby, declare that the materials and datasets used from other sources have been properly acknowledged and given credit. Finally, we declare and confirm that this thesis has never been previously submitted anywhere for research, publication or assessment purposes.

Acknowledgement

Firstly, we would like to express our gratitude to Almighty Allah who gave us the courage and resolve to give our all for this thesis and the strength to complete it.

We would want to dedicate our acknowledgement of gratitude and appreciation to our thesis supervisor, Mohammed Belal Hossain, PhD, Associate Professor, Department of Electrical and Electronic Engineering of BRAC UNIVERSITY for his support, guidance and feedbacks for the completion of our thesis.

We would also like to thank our co-supervisor, Abir Ahsan Akib, Lecturer, Department of Electrical and Electronic Engineering of BRAC UNIVERSITY, who helped us a lot in our thesis project. Last but not the least, we are very grateful to Md. Messal Monem Miah, Lecturer, Department of Electrical and Electronic Engineering of BRAC UNIVERSITY for providing us with his valuable advice and guidance without which we would not have achieved satisfactory result.

Abstract

With the rapid advancement of technology, greener and more efficient means for energy sources are always sought after. Harvesting solar energy is an effective way to generate electricity. Unfortunately, PV panel surface soiling is a major disruption in energy harvesting since it massively lowers the ability of the solar panel to be exposed to sunlight. Given how dire the air pollution situation is in Bangladesh, this is undoubtedly one of the major problems which have to be addressed when it comes to solar panels setup. When thousands of solar panels are setup in a remote location in which sunlight is abundantly available, the PV panel site has to be monitored to check if there are any issues, one of the issues being soiling. Manually checking thousands of PV panel images for soiling is laborious and time-intensive. We intend to automate that process using a lightweight deep learning model that can be incorporated into any system with fairly average computational power. More specifically, our deep learning model can determine if a particular PV panel is clean or soiled and classify the type of soiling. It can also make an approximate power loss prediction through image classification. This process will massively optimize the process of monitoring and negate the need for manually checking all the PV panels for soiling.

In this paper, we propose the aforementioned deep learning model and discuss in detail how it has been developed from scratch and how feasible it is.

.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

More than a quarter of the population of Bangladesh living in the countryside does not have access to electricity; but off-grid solar power also known as stand-alone power system is quickly changing this scenario. In fact, Bangladesh is associated with one of the world's largest domestic solar energy programs [1]. To generate electricity from solar power, we need solar panels which can be colloquially referred to as a photo-voltaic (PV) module. A PV module is an assembly of photo-voltaic cells, and these photo-voltaic cells use sunlight as their source of energy to generate electricity using the photovoltaic effect. PV panels are basically a collection of the PV modules. Bangladesh is a highly polluted country and its capital Dhaka has been ranked as the 21st most polluted city according to 2019 World Air Quality Report [2]. So, we can assume that a lot of dust and other impurities gather on these PV panels which in turn reduce the efficiency of these PV panels leading to power loss. In this paper, we shall mainly discuss how we can determine the soiling type and the percentage of the soiling which covers the PV panel surface so that an approximate power loss can be predicted using Convolutional Neural Networks (CNN) which is a part of the Deep Learning system.

1.1 Background and Motivation

At present, the use of PV panels to generate electricity is becoming extremely popular throughout the world. Most importantly, these panels are one of the key options in the future for being used as energy in our planet alongside wind power and water power. These panels are important because they are used to generate a clean source of energy as it is renewable in nature and causes less harm to our environment. Renewable resources are basically those resources which once used can be replaced so it is readily available for our future generations. In spite of these panels having so many advantages, sometimes it fails to execute its outcome properly due to the degradation in performance of the PV cell over time. One of the reasons of this failure might be due to the collection of dust, sand and other impurities on the panel. Other parameters like weather, humidity and wind speed can also influence the performance of the PV panel which can result in the performance degradation of the panel and this performance degradation can vary from one panel to another depending on its shelf-life [3]. Bangladesh, being a highly polluted country, faces one of the challenges which is the reduction in the efficiency of these panels due to excessive collection of dust. The main purpose of this thesis is to be able to determine not only the power loss but also the soiling type which reduces the efficiency of the PV panels.

Since Bangladesh faces crisis in generating sufficient electricity for its people, it has bright prospects in terms of generating electricity from solar power. According to a report in UN Chronicle, the vision of the government of Bangladesh is to make electricity available to all by 2021 [4]. According to that report, to reduce the crisis of electricity renewable energy will play a major role especially in the off-grid areas of the country. The target of the government is to generate 10 per cent of the of the total electricity supply from the total renewable energy sources by the end of 2020 [4]. Using renewable energy resources to harness electricity can definitely assist in the Government’s declared vision of “Electricity for all by 2021” [4].

1.2 Method

The main method which we will be using in this thesis is Convolutional Neural Networks (CNN) using Tensor Flow to predict the soiling type and power loss. CNN is related to the Deep Learning segment of the machine learning. We have created a set of data which includes 5 types of soiling which are: sand, grey sand, dust, droppings and clean (no soiling). Moreover, we have created another dataset as well which shows the corresponding percentage of the soiling type covering the PV panel: 0%, 10%, 20%, 30%, 40%, 50%, 60%, 70%, 80%, 90% and 100%, where 0% indicates that the PV panel is clean and free of any impurities. The difference in percentages of the PV panel covered in soiling basically corresponds to the power loss.

1.3 Objective

The main objective of this thesis is to be able to predict the soiling type and power loss of the PV panel due to the accumulation of different types of dirt on it. As Bangladesh is polluted, so a lot of dust accumulates on the PV panel leading in the reduction of the efficiency of the PV panels. Considering the dust covered PV panels available in Bangladesh, it is quite suitable for us to be able to conduct our thesis and predict the power loss. As machine learning is becoming increasingly popular and will soon become an inseparable part of all technological advancements in the near future, we have decided to incorporate a segment of it in our thesis. We really look forward to see the success of this project in the machine learning field.

1.4 Overview of the Contents

This thesis paper is divided into six sections to give a proper overview of our work. Chapter 1 includes the introduction to our thesis and then goes on to discuss about the background and motivation, methods, objective of our thesis paper. Chapter 2 includes the introduction to Machine Learning and then goes on to discuss about the more important topics which include: Deep Learning, CNN with Tensor Flow and Image Classification using Keras in Python. Chapter 3 mainly focuses on the data collection process while Chapter 4 mainly discusses the algorithms used in this to build the CNN model. Chapter 5 discusses the results and analysis of the algorithm and conclusion is discussed in Chapter 6.

1.5 Literature Review

The influence of environmental dust particles on the PV module has been assessed in the present-day study by measuring the electrical efficiency index, such as voltage, current and power. Some related works were done earlier with PV panels for determining the power loss and soiling type and how soiling can affect the power loss of the in solar photovoltaic module. According to this paper [13], the authors concluded that soiling like dust can significantly reduce the efficiency of the solar panel. Their study was carried out using different values of voltage and current with different dust samples with different mass and sizes. Then using those values of current and voltage the corresponding power loss (P =I*V) was calculated for their different sets of data. In our experiment, we also followed the same methodology as theirs but in addition we also took clean PV panels into consideration to be able to compare the efficiency with the dirty ones covered in different types of soiling. Another paper [14] "Deep Solar Eye" used CNN based approach for solar panel soiling and defect analysis. The inputs of their system were RGB images and outputs were soiling impact, soiling localization and soiling category. Their model was quite complex compared to ours which is quite simple and can be handled by any data set. On the other hand, just to run their deep solar eye model, it would require high computational power making the system quite inconvenient. Unlike them, we have generated our own dataset of solar panel images to compute the power loss.

Chapter 2

Introduction to Machine Learning

The method of analyzing data that automates an analytical model building is known as machine learning. Machine Learning also known as ML is the study of computer algorithms that can automatically enhance their performance through experience [5]. Artificial Intelligence (AI) is seen as a subdivision of machine learning system. A mathematical model is built based on the sample data also known as the training data by the ML algorithms to make predictions or decisions without the need of being explicitly programmed. Today, Machine Learning is not like the machine learning of the past it has evolved a lot throughout the years because of the advancements in the new technologies. Although many ML algorithms have been used around for a very long time but its ability of applying complex mathematical calculations automatically to big data quite faster is recent addition in terms of development. Machine learning works by exploring different types of data and by identifying different complex pattern based of their past interactions with minimal human intervention. The two main techniques used by machine learning are: supervised learning and unsupervised learning. Supervised learning means that the model of machine learning is getting trained on a labeled dataset which means the dataset will have both input and output parameters. Unlike unsupervised learning where there will be unlabeled datasets and only the input data will be given. ML is very important because of its ability of detecting and identifying complex patterns which would have been ignored during the analysis done by the humans. Due to machine learning, deep learning, machine vision and other natural language processing, the human workers are able to focus more on tasks like product innovation and on improving quality of service as well as efficiency. Machine learning uses programming languages like python, R and plenty of other languages. Out of all the programming languages, Python is considered to be the most popular programming language for ML.

2.1 Programming Language and Environment

The programming language which we chose to implement our proposed model for our thesis is Python. Python is a highly acclaimed high-level programming language used for general purpose programming all over the world. Python's popularity is such that it is almost installed in almost all of the desktops and it is flexible to use it for projects from home or from cloud server like Google Colab. The main reason of choosing Python as programming language is due to its advancing features which enables it to support multiple programming paradigms which include being object-oriented, imperative functional and procedural which can assist us in accomplishing our tasks.

2.2 Deep Learning

Deep learning also known as the sub-field of Machine learning is concerned with algorithms and Artificial Neural Networks (ANN) which took inspiration from the biological functions and structures of the brain to implement its function. Deep learning is important because it allows the machine to any solve complex problems even when the data set with which they are dealing is quite diverse, unstructured or inter-connected. Deep learning is becoming more and more important these days due to its ability of unraveling massive amount of unstructured or unlabeled data which could take nearly decades for the human beings to understand and process this huge amount of information. Deep learning is widely used in almost everything starting from self-driving cars, fraud detection to commercial apps that use image recognition, open-source platforms with consumer recommendation app and even in the development of medical research tools. Day by day deep learning is becoming a very important and inseparable aspect of our day to day life. In deep learning, the word "deep" implies the number of layers through which the data is transformed. Deep learning is strongly related to both Machine learning (ML) and Artificial Intelligence (AI) and their relationship is explained in the figure below. It can be seen from the figure that deep learning is a subset of machine learning.

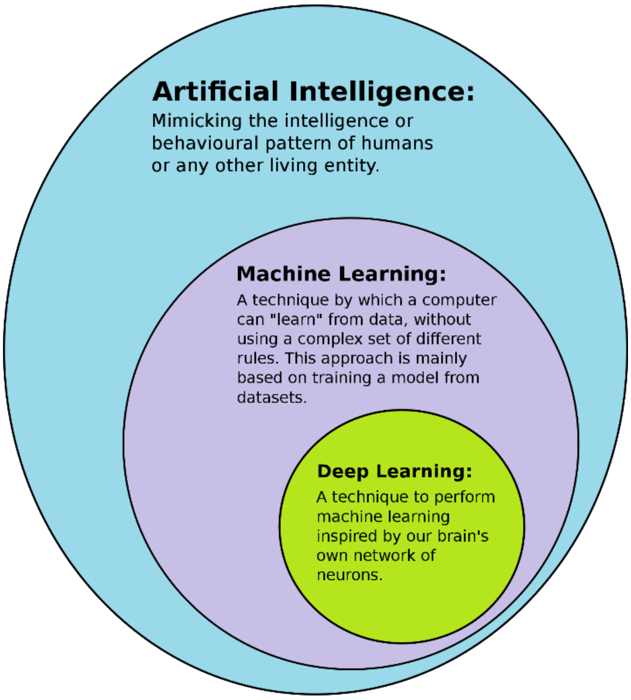


Figure 1: Relationship of Deep learning with Machine learning and Artificial Intelligence [10]

2.3 CNN with TensorFlow

Convolutional Neural Network (CNN) also known as shift invariant or space invariant artificial neural networks is a very powerful neural network that can extract features from images using filters. The CNN does its work in such a way that the information about the position of the pixels is retained. The word convolution in CNN actually refers to a mathematical operation which is applied on a matrix. The image represented in the form of pixels/numbers is the matrix on which the convolutional operation is applied. From the image the convolution operation extracts the features required. Neural networks have contributed to tremendous breakthroughs in machine learning and they are the underlying factor behind the huge success of deep learning [8]. Using CNN for image classification is better than using Artificial Neural Network (ANN) because while using ANN, the numbers of trainable parameters increases which in turn leads to the increase in size of the image. Unlike ANN, the CNN can easily capture the spatial features from an image [8]. Convolutional Neural Networks have become the best way to overcome any obstacle regarding image data. Popular frameworks like TensorFlow-Keras and PyTorch support CNN. Tensor flow is basically an open source library used in machine learning and in numerical computation and it was developed by the Google Brain team. The three basic components needed to define the convolution network are: convolutional layer, pooling layer and the output layer. The function of the convolution layer involves combining the outputs of the local convolutions and reducing the spatial dimensions of the data [9]. The function of the pooling layer is mainly to reduce the spatial size of the image and max pooling is one of the common forms of pooling layer. The output layer has a loss function, such as categorical cross-entropy, to measure the error in the prediction. Despite of the CNN being powerful models one could still face challenges while executing it and some these include: the need for large amount of data and taking up a lot power for processing. Convolution in mathematical representation is often denoted by an asterisk (*) sign. For example, if we have an input image represented as Y and a filter denoted by f, then the expression should be:

$$Z = Y*f \quad (2.1\ 1)$$

In the training of any neural networks two processes are involved and they are: forward propagation and backward propagation [12]. In forward propagation, the images are fed into the input layer in the form of numbers and the intensity of the pixels in the image is denoted by these numbers. In backward propagation, after the output is generated in the next stage the output is compared with the actual value in order to calculate the error and it also updates the parameters of the network. The steps of executing forward propagation are summarized below:

1. The first step is to load the input images in a variable (for e.g. Y)
2. A filter matrix needs to be initialized and the input images are convolved using the filter.

   $Z_{1} = Y * f$ (2.1.2)

3. The sigmoid activation function is applied on the output.

   $$B = \text{Sigmoid}(Z_1) \qquad (2.1.3)$$

4. Weight and bias matrix are randomly initialized and then linear transformation is applied on the values.

   $$Z_2 = W^T . B + b \qquad (2.1.4)$$

5. In the last step the sigmoid function is applied on the data and the final output becomes:

   $0 = \text{Sigmoid}(Z_2)$ (2.1.5)

For backward propagation, the same forward propagation process is repeated but using the updated parameters and then the new outputs are generated.

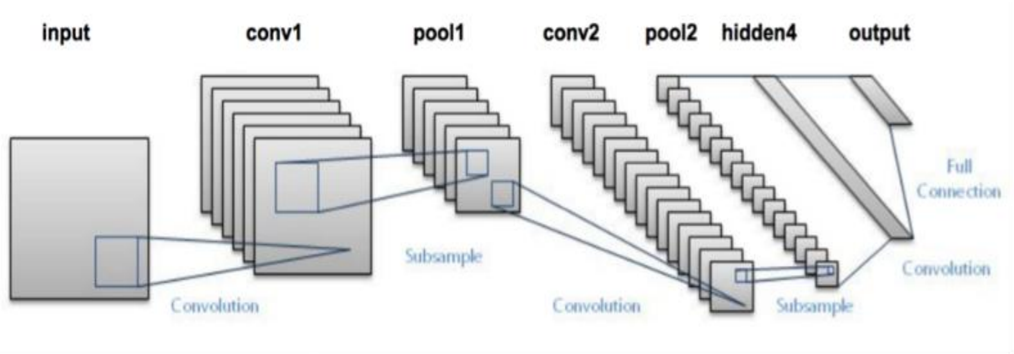


Figure 2: Showing different types of layers in a CNN model [8].

2.4 Using CNN to Classify Images in Keras

Image classification is a technique by which images can be classified into their respective category classes using methods which includes training a small network from scratch and fine tuning of the model's top layers. To train a model from scratch, two types of data are needed: one is the test data and the other one is the train data. A set of observations used to assess the model's performance using certain performance metrics is known as the test data and the set of data to create the model is known as the train data. In Python, Keras is a minimalist library used for deep learning that can run on Theano or TensorFlow. The process for classifying images involves 7 steps which are given below:

1. Collecting the dataset - To train our machine large amount of data is needed which includes the test data and the train data.
2. Importing libraries and splitting the dataset - The libraries need to be imported and then the data set needs to be split into training set data and the testing set data.
3. Building the CNN model.
4. Full connection - It means it connects the CNN to a neural network and compiles the network.

5. Data Augmentation - It is the method using which the amount of training data can be increased using only the information in the training set. In this way, overfitting of the models can be minimized.
6. Training the network.
7. Testing - Done using the data from the testing set.

Chapter 3

Data Collection

We have used two different datasets for our thesis project. For determining whether the PV panel is clean or dirty and classifying the type of impurity on the panel, we have used a dataset comprising of images which we have collected from the internet and the link is given here: https://deep-solar-eye.github.io/. This dataset contains precisely 45,754 images of PV panels including clean ones as well as the different types of impurities with different levels of soiling [11]. There are many different kinds of impurities included in the dataset but for our convenience to implement in the model, we have reduced the type of soiling into five categories: Sand, Grey Sand, Dust, Droppings and Clean (no soiling on it). The images of the same PV panel but with different types of soiling are included below:

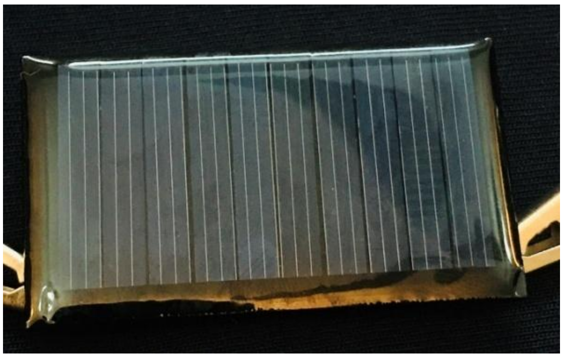



Figure 3: PV panel covered in sand   Figure 4: PV panel covered in grey sand

Figure 5: PV panel covered in dust   Figure 6: PV panel covered in droppings

Figure 7: A clean PV panel

To calculate power loss by determining how much the area of the PV panels are covered with impurities or if there are any impurities present at all, we have created our own dataset. We got ourselves a mini PV panel and used it to experiment and collect data. Just like for soiling classification in which we have used 5 classes, we have also made 11 classes for power loss prediction. The classes are: 0%, 10%, 20%, 30%, 40%, 50%, 60%, 70%, 80%, 90% and 100%. We have collected a total of 1100 data with 100 data for each category. We split the dataset into 3 parts: training, validation and testing. The PV panels have been covered with sand and the images have been captured from different angles. The images for different classes are shown below:



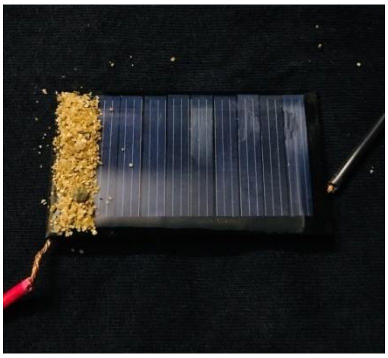


Figure 8: 0% coverage PV panel Figure 9: 10% coverage PV panel

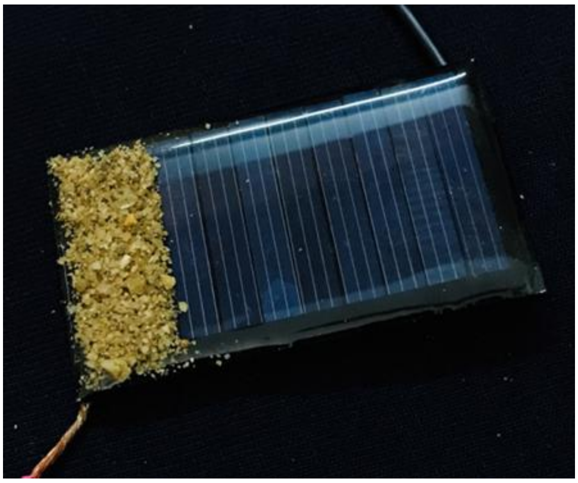


Figure 10: 20% coverage PV panel

Figure 11: 30% coverage PV panel

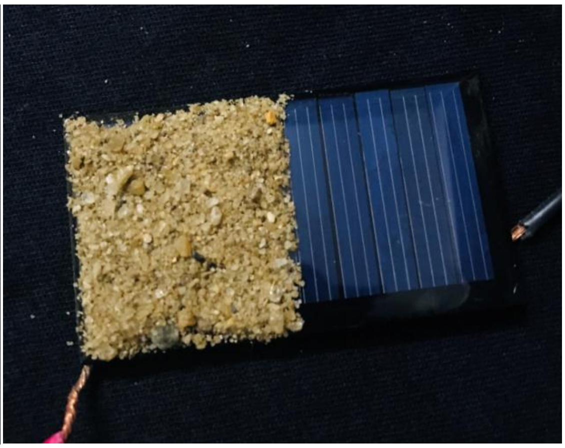
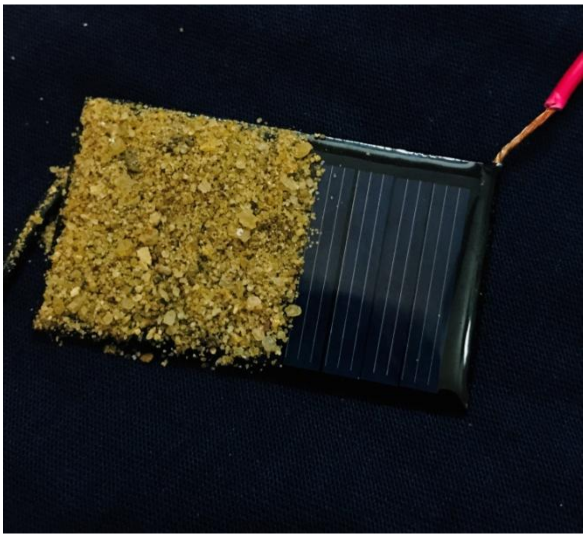
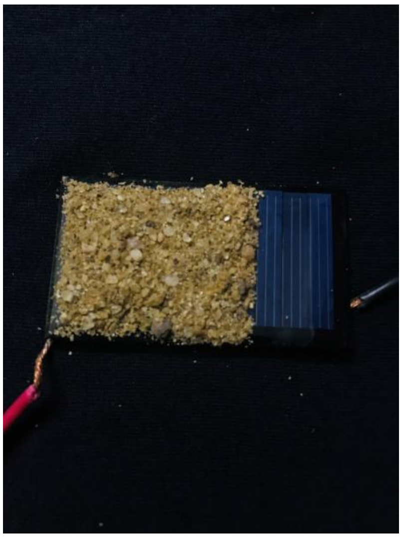


Figure 12: 40% coverage PV panel Figure 13: 50% coverage PV panel





. Figure 14: 60% coverage PV panel Figure 15: 70% coverage PV panel

. Figure 16: 80% coverage PV panel Figure 17: 90% coverage PV panel

Figure 18: 100% coverage PV panel

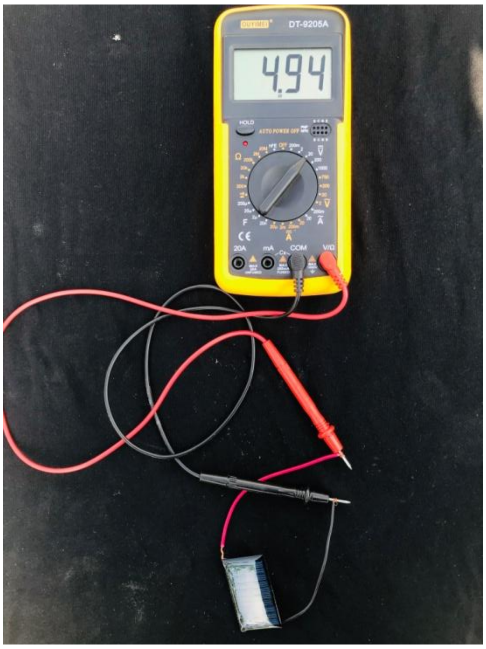


Figure 19 and 20: Demonstration of the experiment

Our mini solar panel had an area of 16 cm^2. For 10% coverage, we calculated the area of the 10% which was covered, which is 1.6 cm2. We followed the same approach for the other different classes as well. By repeating it, we took 100 photos from different angles for each class.

Chapter 4

Algorithms

4.1 Algorithms for Soiling Type Classification

Our algorithm has been developed from scratch after a lot of trial and errors. 4100 total images are allocated for this algorithm. 3100 images are taken for training, 500 for validation and 490 for testing.

i) Image Load and Preprocessing

Training and validation image datasets are initially preprocessed and loaded using the ImageDataGenerator [18] function. The dataset images are further augmented using the same function, whereby the total number of training and validation images increases.

ii) Model Architecture

A custom sequential model architecture is designed, which follows the CNN method. In total, there are 7 layers, 4 of which are convolutional layers and 3 are fully connected layers. Initially, a convolutional layer of 16 filters and a kernel size of 3 by 3 is employed for image feature extraction. This process is repeated in another layer in which 32 image features are extracted. It is repeated two more times, and the final layer contains 128 filters. All convolutional layers have "relu" as the activation signal. Maxpool feature is set to 2 by 2 which follows every convolutional layer. The image features matrices that have been extracted so far are flattened. There is a total of 3 dense layers. First dense layer consists of 128 neurons and uses “relu” as the activation signal. The next dense layer consists of 64 neurons, and it also uses “relu” as the activation signal. The final dense layer, which is basically the output layer, consists of a number which is equal to the number of image classifications which has to be made. In our case, we have used 5 for soiling type classification and 11 for power loss prediction. Softmax [14] is used as the activation signal so that the probabilities of all the different classes can be summed up to 1.

iii) Training the Dataset:

The model architecture is compiled using the "adam" optimizer, and the metrics is set to "accuracy". Categorical crossentropy [15] is used as the cost function since there are 5 different classes of classification. After that, the model is trained and validated upon the training and validation datasets respectively. The validation accuracy and loss results as well the test and prediction results will be discussed in Chapter 5.

4.2 Architecture for Soiling Classification

Here is a tabulated summary of our model architecture for soiling type classification:

Layer Type/ Number	Kernel Size	Activation	Filter Size	Output Size	Parameters
Convolution 1	3x3	Relu	16	198x198	448
Max Pool 1	2x2			99x99	0
Convolution 2	3x3	Relu	32	97x97	4640
Max Pool 2	2x2			48x48	0
Convolution 3	3x3	Relu	64	46x46	18496
Max Pool 3	2x2			23x23	0
Convolution 4	3x3	Relu	128	21x21	73856
Flatten	12,800				0
Dense Layer 1	128	Relu			1638528
Dense Layer 2	64	Relu			8256
Output Layer	5	Softmax			325

Figure 21: Architecture model for soiling classification

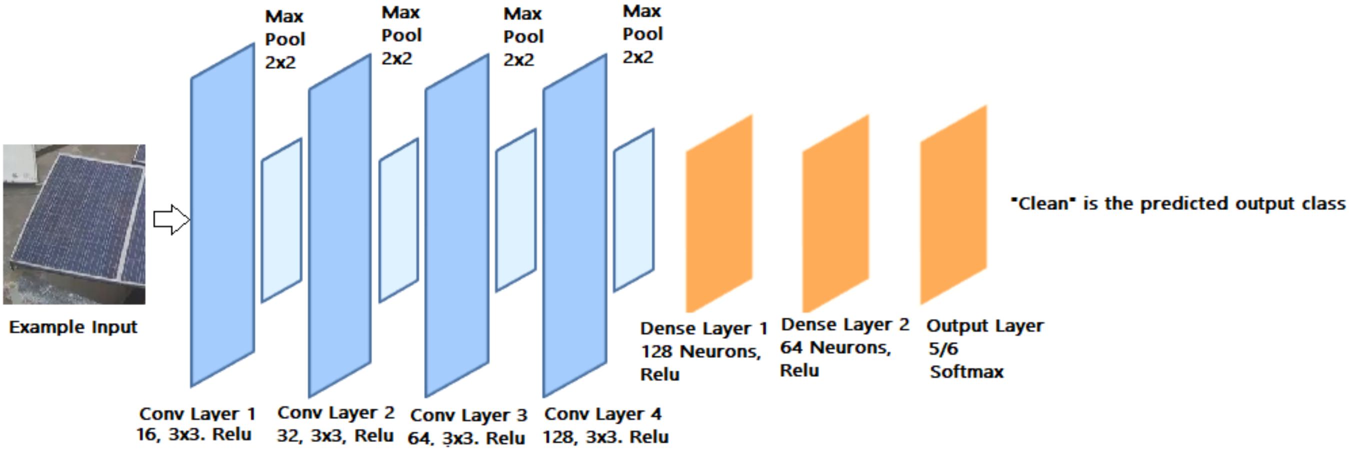


Figure 22: Overview of soiling classification

4.3 Dataset for Soiling Type Classification:

As discussed in Chapter 3, the image dataset we collected has a solar panel image dataset containing almost 46,000 images. We utilized only 4,000 of those for the sake of simplicity since our ultimate goal is to determine how effective and accurate our algorithm is even when lower number of images is used for training. 3,000 images were set for training. 500 were set for validation, and 500 more were set for testing purpose.

In our first classification model, our model classifies the type of soiling in a solar panel. We used 5 classes in our model, namely: clean, sand, grey sand, bird droppings and dust.

4.4 Code for Soiling Type Classification:

Here is a snippet of our model architecture:

```
model = Sequential()
model.add(Conv2D(filters=16, kernel_size=(3, 3), activation="relu", input_shape=(200,200,3)))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Conv2D(filters=32, kernel_size=(3, 3), activation='relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Conv2D(filters=64, kernel_size=(3, 3), activation="relu"))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Conv2D(filters=128, kernel_size=(3, 3), activation='relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Flatten())
model.add(Dense(128, activation='relu'))
model.add(Dense(64, activation='relu'))
model.add(Dense(5, activation='softmax'))
```

Figure 23: Snippet of model architecture for soiling classification

4.5 Flowchart

Here is the flowchart:

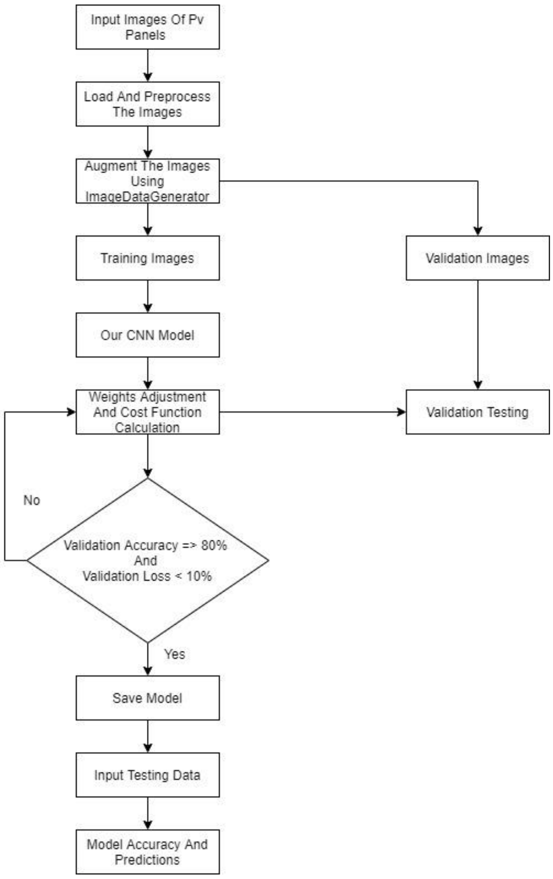


Figure 24: Flowchart for soiling type classification

4.6 Algorithm for Area Coverage Classification

We used the same algorithm as the soiling classification but fine-tuned it even more to get more accurate results for area coverage classification. A total of 1100 total images are allocated for this algorithm with 100 images for each category.

Just like the previous part, ImageDataGenerator [18] function is used. This allows the dataset images to get further augmented using the same function, whereby the total number of training and validation images increases.

For determining the power loss by finding out the impurity coverage on the PV panel, we have bought a mini PV panel with a specification of 5V and 30mA whose maximum power output is 0.15W and has a total area of 16cm^2. With this PV panel and by using a multimeter, we have recorded data for 11 classes from 0% to 100% with an increment of 10%. These classes represent the amount of impurities covering the PV panel. By recording the voltage and the current for each reading, we have calculated the output power. For the 11 classes (0%, 10%, 20%, 30%, 40%, 50%, 60%, 70%, 80%, 90% and 100%), we tried to keep the same number of data in each category so that the dataset does not get skewed. We followed the same architecture model for predicting the area covered by a PV panel but we fine-tuned it to get more accurate results for area coverage prediction. Just like the previous model, we used a total of 4 convolutional layers and 3 fully connected layers. We used the same number of filters which is 16 for the first layer but we used a kernel size of 5 by 5 for extracting the image characteristics. In the second layer, we used a kernel size of 3 by 3 and a filter of 32 and this repeated two more times. For the two times, the kernel size was the same but we increased the filter size from 32 to 64 and finally to 128. The MaxPool feature was set to 2 by 2 for the whole process and "relu" activation was used in every layer except for the last one. Before proceeding to the dense layer, the matrices which were created in the previous layers are flattened as the machine can only read data in 1D. We set the properties of the dense layers as the same as the one used in the soiling classification algorithm. The only difference we made here is that we set the last dense layer to 11 as we had 11 classes for area coverage prediction. Softmax was used for the activation layer and "adam" optimizer was used with categorical crossentropy set as the cost function and the metrics being set to "accuracy".

4.7 Architecture for Area Coverage Classification

And here is the tabulated summary for area coverage prediction:

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_4 (Conv2D)            (None, 196, 196, 16)      1216
_________________________________________________________________
max_pooling2d_4 (MaxPooling2 (None, 98, 98, 16)        0
_________________________________________________________________
conv2d_5 (Conv2D)            (None, 96, 96, 32)        4640
_________________________________________________________________
max_pooling2d_5 (MaxPooling2 (None, 48, 48, 32)        0
_________________________________________________________________
conv2d_6 (Conv2D)            (None, 46, 46, 64)        18496
_________________________________________________________________
max_pooling2d_6 (MaxPooling2 (None, 23, 23, 64)        0
_________________________________________________________________
conv2d_7 (Conv2D)            (None, 21, 21, 128)       73856
_________________________________________________________________
max_pooling2d_7 (MaxPooling2 (None, 10, 10, 128)       0
_________________________________________________________________
flatten_1 (Flatten)          (None, 12800)             0
_________________________________________________________________
dense_3 (Dense)              (None, 128)               1638528
_________________________________________________________________
dense_4 (Dense)              (None, 64)                8256
_________________________________________________________________
dense_5 (Dense)              (None, 11)                715
=================================================================
Total params: 1,745,707
Trainable params: 1,745,707
Non-trainable params: 0
```

Figure 25: Architecture model for area coverage classification

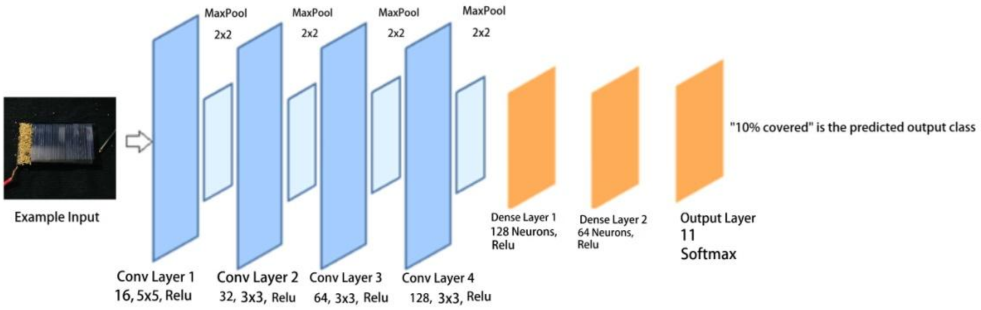


Figure 26: Overview of area coverage classification

4.8 Dataset for Area Coverage Classification:

As previously mentioned, the image dataset we collected has an image dataset of 1100 images. For our area coverage classification model, our model classifies the type of area coverage due to soiling in a solar panel. We used 11 classes in our model: 0%, 10%, 20%, 30%, 40%, 50%, 60%, 70%, 80%, 90% and 100%.

For power loss prediction, our model classifies what percentage of the solar panel is covered in soiling and makes an approximate power loss estimation from it by subtracting the output power for given coverage amount from the power found from a 0% impurity covered PV panel.

4.9 Code for Area Coverage Classification

Here is a snippet of our model architecture for area coverage classification:

```
[27] model = Sequential()
     model.add(Conv2D(filters=16, kernel_size=(5, 5), activation="relu", input_shape=(200,200,3)))
     model.add(MaxPooling2D(pool_size=(2, 2)))
     model.add(Conv2D(filters=32, kernel_size=(3, 3), activation='relu'))
     model.add(MaxPooling2D(pool_size=(2, 2)))
     model.add(Conv2D(filters=64, kernel_size=(3, 3), activation="relu"))
     model.add(MaxPooling2D(pool_size=(2, 2)))
     model.add(Conv2D(filters=128, kernel_size=(3, 3), activation='relu'))
     model.add(MaxPooling2D(pool_size=(2, 2)))
     model.add(Flatten())
     model.add(Dense(128, activation='relu'))
     model.add(Dense(64, activation='relu'))
     model.add(Dense(11, activation='softmax'))
```

Figure 27: Snippet of model architecture for area coverage classification

4.10 Flowchart for Area Coverage Classification

Figure 28: Flowchart for area coverage and power loss prediction

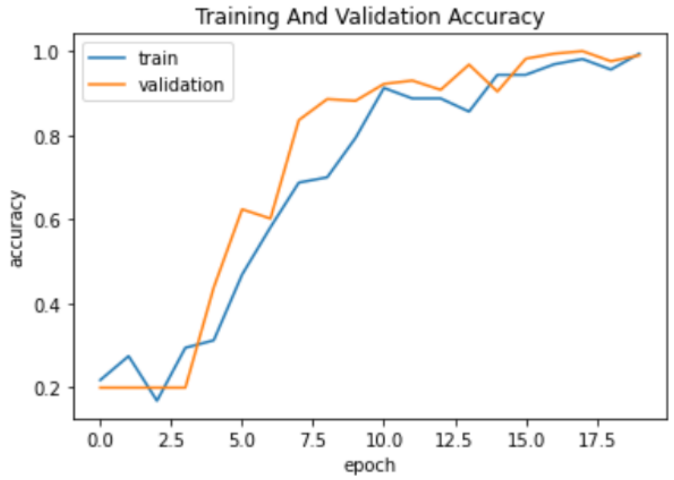
Chapter 5

Results and Analysis

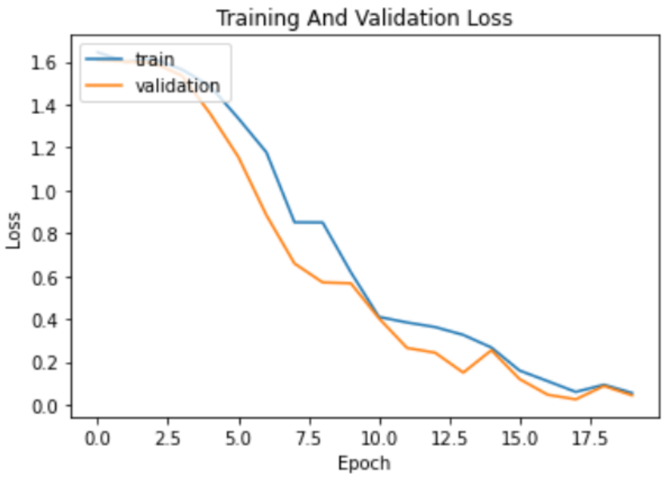
5.1 Soiling Type Classification Validation Accuracy and Loss

The results of the algorithm in Chapter 4.1 will be discussed in details here. For soiling classification, after 20 epochs, the validation accuracy and loss of our model architecture stands at 97% and 9.7% respectively, which means our model is extremely reliable for image classification.

Here is the accuracy graph:



Here is the loss graph:



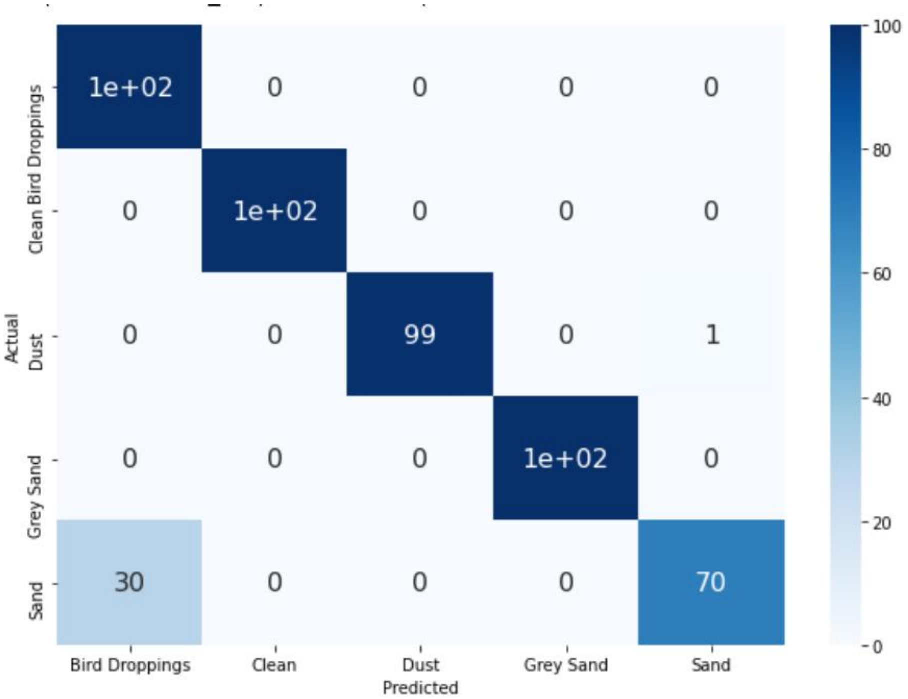
5.2 Test Accuracy for Soiling Type Classification

Our testing dataset accuracy, as of writing this, stands at 99.4%, and the loss stands at 5.42% using model predict function of Keras.

```
Evaluate on test data
16/16 [==============================] - 1s 72ms/step - loss: 0.0542 - categorical_accuracy: 0.9939
Test loss, Test accuracy: [0.05415598303079605, 0.9938775300979614]
```

5.3 Confusion Matrix Plot and Prediction Results for Soiling Type Classification

Here is our Confusion Matrix [18] chart:

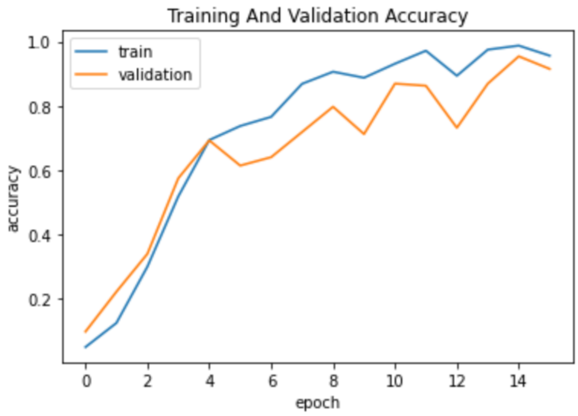


The model is finally put into practice and a set of 10 unseen PV panel image data is fed into the system for the purpose of prediction. 9 of the 10 images are corrected predicted by our model.

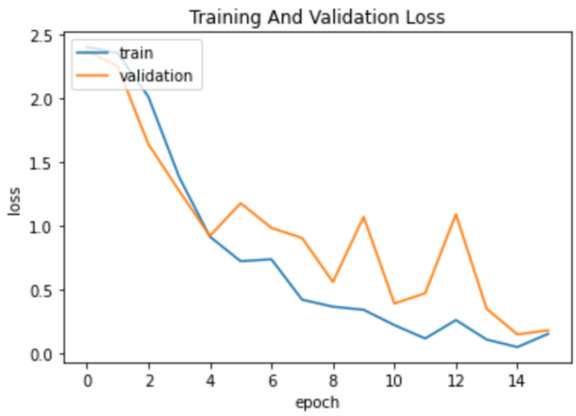
5.4 Area Coverage Classification Validation Accuracy and Loss

For area coverage classification, after 16 epochs, the validation accuracy and the validation loss stand at 91.50% and 17.66% respectively. This means that our model is able to detect how much area is covered on a PV panel by soiling to an impressive accuracy.

Here is the accuracy graph:



Here is the loss graph:



Also, with the output power calculated by multiplying V_{oc} and I_{sc} and determining the area covered for the 11 classes, we plotted an output power vs. area covered by soiling graph. A straight downward slope was observed which is correct for this graph. To calculate power loss for a particular class, we found out the output power for that particular area coverage on the PV panel and subtracted it from the output power for 0% coverage. The difference in the values gives us the approximate power loss.

Here is the tabulated data of the output power and area covered by soiling:

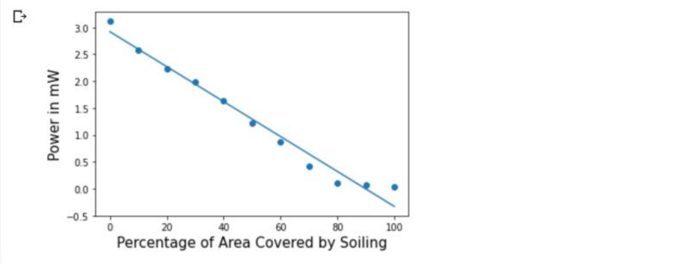
Percentage of Area Covered by Soiling in cm^2	Power in mW
0	3.112
10	2.578
20	2.233
30	1.1988
40	1.632
50	1.211
60	0.864
70	0.421
80	0.112
90	0.062
100	0.031

Here is the graph of Power in mW vs. Percentage of Area Covered by Soiling:

```
import numpy
import matplotlib.pyplot as plt

x = [0, 10, 20, 30, 40, 50, 60, 70, 80, 90, 100]
y = [3.112, 2.578, 2.233, 1.988, 1.632, 1.211, 0.864, 0.421, 0.112, 0.062, 0.031]

mymodel = numpy.poly1d(numpy.polyfit(x, y, 1))
myline = numpy.linspace(0, 100, 100)
plt.scatter(x, y)
figure = plt.plot(myline, mymodel(myline))
plt.xlabel('Percentage of Area Covered by Soiling',fontsize = 15)
plt.ylabel('Power in mW', fontsize = 15)
plt.show()
```

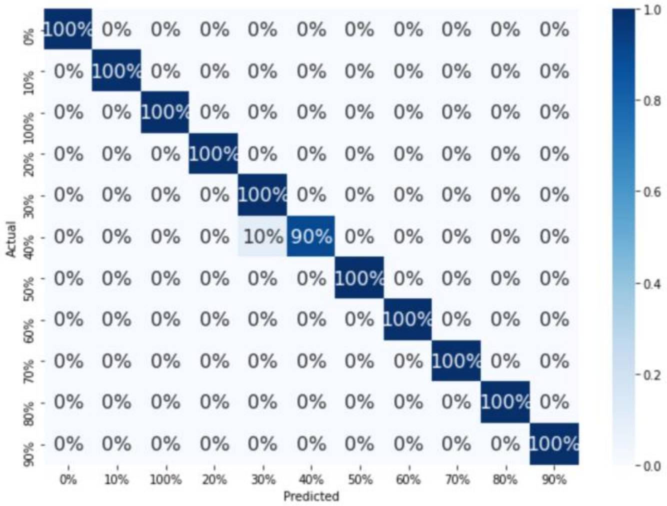


5.5 Test Accuracy for Area Coverage Classification

```
Evaluate on test data
4/4 [==============================] - 11s 3s/step - loss: 0.0415 - accuracy: 0.9909
test loss, test acc: [0.04148133844137192, 0.9909090995788574]
```

The test accuracy stands at 99.09%, and the loss stands at 4.15% using model evaluate function of Keras.

5.6 Confusion Matrix Plot and Prediction Results for Area Coverage Classification



The model is put into practice and a set of 10 unseen and different amount of surface covered PV panel image data is fed into the system for the purpose of prediction. 9 of the 10 images are corrected predicted by our model.

Chapter 6

Conclusion and Future Perspective

Our ultimate goal was to develop an automated model for soiling type classification and power loss prediction that is computationally efficient and fairly accurate. However, this particular model is limited to image classification model only. It can be further improved by using Image Segmentation [17] method which can both detect and localize the soiling area upon the PV panel surface. Currently, Mask R-CNN [16] is the most effective method in doing that task. However, that requires immense computation power and multiple layers of convolution.

Our model architecture can be further improved by adding more layers of convolution with greater filter sizes which can specifically target the PV panel surface only in an image.

Reference

[1] Solar power in Bangladesh [online]

https://www.weforum.org/agenda/2020/01/bangladesh-solar-power-energy-grid-rural-life/

[2] Air pollution in Bangladesh [online]

https://www.thedailystar.net/backpage/news/bangladesh-air-worst-world-1872904

[3] Sharma V, Chandel SS. Performance and degradation analysis for long-term reliability of solar photovoltaic systems: a review. Renew Sustain Energy Rev 2013; 27:753–67

[4] Solar power prospects [online] https://www.un.org/en/chronicle/article/achieving-sustainable-energy-targets-bangladesh

[5] Machine Learning [online] https://en.m.wikipedia.org/wiki/Machine_learning

[6] Machine learning [online] https://www.sas.com/en_us/insights/analytics/machine-learning.html

[7] Supervised and unsupervised learning [online]

https://machinelearningmastery.com/supervised-and-unsupervised-machine-learning-algorithms/

[8] Gupta, D. (2018). Architecture of convolutional neural networks (CNNs) demystified. Analytics Vidhya.

[9] Ke, Q., Liu, J., Bennamoun, M., An, S., Sohel, F., & Boussaid, F. (2018). Computer Vision for Human–Machine Interaction. Computer Vision for Assistive Healthcare,127-145.

[10] Relationship of Deep learning with Machine Learning and Artificial Intelligence [online] https://en.m.wikipedia.org/wiki/Deep_learning

[11] Data (images of PV panel) *S. Mehta, A. P. Azad, S. A. Chemmengath, V. Raykar and S. Kalyanaraman, DeepSolarEye: Power Loss Prediction and Weakly Supervised Soiling Localization via Fully Convolutional Networks for Solar Panels," 2018 IEEE Winter Conference on Applications of Computer Vision (WACV), Lake Tahoe, NV, 2018, pp. 333-342.*

[12] Singh, A. (2020). Demystifying the Mathematics Behind Convolutional Neural Networks. Analytics Vidhya.

[13]Hussain, A., Batra, A. & Pachauri, R. An experimental study on effect of dust on power loss in solar photovoltaic module. *Renewables* **4,** 9 (2017).

[14] Thomas Wood. (n.d.). What is the Softmax Function? https://deepai.org/machine-learning-glossary-and-terms/softmax-layer

[15] Peltarion. Categorical Crossentropy. (n.d.). https://peltarion.com/knowledge-center/documentation/modeling-view/build-an-ai-model/loss-functions/categorical-crossentropy

[16] Kaiming He, Georgia Gkioxari, Piotr Dollár, Ross GirshickKaiming He, Georgia Gkioxari, Piotr Dollár, Ross Girshick. 2017. Mask R-CNN

[17] Pulkit Sharma. 2019. https://www.analyticsvidhya.com/blog/2019/04/introduction-image-segmentation-techniques-python/

[18] Simple guide to confusion matrix terminology. (n.d.). https://www.dataschool.io/simple-guide-to-confusion-matrix-terminology/

[19] Image data preprocessing. (n.d.). https://keras.io/api/preprocessing/image/

[20] Promising soiling mitigation solutions should not be left to gather dust. (n.d.). https://www.pv-magazine.com/2019/10/24/promising-soiling-mitigation-solutions-should-not-be-left-to-gather-dust/