**DUST DETECTION ON SOLAR PV PANELS USING CONVOLUTIONAL NEURAL NETWORKS**

## 1.0 Introduction

The use of solar photovoltaic (PV) panels to generate electricity have become increasingly favourable today, since they produce zero carbon emissions (IEA, 2020). The presence of dust on the surface of a solar PV panel can significantly impact the efficiency of a solar system, thereby causing a drop in the energy output (Tarigan, 2019). **Figure 1** shows the impact of dust on solar PV panels and the impact on the output power. At a solar radiation intensity of 805 W/m2, the clean panel can generate a maximum output power of 113.04 W, while the dusty panel can only produce 73.26 W (Qdah et al., 2019).

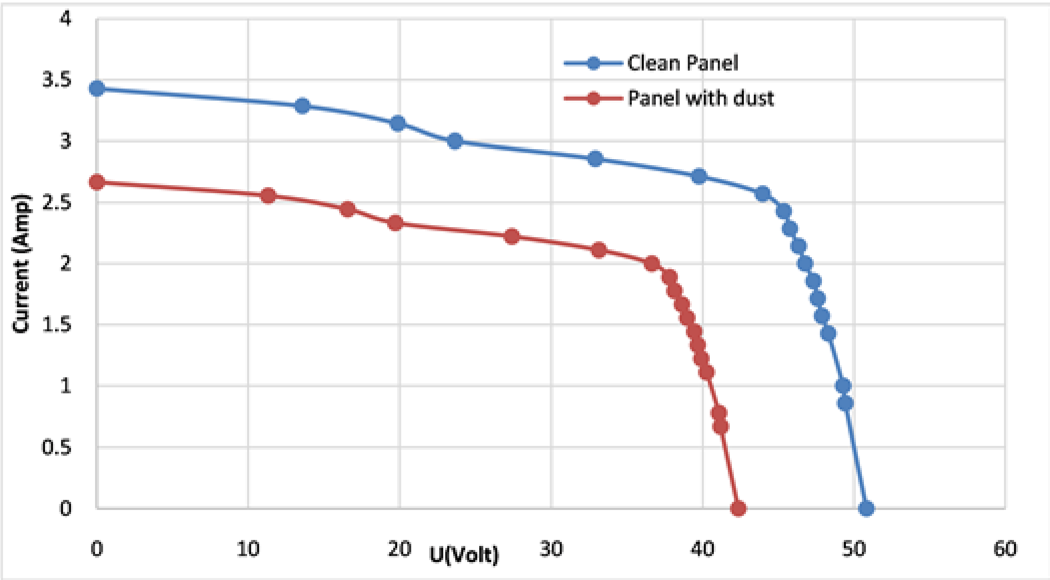



**Figure 1**. *Measured current and voltage characteristics input power of 805 W/m2 solar radiation intensity. Where output power P is the multiplication of current and voltage (Qdah et al., 2019).*

These challenges may be more impactful at solar farms where we have series of PV panels, it is essential to develop effective methods for detecting and possibly removing dust from solar PV panels to optimize their performance and build better confidence for existing and potential users of solar energy sources.

When there is a drop in the output power due to dust particles on solar panels, there may be a need to generate more electricity from a conventional non-renewable energy source, resulting to an increase in carbon emission (Tripathy, 2016).

Convolutional neural networks (CNN) which is a type of deep learning algorithm can be used to detect dust on solar PV panels. This approach makes use of images as input data, then identifies patterns for learning. I believe the use of CNN for dust detection on solar PV panels has the potential to improve both the efficiency and performance of a solar PV system, thereby impacting positively in our world.

## 2.0 BACKGROUND

A net zero scenario can be achieved when the release of carbon emission into the atmosphere is balanced with its removal, this can be achieved through the help of renewable energy sources (IPCC, 2018).

In 2021, there was a significant increase in power generation from solar PV, reaching a record high of 179 TWh and experiencing a 22% growth compared to 2020. To achieve the Net Zero Scenario target of producing about 7,400 TWh of solar PV energy per year by 2030, an average annual growth rate of approximately 25% will be necessary between 2022 and 2030 (Bojek, 2021).

Dust accumulation on solar photovoltaic (PV) panels results in losses, which poses a challenge in attaining a future state where net-zero emissions are achieved.

### 2.1 Related Works

Few related works were used for this project which are aimed at detecting dust on solar PV panels using CNN models.

Tan et al. (2019) proposed a denoising convolutional neural network (CNN) based approach for evaluating dust accumulated on PV panels. The process involves capturing images of the panels using a digital camera, then pre-processing the images to remove noise on the solar panel, before using denoised images as input to a CNN model for classifying the accumulation dust. Three pretrained CNN models with the involvement of transfer learning were used in this paper namely ResNet, VGG and AlexNet.

Maity et al. (2020) also maintained a CNN-based approach, but rather focused on predicting the associated power loss. The approach involved capturing RGB images of the panels, pre-processing the images to remove noise on the solar panel before then normalizing the data. A CNN model called Lenet was used to classify the images as being dusty or not, before then predicting the power loss based on the amount of dust detected.

Onim et al. (2022) on the other hand made use of various fine-tuned pre-trained models such as VGG16, InceptionV3, Resnet50, and AlexNet, along with a proposed model called SolNet which also uses CNN. The approach also involves capturing images of the panels, pre-processing the images to remove noise before normalization, and then using a CNN model to classify the images as dusty or clean. The authors of this paper claimed that their approach achieved higher accuracy than existing methods and could potentially be used for real-time monitoring of solar panels.

The last work was by Cipriani et al. (2020) who's approach involved capturing thermal images of the PV panels, with a similar approach as the latter, but then they used a CNN model to classify the images as dusty, hotspot, or clean. In this paper, no pre-trained model was used although there were various augmentation techniques used to improve performance.

For my project, I experimented on different computer vision (CV) colour models such as HSV, Grayscale, and RGB, against different pre-trained models such as VGG16 (Simonyan and Zisserman, 2014), InceptionV3 (Szegedy et al., 2016), Resnet50 (He et al., 2016), and MobileNet (Howard et al., 2019). The main goal is to compare the performance of different combinations to see if the colour mode has a significant effect on the performance of the models since it was not considered in the related works.

### 2.2 Aims and Objectives

The aim of this project is to take an alternative approach to the previous related works by exploring different CV colour models with a set of pre-trained models with the application of transfer learning to achieve a result that best enhance the detection of dust on solar panels at any given condition.

Our baseline models for comparison will be the of fine-tuned pretrained models using RBG colour mode since it is the default CV colour model used in the related works.

I will also be exploring few hyperparameter tunings to see if we can better improve a selected set of trained models before then finally testing with new image samples.

To achieve optimal result, the complexity which involves the number of parameters for each model will also be considered alongside the performance.

## 3.0 Methodology

In this section, we described the process of analysing data and developing the most effective combination of a CNN pretrained model and various CV colour models for detecting dust on solar PV panels. **Figure 2** shows the process breakdown for our project.

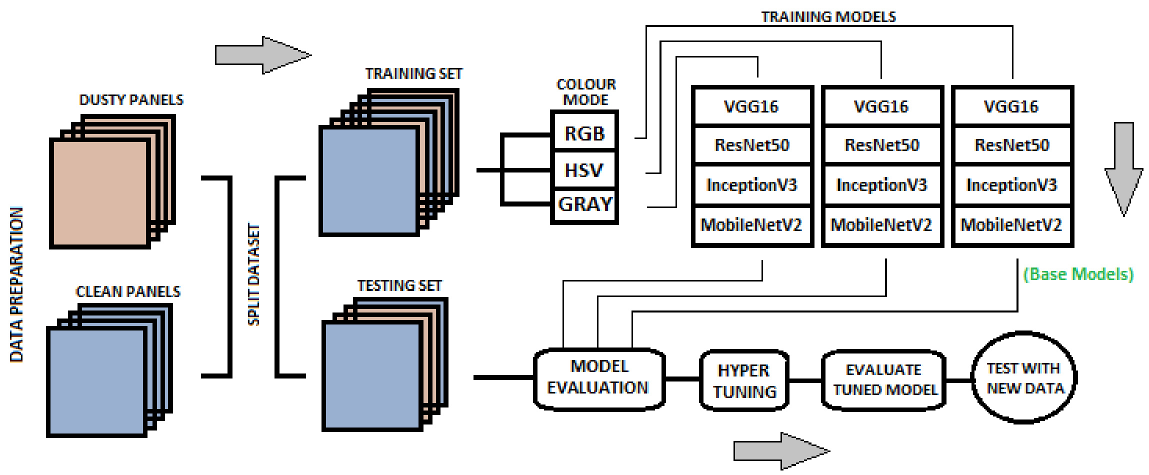



**Figure 2.** *Process diagram.*

During data preparation we need to ensure that the right data are well categorised in other to avoid poor classification and prediction during training and evaluation, respectively. Splitting our data will be done to enable us to evaluate our model while training. In this project, we made use of three CV colour mode RGB, HSV and Gray Scale. Each of these colour model passed through various pretrained models namely VGG16 (Simonyan and Zisserman, 2014), ResNet50 (He et al., 2016), InceptionV3 (Szegedy et al., 2016) and MobileNetV2 (Howard et al., 2019). We then evaluated and processed further using hyperparameter tuning before finally testing the best performing model with new data.

### 3.1 Dataset Description

The image dataset used for this project was gotten from an open source on kaggle (Sai, 2022). These images have two categories which were split using a directory. Due to the source of the images being scraped from the internet, I saw it necessary to investigate and clean the data before proceeding further. The **Table 1** shows the splitting of the dataset into training and testing. We can also see in **Figure 3** a sample of a clean and dusty panel.

**Table.1** *Split Dataset into Training and Testing.*

| Image Split | Clean PV Panel | Dusty PV Panel |
|---|---|---|
| Train Images | 865 | 733 |
| Test Images | 191 | 191 |
| Total Images | **1056** | **924** |


**Figure 3.** *A sample of both clean panel and dusty panel.*

### 3.2 Computer Vision (CV) Colour Mode

The CV colour mode is important, and we will be looking at the Red, Green & Blue mode (RGB), the Grayscale mode and the Hue, Saturation, and Value mode (HVS). In **Figure 4** below, we can see the image representation of the three colour models used for this project.

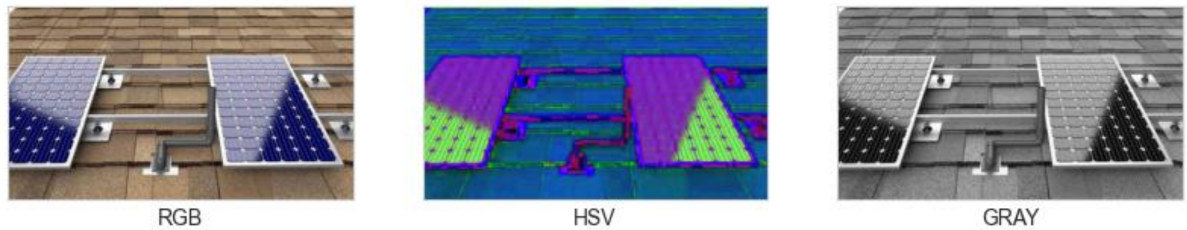



**Figure 4.** *Image sample of the HSV, RBG and Grayscale mode.*

While selecting a CV colour mode, it is important to know the number of channels involved. The RGB and HSV both have 3 channels with different colour attributes. Unlike the RGB and HSV, the grayscale has only a single channel. For us to maintain a uniform number of channels, we will need to convert the grayscale into a 3-channel image which will involve replicating the same grayscale output value in all three channels (Yamini et al., 2020).

### 3.3 Data Augmentation

Our total training dataset is slightly above 1500 images which I believe to be a low volume for training a CNN model. The aim of data augmentation for this project is to help increase the size of the training data set by adding new variation of the existing dataset, thereby improve the performance of the model by reducing the changes of over fitting.

Image Data Generator was used for both data augmentation and data processing. Since we split the category of images in our dataset in a directory folder, we made use of the flow from directory method to load and process the image. The Image Data Generator also enabled us to pass in a pre-processing function which takes in the CV colour mode which will be used for training.

### 3.4 Convolutional Neural Network (CNN) Model Breakdown

Convolutional Neural Network (CNN) consist of two stages, the feature learning stage and the classification stage. In this project we will be using pretrained models where we will be applying transfer learning within the second stage of the models. **Figure 5** shows the breakdown of a CNN model.

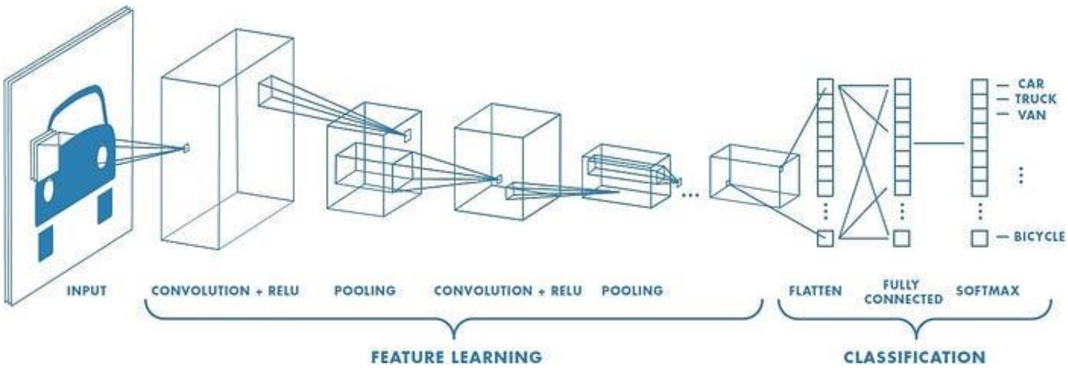



**Figure 5.** *Convolutional Neural Network Architecture (Saha, 2018).*

### 3.4.1 VGG16 Model

The VGG16 model by Simonyan and Zisserman (2014) has a large network with 13 convolutional layers at the feature learning stage and 3 fully connected layers at the classification stage which sums up to 16 layers.

This pretrained model uses a uniform and simple hyperparameters configuration consisting of a (3x3) filter with a stride of 1. Also, using a max pooling of (2 x 2) with a stride of 2 is maintained to ensure all layers of the previous convolution were inspected (Ng, 2020).

The filter is used to extract features from the input image, with a single stride and same padding the dimension remains the same as the input dimension. For the case of the max pooling, it helps to identify the highest feature within a convolutional network by using smaller dimensions.

### 3.4.2 ResNets50 Model

These network by He et al. (2016) are built from a residual block which uses a skip connection to train very deep networks with numerous layers. From the **Figure 6** we can see a breakdown of a residual block where we made use of both linear function and a ReLU activation function.

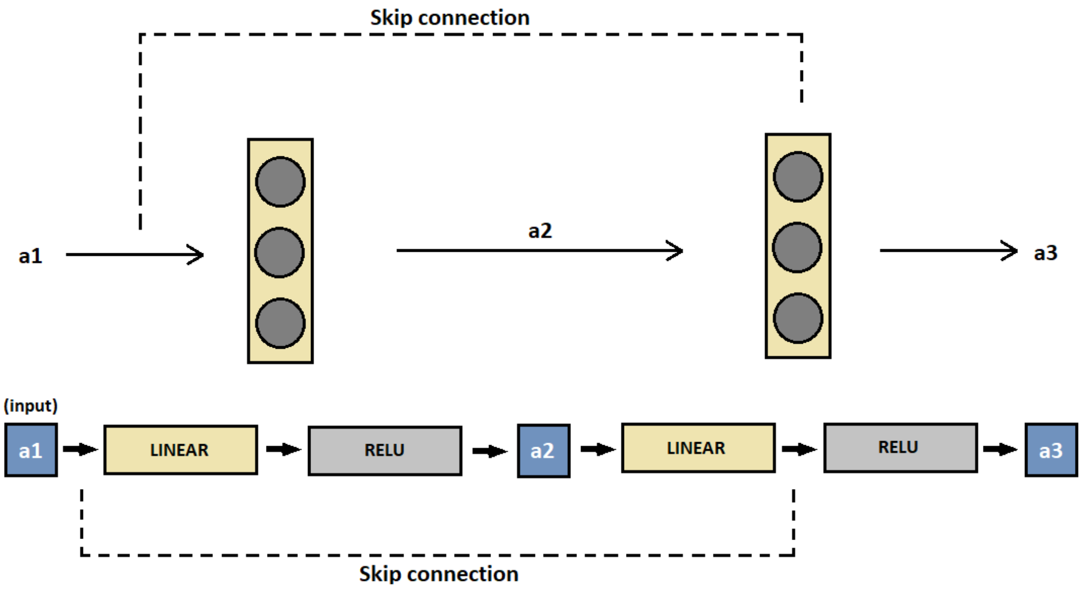



**Figure 6.** *Residual block breakdown structure (Ng, 2020).*

Here are some equations to consider, where the ReLU activation function which returns the linear function output (z) if positive else returns 0 as shown in equation (1) (Ng, 2020).

The mathematical representation for a single residual block is shown in equation (6) (Ng, 2020).

$$g(z) = \max(0, z) \tag{1}$$

$$z^{[2]} = w^{[2]} a^{[1]} + b^{[2]} \tag{2}$$

$$a^{[2]} = g(z^{[2]}) \tag{3}$$

$$z^{[3]} = w^{[3]} a^{[2]} + b^{[2]} \tag{4}$$

$$a^{[3]} = g(z^{[3]}) \tag{5}$$

$$a^{[3]} = g(z^{[3]} + a^{[1]}) \tag{6}$$

For us to construct a ResNet model, we will combine multiple residual blocks and arrange them in stacks to form a deep network.

#### 3.4.3 Inception Model

In the case of inception pretrained model by Szegedy et al. (2016), rather than selecting a specific filter size for a Convolutional layer or deciding between using a Convolutional layer or a Pooling layer, we can include all these options in our model (Ng, 2020). The issues faced using this model in this form is the computational cost. This computational cost for this model is resolved using a (1x1) convolution of (n) number which creates a bottle neck layer with (n) channel, where (n) is less than the input channel as shown in **Figure 7** (Ng, 2020).

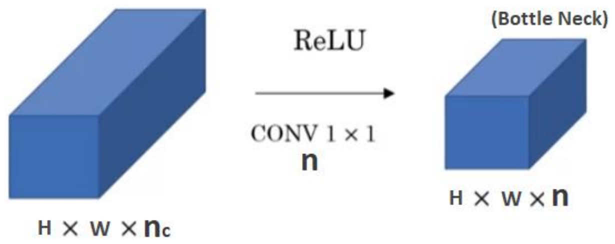



**Figure 7**. *(1 x 1) Convolution diagram*, Where $n < n_c$ (Ng, 2020).

#### 3.4.4 MobileNet Model

This model by Howard et al. (2019) has a low computational cost which can be used with less powerful devices including mobile phones. The Mobile makes use of a depth wise separable convolution, consisting of a depth wise convolution followed by a point wise convolution.

The depth wise convolution uses a filter with no channel which outputs a convolution layer with no channel. A pointwise convolution is then applied to the output from the depth wise to add channels to the convolution.

## 4.0 Experiment

In our methodology we investigated the datasets used for this project and looked at the importance of data augmentation in improving the performance of the model on a small dataset. In addition, we will be discussing on how we can experiment using nested loops iterations within a function to investigate the performance of the combination of the CV colour mode and the pre-trained models where transfer learning was applied to meet our desired output. After experimenting on all combinations of the models, we will explore on various hyperparameter tuning for the top performing models.

#### 4.1 Model Experimentation

For us to achieve our desired result of various model combinations, we made use of a function which consist of a nested loop. The Image Data Generator iterating through different colour modes were within the first loop of the function, while four pre-trained models were in the second loop as show in **Figure 8**.

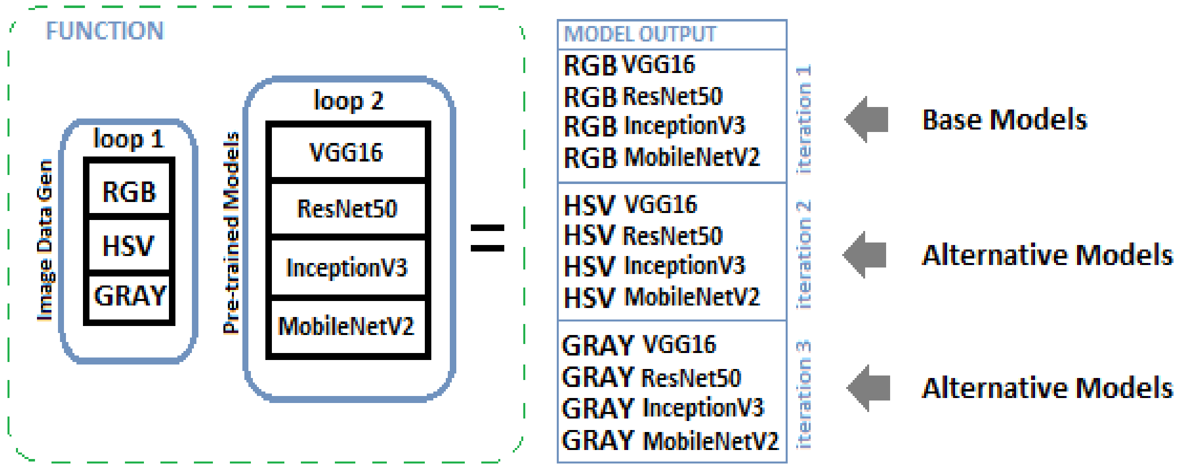



**Figure 8.** *Model function consisting of a nested loop.*

During data pre-processing, we rescaled our input shape to help standardize and normalize our data during training. In our selected pre-trained models, transfer learning was done at the output layer, and we maintained the same input shape value of (150 x 150 x 3) to avoid mismatch. The transfer learning architecture was similar across all models which consist of a single flattening layer, two fully connected layers of same size (2048) with ReLU activation functions, a dropout of (0.2) and a binary output with a SoftMax activation function.

Equation (7) by Wood (2019) shows the SoftMax activation function where **Z**i are the input vector values and K is the number of classes which in this case is two. A sigmoid activation function is equivalent to a two class SoftMax activation function (Wood, 2019).

$$\sigma(\vec{z})_i = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}} \tag{7}$$

The dropout help reduce the complexity of the model to avoid overfitting. I used a low value of (0.2) since we have not trained our model. The fully connected layer size of 2048 was selected due to the rescaling of the input shape.

Before fitting our model, a check point was implemented to save the best model from each of the 12 iterations. We were able to display a data frame which shows the model performance, the number of parameters in the model architecture, the saved path directory, and the rating of each saved model performance. When fitting our model, we maintained an epoch of 10 with a step of 10 at this stage before applying hyperparameter on fewer models. **Table 2** shows the output of the 12 models.

**Table.2** *Details of the model performance and complexity for each combination.*

| | Models | Accuracy | Loss | Parameters | Path | Rating |
|---|---|---|---|---|---|---|
| 0 | VGG with RGB | 0.863 | 0.340 | 35694402 | D:/Applied AI (save)/model_VGGRGB.h5 | 96.0 |
| 1 | ResNet with RGB | 0.674 | 0.616 | 132647810 | D:/Applied AI (save)/model_ResNetRGB.h5 | 25.0 |
| 2 | Inception with RGB | 0.802 | 0.435 | 63754018 | D:/Applied AI (save)/model_InceptionRGB.h5 | 73.0 |
| 3 | Mobilenet with RGB | 0.863 | 0.357 | 71996482 | D:/Applied AI (save)/model_MobilenetRGB.h5 | 94.0 |
| 4 | VGG with HSV | 0.771 | 0.508 | 35694402 | D:/Applied AI (save)/model_VGGHSV.h5 | 57.0 |
| 5 | ResNet with HSV | 0.582 | 0.684 | 132647810 | D:/Applied AI (save)/model_ResNetHSV.h5 | 0.0 |
| 6 | Inception with HSV | 0.735 | 0.559 | 63754018 | D:/Applied AI (save)/model_InceptionHSV.h5 | 44.0 |
| 7 | Mobilenet with HSV | 0.790 | 0.595 | 71996482 | D:/Applied AI (save)/model_MobilenetHSV.h5 | 48.0 |
| 8 | VGG with GRAY | 0.848 | 0.446 | 35694402 | D:/Applied AI (save)/model_VGGGRAY.h5 | 79.0 |
| 9 | ResNet with GRAY | 0.643 | 0.622 | 132647810 | D:/Applied AI (save)/model_ResNetGRAY.h5 | 19.0 |
| 10 | Inception with GRAY | 0.838 | 0.385 | 63754018 | D:/Applied AI (save)/model_InceptionGRAY.h5 | 86.0 |
| 11 | Mobilenet with GRAY | 0.872 | 0.326 | 71996482 | D:/Applied AI (save)/model_MobilenetGRAY.h5 | 100.0 |

The rating of the performance was derived from using mean max scaler both for the accuracy and loss. The equation (8) below shows the rating formula R.

$$\text{Rating (R)} = \frac{\text{Scaled Accuracy} + (1 - \text{Scaled Loss})}{2} \tag{8}$$

Before we applied hyperparameter, we created a function that returns the top ranked models with a visualized output as shown in **Figure 9.**

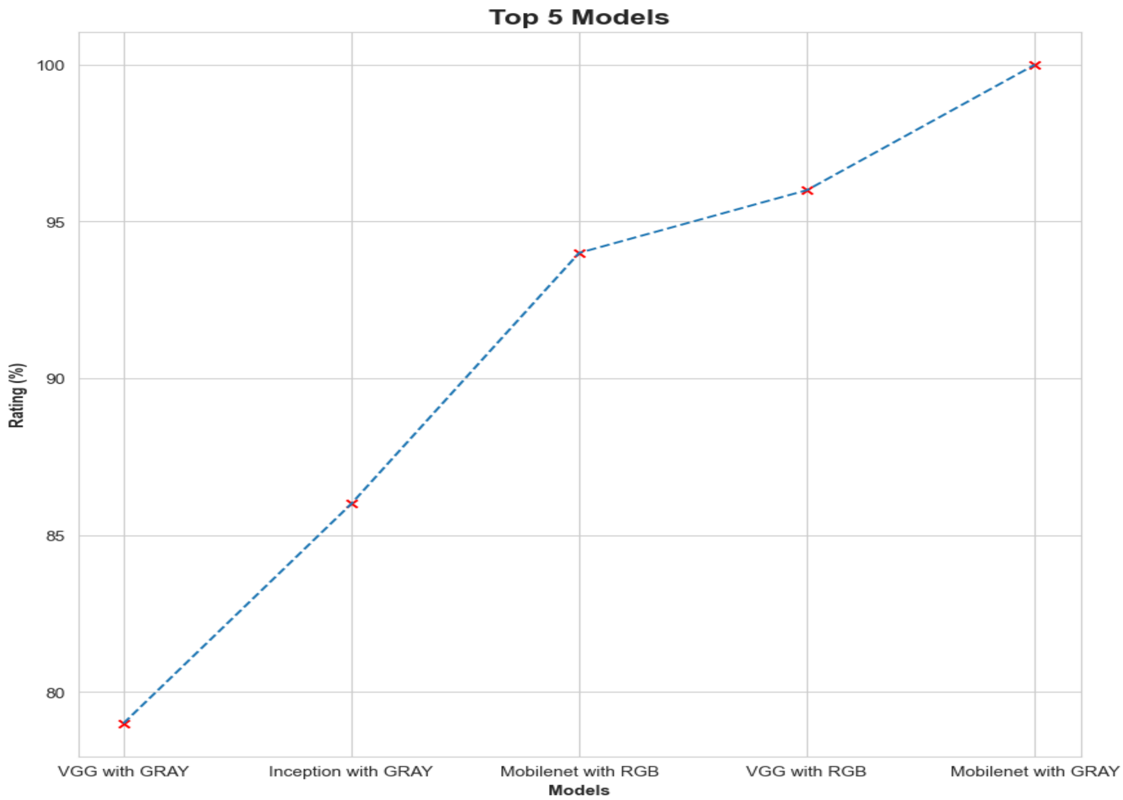



**Figure 9.** *Top 5 Best Performing Models.*

### 4.2 Hyperparameter Tuning (HPT)

We will be applying hyperparameter tuning on the best two models, a base model (VGG with RGB) and an alternate model (MobileNet with Grayscale) both ranked second and first respectively as shown in **Figure 9**.

For our hyperparameter tuning, we made use of the keras turner library function (Keras Team, 2019). This function iterates through different combination of tuning parameters such as the number of input connections per layer, the learning rate of selected activation function, and the dropout value.

Since we used transfer learning on the pretrained models, we will only be altering the classification stage of out model architecture where we have the fully connected layers. **Table 3** shows the best tuned combination of the classification stage for both selected models.

**Table 3.** *Best Parameter selection after tuning***.**

(a) VGG with RGB

| Best Value So Far | Hyperparameter |
|---|---|
| 1952 | unit_1 |
| 2080 | unit_2 |
| 0.4 | dropout_rate |
| 0.0001 | learning rate |
| 10 | tuner/epochs |
| 4 | tuner/initial_epoch |
| 1 | tuner/bracket |
| 1 | tuner/round |

(b) MobileNet with Grayscale

| Best Value So Far | Hyperparameter |
|---|---|
| 160 | unit_1 |
| 2080 | unit_2 |
| 0.2 | dropout_rate |
| 0.0001 | learning rate |
| 10 | tuner/epochs |
| 4 | tuner/initial_epoch |
| 2 | tuner/bracket |
| 2 | tuner/round |

## 5.0 Results

In our experiment we were able to investigate on four base line models against two sets of alternative models. We were also able to apply hyper parameter tune on the best of both sides.

In this section, we will look at the general performance of all models, then analyse the selected best two models from the experiment before visualizing their performance using a confusion matrix. Finally, we will test our models with new images.

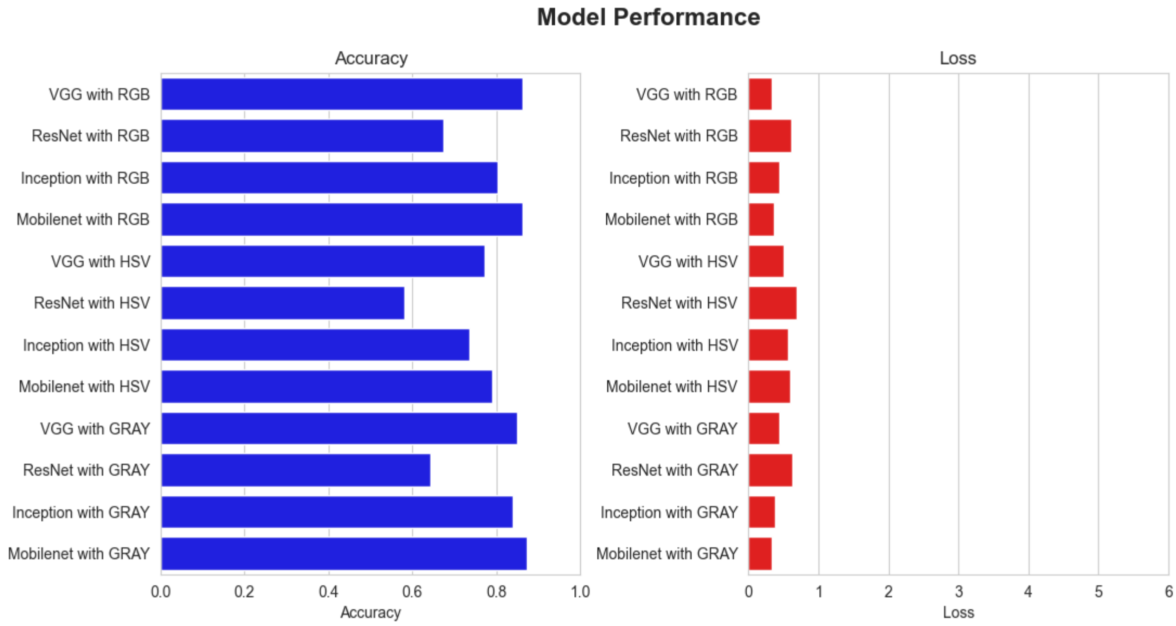
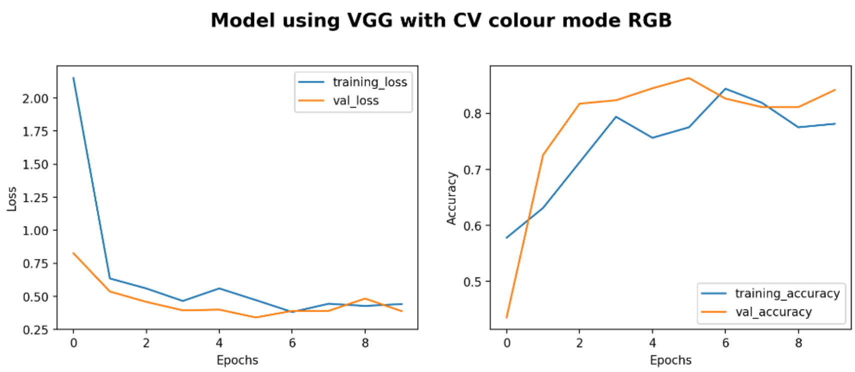


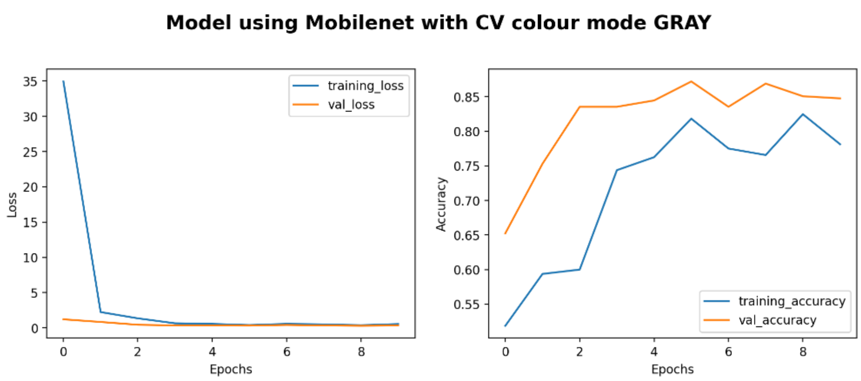

**Figure 10.** *Model Performance visualization.*

We will Start by looking at all the model performance as shown in **Figure 10**. The ResNet model seem to do poorly in comparison to the rest of the models. We can also see that the RGB and the Grayscale CV colour mode performed well.

Before Hyperparameter tuning, **Figure 11** shows us the loss and accuracy of the best two model while training, consisting of a base and an alternative model.




(a)
VGG with RGB performance.
The Peak Validation loss is **0.340** while the accuracy is **0.863**.

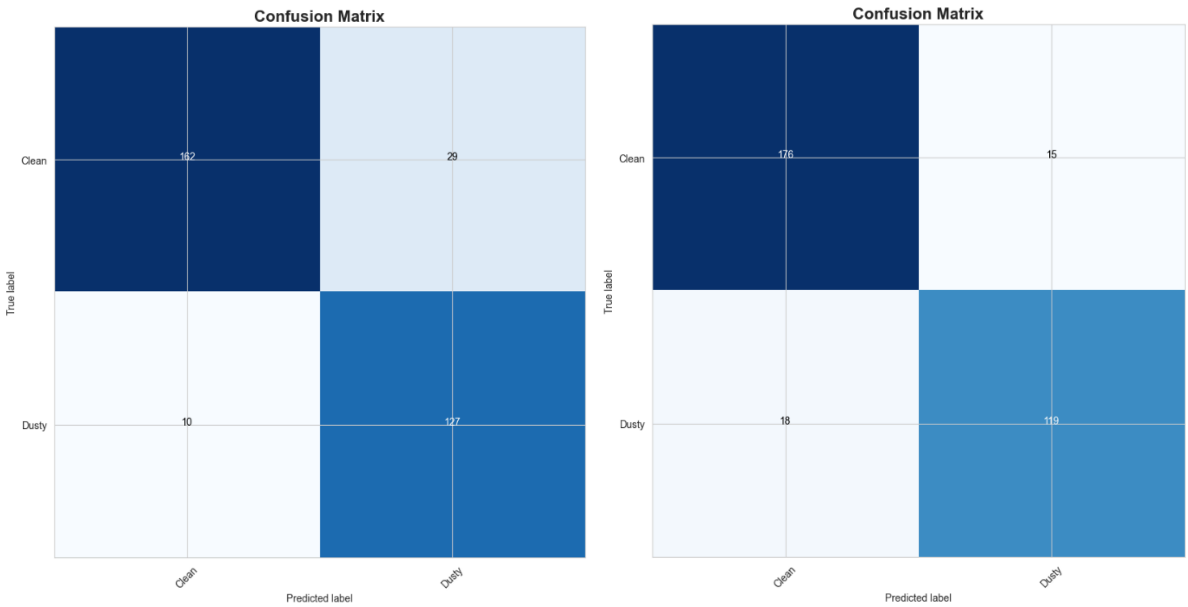



(b)
MobileNet with Grayscale performance.
The Peak Validation loss is **0.326** while the accuracy is **0.872**.

**Figure 11.** *Loss and accuracy performance during training.*

A confusion matrix was used to evaluate our performance after applying hyperparameter tuning for both models. This can be seen in **Figure 12** and **Table 4** below.

The precision performance will be used to evaluate our models based on the accuracy of the positive cases for both the clean and dusty panels. We can calculate our precision using equation (9).




(a) VGG with RGB (b) MobileNet with Grayscale

**Figure 12.** *Model evaluation performance using confusion metrics.*

**Table 4.** *Confusion metrics precision performance.*

| | **BASE MODEL** | | **ALTERNATE MODEL** | |
|---|---|---|---|---|
| | **CLEAN** | **DUSTY** | **CLEAN** | **DUSTY** |
| **CLEAN** | **162** | **29** | **176** | **15** |
| **DUSTY** | **10** | **127** | **18** | **119** |
| **PRECISION** | **94%** | **81%** | **91%** | **89%** |

$$Precision = \frac{True\ Positive}{True\ Positive + False\ Positive} \tag{9}$$

While testing we used 8 new images in our models and both models predicted 100% correctness as shown in **Figure 13**.


**Figure 13**. *Test Prediction with new sample data.*

From our result we have shown that an alternate CV colour mode can performance better in some model architecture. We also noticed that the complexity of a model architecture is not a determining factor to its performance.

Comparing our performance to the related works, our dataset had different types of solar PV panel images with different background noises such as the environment and houses unlike the latter. This added more complexity during training thereby affecting performance. The good side is that our model is more robust and applicable in real-world scenarios.

By adding more data, we can improve the performance of our model. It is also worth investigating on more computer vision colour mode.

## 6.0 Conclusion

We were able to see the importance of improving solar (PV) panel power efficiency and how CNN models can help actualize that. We then experimented on the RBG, HSV and Grayscale CV colour models and how it impacts the performance of several fine-tuned CNN models.

To conclude this report, it will be encouraging to see more related works that uses different CV colour models with a much larger dataset. This can help better improve solar power efficiency through (PV) panel dust detection, thereby achieving the Net Zero Scenario target to combat carbon emission.

## Reference

Bojek, P. (2021) *Solar PV – Analysis*. IEA. Available online: https://www.iea.org/reports/solar-pv.

Cipriani, G., D'Amico, A., Guarino, S., Manno, D., Traverso, M. and Di Dio, V. (2020) Convolutional Neural Network for Dust and Hotspot Classification in PV Modules. *Energies*, 13(23), 6357.

He, K., Zhang, X., Ren, S. and Sun, J. (2016) *Deep Residual Learning for Image Recognition*. openaccess.thecvf.com. Available online: https://openaccess.thecvf.com/content_cvpr_2016/html/He_Deep_Residual_Learning_CVPR_2016_paper.html.

Howard, A., Sandler, M., Chu, G., Chen, L.-C., Chen, B., Tan, M., Wang, W., Zhu, Y., Pang, R., Vasudevan, V., Le, Q.V. and Adam, H. (2019) *Searching for MobileNetV3*. openaccess.thecvf.com. Available online: https://openaccess.thecvf.com/content_ICCV_2019/html/Howard_Searching_for_MobileNetV3_ICCV_2019_paper.html.

IEA (2020) *Renewables 2020 – Analysis*. IEA. Available online: https://www.iea.org/reports/renewables-2020.

IPCC (2018) *Summary for Policymakers — Global Warming of 1.5 ºC*. Ipcc.ch. Available online: https://www.ipcc.ch/sr15/chapter/spm/.

Keras Team (2019) *Keras documentation: KerasTuner*. keras.io. Available online: https://keras.io/keras_tuner/.

Maity, R., Shamaun Alam, Md. and Pati, A. (2020) An Approach for Detection of Dust on Solar Panels Using CNN from RGB Dust Image to Predict Power Loss. *Cognitive Computing in Human Cognition*, 41–48.

Ng, A. (2020) *Convolutional Neural Networks*. Coursera. Available online : https://www.coursera.org/learn/convolutional-neural-networks.

Onim, M.S.H., Sakif, Z.M.M., Ahnaf, A., Kabir, A., Azad, A.K., Oo, A.M.T., Afreen, R., Hridy, S.T., Hossain, M., Jabid, T. and Ali, M.S. (2022) SolNet: A Convolutional Neural Network for Detecting Dust on Solar Panels. *Energies*, [online] 16(1), 155. Available online : https://www.mdpi.com/1996-1073/16/1/155/pdf#:~:text=SolNet%2C%20a%20CNN%20architecture%20that.

Qdah, K.S.A., Abdulqadir, S.A., Harbi, N.Y.A., Soqyyah, A.Z., Isa, K.J., Alharbi, M.Y. and Binsaad, N.M. (2019) Design and Performance of PV Dust Cleaning System in Medina Region. *Journal of Power and Energy Engineering*, [online] 07(11), 1–14. Available online: https://www.scirp.org/journal/paperinformation.aspx?paperid=96319.

Saha, S. (2018) *A Comprehensive Guide to Convolutional Neural Networks — the ELI5 way*. Towards Data Science. Available online: https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53.

Sai, H. (2022) *Solar Panel dust detection*. www.kaggle.com. Available online: https://www.kaggle.com/datasets/hemanthsai7/solar-panel-dust-detection.

Simonyan, K. and Zisserman, A. (2014) *Very Deep Convolutional Networks for Large-Scale Image Recognition*. arXiv.org. Available online: https://arxiv.org/abs/1409.1556.

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J. and Wojna, Z. (2016) *Rethinking the Inception Architecture for Computer Vision*. openaccess.thecvf.com. Available online: https://openaccess.thecvf.com/content_cvpr_2016/html/Szegedy_Rethinking_the_Inception_CVPR_2016_paper.html [Accessed 6 Apr. 2023].

Tan, Y., Liao, K., Bai, X., Deng, C., Zhao, Z. and Zhao, B. (2019) *Denoising Convolutional Neural Networks Based Dust Accumulation Status Evaluation of Photovoltaic Panel*. IEEE Xplore. Available online: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8791298 [Accessed 4 Apr. 2023].

Tarigan, E. (2019) The Effect of Dust on Solar PV System Energy Output Under Urban Climate of Surabaya, Indonesia. *Journal of Physics: Conference Series*, 1373(1), 012025.

Tripathy, M., Sadhu, P.K. and Panda, S.K. (2016) A critical review on building integrated photovoltaic products and their applications. *Renewable and Sustainable Energy Reviews*, 61, 451–465.

Wood, T. (2019) *Softmax Function*. DeepAI. Available online: https://deepai.org/machine-learning-glossary-and-terms/softmax-layer#:~:text=Softmax%20Formula&text=where%20all%20the%20zi%20values.

Yamini, K., Sai Swetha, K., Prasanna, P., Rupa, M., Swathi, V. and Rao Maddumala, V. (2020) IMAGE COLORIZATION WITH DEEP CONVOLUTIONAL OPEN CV. *Journal of Engineering Science*, [online] 11(4). Available online: https://jespublication.com/upload/2020-110472.pdf [Accessed 17 Apr. 2023].