COMP9517 Group Alloc 7

Junhyeok Lee
UNSW
z5267156
Sydney Australia
z5267156@unsw.edu.au

Madeleine Galicia
UNSW
z5309597
Sydney Australia
z5309597@unsw.edu.au

Prabhnoor Kahlon
UNSW
z5420174
Sydney Australia
z5420174@unsw.edu.au

Zimeng Kuang
UNSW
z5434038
Sydney Australia
z5434038@unsw.edu.au

Zhang Qixiang
UNSW
z1234
Sydney Australia
z54097169@unsw.edu.a

I. Introduction

The increasing adoption of solar panels for off-grid energy solutions necessitates robust methods for assessing and maintaining their efficiency. This report delves into the development and evaluation of computer vision techniques aimed at predicting the health of photovoltaic (PV) cells using Electroluminescence (EL) imaging of solar modules. As a response to the challenges posed by damages from environmental elements or manufacturing errors, this project focuses on leveraging the ELPV dataset—a collection of 2,624 EL images with varying degrees of degradation—to classify cell images based on the probability of defectiveness.

By exploiting advanced technology like EL imaging, capable of visualising a broad spectrum of imperfections in PV modules, this project aims to automate the detection and classification of defects. The dataset, normalised and annotated with defect probabilities ranging from 0% to 100%, serves as the foundation for training and evaluating computer vision classifiers. This report outlines the development and testing of novel methodologies, combining principles from our coursework with insights from the literature, to advance the field of automated defect classification in solar panels.

II. Literature Review

A. Support Vector Machine (SVM)

SVM is a simple algorithm that is used in machine learning and can be used for classification and regression tasks, however it is mostly used for classification. The main task of the SVM algorithm is to attain an optimal hyperplane in a dimensional space that is determined by the number of input features to classify data points inputted.

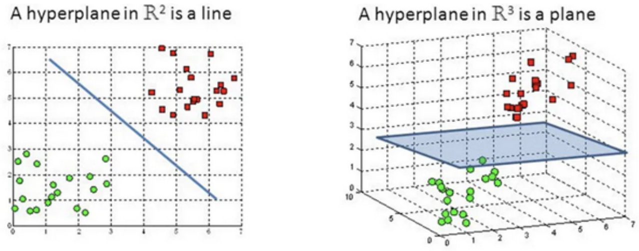


Figure 1:Hyperplane in a 2D and 3D space [1]

A hyperplane is the periphery that is used to define classes. The data points that are on different sides of the hyperplane can be sorted into separate classes. Another important concept related to SVM are support vectors. Support vectors are defined as the data points that define the hyperplane. These support vectors are the most difficult points to define and are directly related to creating an optimal hyperplane. Removing these data points from the training set should alter the position of the hyperplane. Lagrange multipliers can be used to solve the problem of finding the optimal hyperplane in an efficient and fast manner. For the hyperplane to be considered optimal, it must have the maximum margin that is equidistant to both classes. The margin is designed by the support vectors. It must be able to attain maximum classification, which requires the hyperplane to classify all data points into the class it belongs to. Additionally, it must be able to separate all the data points. For example, for two classes, it must not undergo a situation when a data point is on the line, which prevents it from being sorted into either class. [2]

The general input for SVMs is pairs of input and output training samples. In a typical SVM, the amount of input data points will be large. The output of SVM will assign weights for each feature that has a linear combination that will predict the output training sample value. The differentiating factor of SVM is that the number of weights that are not zero will be reduced through the optimization of ensuring the hyperplane has the maximum margin. This will result in the remaining weights corresponding to support vectors only.

The advantages of SVM begin with it working efficiently when there is a clear distinction between classes. It can be used in high dimensional data sets and becomes more effective the higher the dimension. It is efficient when the number of data points is far lower than the number of dimensions. It is also memory efficient when compared to other algorithms. The first disadvantage of SVM is that it performs poorly when given large data sets and when the data classes overlap. When data points have a higher number of features than the training data samples, it performs poorly. Lastly, it is not a descriptive modelling tool and cannot provide descriptions of any relationships in the data. [3]

B. Random Forest

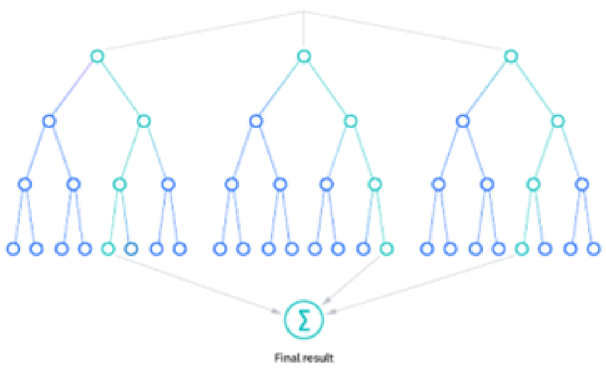


Figure 2: Random Forest diagram [4]

Random forest is a machine learning algorithm that uses the bagging method to create a "forest" of decision trees and is often used for classification and regression tasks. The aim is for the prediction to be constant and accurate due to merging multiple decision trees. [5] An important concept to understand how random forest works is bagging. Bagging is an algorithm that uses multiple models that are trained on different subsets of the input dataset. The final prediction is created by combining the outputs of these models together. The second important algorithm to understand random forests is decision trees. Decision trees are made up of decision nodes, which can be seen as a series of questions, which will divide the input data. This algorithm aims to effectively find the best way to separate the data. The final decision will be situated at the leaf nodes. Random forest has the similar hyperparameters as a bagging classifier and decision trees as it is a combination of the two that can also deal with regression tasks. It also provides randomness during the branching of the trees as it searches for feature splits and only selects a subset of those features. This is known as feature bagging which accounts for possible variability in the input data. The risk of variance, overfitting and overall variance is reduced. As a result, the predictions created by random forests are more precise. [4]

Random forest is defined by its three central hyperparameters. The first is the maximum number of features otherwise known as node size. This hyperparameter determines the number of features a random forest must consider before splitting a node. The second is the number of trees. This affects the amount of trees Random Forest creates before averaging the predictions for a regression task or doing a majority vote for classification tasks. The last is the lowest number of leaves required to separate an internal node. [5] The speed of the model can be altered by the number of processes it can use and its oob sampling. The advantages of using Random Forest are that it's easy to understand, versatile, doesn't overfit and has a stable prediction result. Its main disadvantage is that the speed is reduced when the trees are increased, and that accuracy relies on adding more trees. Additionally, it is not a descriptive modelling tool and cannot provide descriptions of any relationships in the data. [5]

C. Logistic Regression

Logistic regression is defined by the following formulas:

$$logit(\pi) = \frac{1}{1 + \exp(-\pi)} \tag{1}$$

$$ln\left(\frac{\pi}{1-\pi}\right) = \beta_0 + \beta_1 * x_1 + \cdots + \beta_k * K_k \tag{2}$$

where is the dependent variable and x is the independent variable. is an estimated variable using the method of maximum likelihood estimation. Then the variable is changed throughout the iterations of the test to get to a value that optimises the fit of the log odds. These iterations are compounded to create the log likelihood equation. The aim of logistic regression is to find the best variable which is done by maximising this equation. The conditional probabilities of the iterations used to find the optimal coefficient are logged and summed together. This yields a predicted probability. To determine the goodness of fit, the Hosmer-Lemeshow test is one of the most popular methods to assess the fit. [6]

Logistic regression is often compared to linear regression. The difference between the two is that logistic regression is used for classification, while linear regression is used for prediction. [7] Binary, multinomial, and ordinal logistic regression are the types of logistic regression models. Binary logistic regression only has two outcomes and is the most used logistic regression model. The second model has three or more outcomes, but the values have no specified order. Ordinal logistic regression also has three or more outcomes and these values do have a defined order. Additionally, it is a discriminative model which concentrates on the border between classes. [6]

The advantages of logistic regression start with it performing effectively when the dataset is linearly separable. When compared to other algorithms, it is less likely to be over-fitting. It is easy to implement and train. Additionally, it is easy for others to interpret and gives a measure relevant to a predictor and its association direction. The main disadvantage of logistic regression is that it assumes that the data's dependent and independent variables are linearly associated. Logistic regression will overfit if the number of observations is less than the number of features. Lastly, the algorithms can only be used to predict discrete function. [8]

D. OTSU Method

Otsu's method is a histogram-based thresholding technique that automatically finds the optimal threshold value to separate an image into foreground and background. It's commonly used in image segmentation tasks where you want to separate objects from the background. In the context of predicting the health of photovoltaic (PV) cells in electroluminescence (EL) images of solar modules, Otsu's method can play a crucial role in image analysis.

The Otsu algorithm aims to find the optimal threshold that minimises the variance within each class or segment and maximises the variance between classes or segments. Once the optimal threshold is determined, it is applied to the image. Pixels with intensities below the threshold are assigned to one class, and pixels above the threshold are assigned to another class. This results in a binary image where pixels are classified as either part of the object of interest or the background. [9]

Advantages of OTSU in PV Cell Health Prediction

This method automatically determines the threshold without requiring manual intervention. This is crucial for large-scale image analysis such as predicting the health of photovoltaic cells in electroluminescence images of solar modules as it would be impractical to implement thresholding manually. Otsu method is also known to adapt to the different characteristics of the image given as an input. This makes it suitable for different lighting conditions and variations that may occur in the electroluminescence images of solar modules. Due to the design of the Otsu method being based on statistical analysis, it has the tendency to minimise subjectivity in the thresholding process. This is a crucial point in order for the image processing to have a consistency in defect detection across different images and datasets. Electroluminescence images of solar modules have the tendency to appear darker for defective cells due to disconnected areas that are shown in the images and since the Otsu method thrives in an environment where segmentation and highlighting is one of the best methods to identify and isolate potential defects in PV cells. Otsu method allows rigorous analysis of the electroluminescence images in order to assess the severity of the defects that exist in the solar modules as it provides a quantitative measure of the optimal threshold. It is also known to be quite efficient when it comes to computational power, making it suitable for real-time or near-real-time analysis of large datasets, which is essential for monitoring the health of solar modules in a practical setting.

Disadvantages of OTSU in PV Cell Health Prediction

Otsu relies on the assumption that the image has a bimodal intensity distribution. What this means is that the contrast between the class and the subject must be highly distinguishable such that the algorithm is able to make a clear separation between the object and the background. In the case that the object and the intensity of the background is not clearly distinguished or if there are multiple intensity peaks in the image provided, the otsu method may not be able to perform well in comparison to the other methodologies that were mentioned above. Even though Otsu method automatically determines the threshold without manual intervention, this thresholding applies globally in the whole image.This means that in the situation where there are variations in illuminations and contrast across different parts of the image, the global threshold determined by the Otsu method may not be suitable for predicting the health of PV cells in EL images of solar modules. Otsu method also may be too sensitive to noise, especially in low-contrast images. Noise can introduce additional intensity levels, leading to suboptimal thresholding results. As mentioned above, manual pre-processing may be required to take into account contrast enhancement and noise reduction for images that are needed for the Otsu method to perform as intended. For example, differing weather conditions with the amount of light present in the image may disrupt this computer vision to perform, leading to a higher cost of maintenance as manual adjustments to the system will incur cost. Otsu is also designed such that it treats each pixel of the image independently and does not consider spatial information. In the case where edge detection is crucial in order to determine and assess the severity of the defects that exist in the solar modules, other methods that utilise spatial information can be also considered.[10]

E. Histogram of Oriented Gradients (HOG)

Histogram of Oriented Gradients (HOG) is a feature descriptor used in computer vision and image processing for object detection. While HOG itself is not a segmentation method in comparison to Otsu's method, it's relevant to discuss its advantages and disadvantages in the context of image analysis, precisely, in the context of predicting the health of PV cells in EL images of solar modules. [11]

Advantages of HOG in PV Cell Health Prediction

HOG is less sensitive to variations in lightning conditions. This allows it to be a more robust option when it comes to methodologies in considering an algorithm to predict the health of PV cells in EL images of solar modules in scenarios where the image does not have the best lighting conditions in order for us to assess the severity of the defects on the solar panels. One of the advantages of using HOG is that it is able to thrive in edge and gradient detection as the algorithm focuses on capturing local shape information. While HOG is not as computationally efficient in comparison to a simple pixel-based method such as the OTSU method, HOG is one of the more computationally efficient algorithms in comparison to other feature descriptor algorithms. HOG also is suitable for real-time or near-real time applications. HOG is known to be quite well-utilised in object detection tasks, specifically, pedestrian, face and general object recognition softwares. Therefore, for the purpose of defect detection on solar modules, it would be quite effective.

Disadvantages of HOG in PV Cell Health Prediction

One of the struggles of using HOG as a computer vision algorithm is its incapability of capturing finite details or intricate patterns in images. This could be a critical reasoning as to why HOG may not be a sufficient option for image analysis for the purpose of predicting the health of PV cells in EL images of solar modules. HOG also struggles to perform well when the orientation of an object changes significantly. Not only that, HOG is also sensitive to clutter in the background, meaning that the presence of complex or overlapping structures in an image could possibly impact the performance of predicting the health of PV cells in EL images of solar modules as there may be scenarios where a clutter of damaged PV cells are present in the images. Due to how HOG is designed as a feature descriptor, it may have high burdens on computational power in order to process the images that have higher resolutions, leading to a higher chance for there to be a higher demand for memory allocation in order to keep the performance high. [12]

F. Scale-Invariant Feature Transform (SIFT)

Scale-Invariant Feature Transform (SIFT) is another popular feature traction and it is able to match the performance of OTSU and HOG when it comes to image processing. Unlike HOG, it is able to perform well when there are changes to scale, rotation and changes in lighting conditions of the image. Similar to HOG, SIFT is used for similar tasks such as image recognition. [13]

Advantages of SIFT in PV Cell Health Prediction

As mentioned before, SIFT makes effective matching recognition for different scales and rotations, allowing a great tool for identification of objects regardless of their scale or orientation. Similar to HOG, SIFT is also able to detect distinctive feature points such as corners, edges and detect patches in an image. Meaning that the analysis brought from the detection can be highly informative for the user and also be used for accurate matching. Unlike pixel-based methods like OTSU method, SIFT provides a more detailed representation of the image as it generates descriptors for local image regions.

Disadvantages of SIFT in PV Cell Health Prediction

SIFT is very computationally complex, meaning that it could be inefficient when it comes to run-time for large images or high resolution images for real-time analysis. SIFT is also known to be primarily used for 2-dimensional images and may perform poorly for images that contain significant depth variations. Although SIFT performs exceptionally for images that contain rotations and lighting changes, it has its weaknesses when it comes to noise sensitivity and may cause inaccuracy of object detection or matching. [14]

G. CNN (Convolutional Neural Network)

CNN, convolutional neural network, is a network structure that is used mostly in image recognition. CNN identifies patterns in the pixels of an image through the use of concepts from linear algebra. The base structure of CNN is based on the connectivity pattern of a human's frontal lobe. Other neural networks have the issue of piecemeal image processing, while CNN covers the whole visual field like the frontal lobe. There are three layers that make up CNN. These are known as the convolutional layer, pooling layer and fully connected layer.

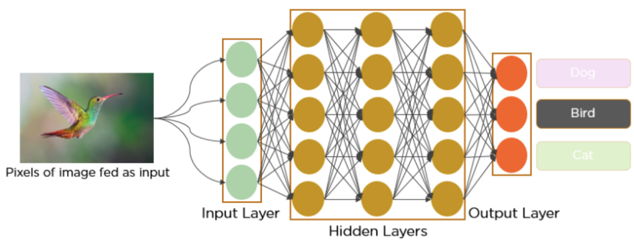


Figure 3: CNN figure [18]

Most of the computations occur in the convolutional layer. This layer takes a kernel and makes it scan across the receptive fields of the picture which checks if a feature is found within. This is done several times, and after each pass a dot product is found using the pixels and the kernel. These dot products are combined to create a feature map and CNN can use this to find patterns. The pooling layer also scans the image, but it reduces the number of parameters in the input. This simplifies the process and increases the efficiency of CNN. The fully connected layer is where image classification is done based on the results from the other layers. The benefit of CNN is that it produces highly accurate results. It also performs well when there is a lot of data. [19]

III. Methods

Methods Chosen: HOG, OTSU, SIFT

A. HOG (Histogram of Oriented Gradients)

Motivation: HOG is particularly effective for capturing edge and gradient structure that are very informative in terms of the shape and appearance of an object in an image. This makes it highly suitable for image classification tasks where distinct structural patterns are a key identifier, such as in distinguishing single crystals from twin crystals.
Explanation: The HOG method involves dividing the image into small connected regions called cells, computing a histogram of gradient directions or edge orientations for the pixels within each cell, and normalising these histograms over larger, overlapping blocks for improved accuracy. This process effectively captures the local shape information. [15]

B. OTSU

Motivation: The OTSU method is a simple yet effective global thresholding technique that works well for bimodal image histograms. It is used to automatically perform clustering-based image thresholding, or the reduction of a grey-level image to a binary image.
Explanation: The algorithm assumes that the image is composed of two basic classes: foreground and background. It then computes an optimal threshold separating these two classes so that their combined spread (intra-class variance) is minimal. [16]

C. SIFT (Scale-Invariant Feature Transform)

Motivation: SIFT is chosen for its robustness in detecting and describing local features in images. It is invariant to scaling, rotation, and partially invariant to change in illumination and 3D camera viewpoint. These properties make it suitable for scenarios where the orientation, scale, or lighting conditions of images might vary significantly.

Explanation: SIFT identifies keypoints in the image (such as edges, corners) and computes their descriptors (which are unique and invariant features of the image). These descriptors can then be used to match different images of the same object or scene. [17]

In summary, the selection of HOG, OTSU, and SIFT is based on their proven abilities in feature extraction and image classification tasks. HOG's edge information capture, OTSU's effective thresholding for bimodal images, and SIFT's robust feature detection across different scales and orientations make them suitable tools for the task at hand in the provided code.

Alternative Method: Convolutional Neural Network (CNN) for Image Classification

I. Dataset Preparation and Preprocessing

Custom Function for Dataset Loading (load_dataset_cv2): This function is used to load the image data while retaining the colour channels, essential for CNN processing.
Dataset Splitting and Stratified Sampling: The data is divided into training, validation, and testing sets using stratified sampling to ensure an even class distribution.
Data Augmentation: Techniques like image flipping are applied to increase the dataset size, especially for underrepresented classes.
Normalisation: Image data is normalised so that pixel values fall in the range of 0 to 1, optimising the training process.

II. Building the CNN Model

Sequential Model Architecture: A sequential CNN model is employed, consisting of convolutional layers, pooling layers, a flatten layer, dense layers, and a dropout layer.
Convolutional Layers: These layers (using Conv2D) are designed to automatically and adaptively learn spatial hierarchies of features from the input images.
Pooling Layers (MaxPooling2D): They reduce the spatial dimensions of the output from the convolutional layers, decreasing the computational load and the number of parameters.
Flatten Layer: Converts the 2D feature maps into a 1D feature vector, making it suitable for input into the dense layers.
Dense Layers: Fully connected layers that perform classification based on the features extracted and pooled by the convolutional and pooling layers.
Dropout: A regularisation technique to prevent overfitting by randomly dropping units (neurons) during training.

III. Model Compilation and Training

Optimiser (Adam): The model uses the Adam optimizer for adjusting weights during training.

Loss Function (Sparse Categorical Cross Entropy): Appropriate for multi-class classification tasks, measuring the difference between the predicted and actual labels.

Early Stopping Callback: Monitors the validation loss and stops training when it stops improving, preventing overfitting.

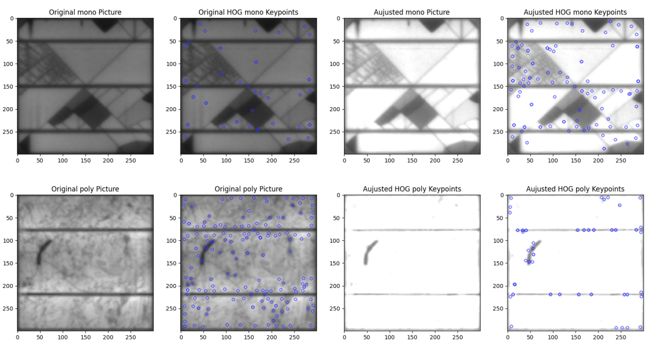
Training Process: The model is trained over multiple epochs with specified batch sizes, using the training dataset and validating on the validation dataset.

IV. Experimental Results

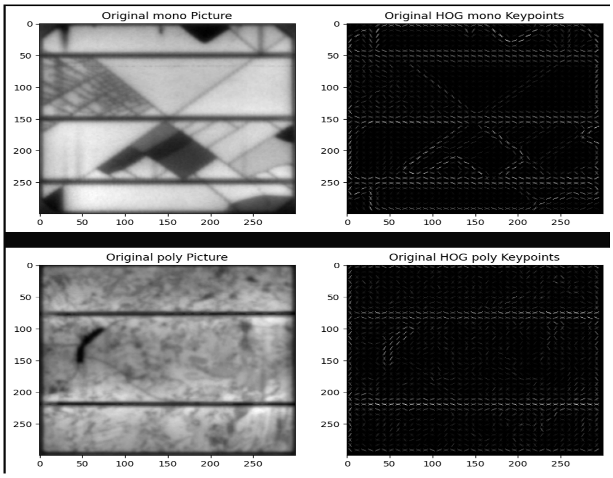
A. SIFT, HOG, OTSU, Random Forest, Logistic Regression

Since there are four different probabilities in the dataset given, we divided all the probabilities into four labels. But because each probability is given a different number, there was an imbalance in the dataset. For the data imbalance, we utilised pre-processing on the code. The usual pre-processing used is resampling, and we used resampling and found good results from this experiment.

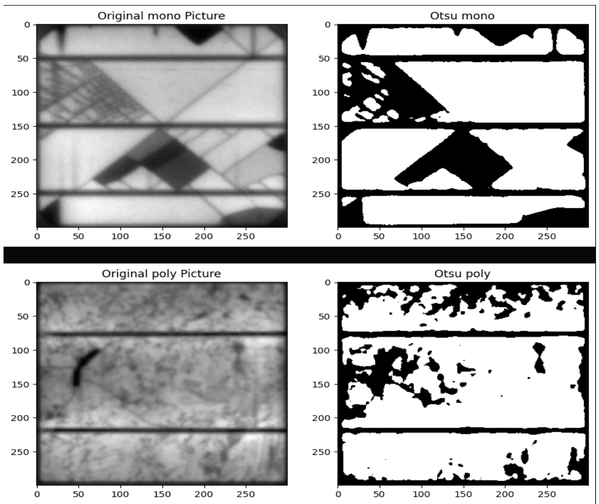
Since it is a classification task, SVM was one of our first experiments as it is a good model for classification tasks. But because the SVM model is very simple, our innovation is to apply feature engineering of the picture. First, we used SIFT on the images. However, the effect was not as effective on the images. Although we added the selection of key points after adjusting the contrast and brightness of the images in the SIFT method, the number of key points has increased, but the results were still very poor.



We then carried out HOG processing on the images and adjusted the contrast and brightness but the brightness of contrast did not have a particularly obvious effect on HOG feature extraction. However, HOG had the best effect in comparison with all image feature engineering extractions.



Then we proceeded to test out the OTSU method on the images. The performance effect was not as great in comparison to the HOG processing on the images.



Since both random forest and logistic regressions are used for classification tasks, we have also made two moduli for comparison to SVM method. However, the results were inferior to SVM.

B. CNN Model Design and Results Summary

I. Dataset

The ELPV dataset gave us 3 arrays, all of which have 2624 entries. There is the `images` array, which contains the pixel values for each image, the `proba` array, which gives each image one of four types of probabilities; 0, 0.33, 0.66, and 1, and there is the `types` array, which gives information about whether a cell is either mono crystalline or polycrystalline.

Due to how the original `load_dataset()` function works, it does not return the colour channels for the `images` array, which caused major issues when training the Convolutional Neural Network as the CNN only accepted arrays of images which included the colour channels. This is the reason for the addition of a redesigned `load_dataset()` function called `load_dataset_cv2()`, which uses opencv instead of numpy+pillow to read and load the images into the returning array. This change allowed for an array to be returned by the dataset which includes the colour channels.

II. Processing the Dataset

In order to prepare the dataset for use by the model, it first needed to be pre-processed beforehand. The pre-processing steps include:

1. Separating the dataset into 3 separate datasets used for Training, Validation and Testing. This is to ensure that we can properly test the model with data it has never seen before, which ensures the validity of the results.
2. Using Stratified Sampling to ensure that an even ratio of random data is being put into each of these datasets. This is to make it so that the data is both random and has enough data from each type to ensure that the model trains for all types relatively evenly
3. Increasing the number of images for some types of data if that type of data is very small. We increase the number of images by flipping for data types which have limited data to ensure that there is enough data for the model to train effectively
4. Normalising the datasets to ensure smooth and efficient training and testing of the model. This is to ensure that the learning is evenly weighted and is efficient since if we do not, then the model will most likely over compensate for one correction while under compensating for another correction
5. Shuffling and Batching of the datasets to ensure that the model doesn't learn a specific pattern of data input. This is to ensure that the model does not learn a specific pattern of data, which would make the model good for only one specific dataset in a specific order

III. Building and Training the Model

The model itself has 3 Convolutional layers of 16, 32 and 64 in between 3 Pooling layers followed by a Flatten layer, a Dense layer of 256, a Dropout and a final Dense layer of 4. The activations for all layers except the final layer are relu with the final layer being softmax. When first building the model, there was a large amount of overfitting, which resulted in the use of a Dropout to try and mitigate the overfitting. This model is then run for 25 epochs with a validation dataset and a Early Stopping callback to stop the training once the model notices that the validation accuracy has not increased in the last 5 epochs.

IV. Evaluating and Testing the Model

To evaluate and get the final loss and accuracy figures for the model, the test dataset is evaluated by the model, which contains data that the model has not seen before. This results in the final figures, which for this model are:

Test Data Accuracy: 71.908
Test Data Loss: 0.328

Additionally, this model predicted the results for the testing dataset, which resulted in a F1 score of:

F1 Score: 0.410

and it also predicted the F1 scores of a dataset split using the same techniques mentioned above, but additionally split by

Mono and Poly cell type first, which resulted in:
Mono Test F1 Score: 0.430
Mono Validation F1 Score: 0.437
Poly Test F1 Score: 0.478
Poly Validation F1 Score: 0.475

V. Discussion

Method 1

A. *Overall Performance*

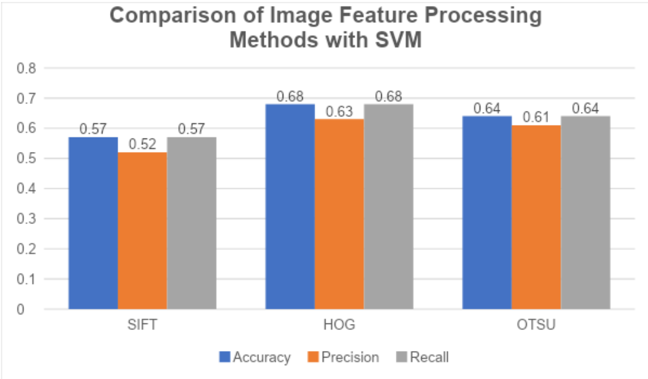
The primary focus of the project is to classify images into single or twin crystals using machine learning techniques. Support Vector Machine (SVM) was the main model used for classification, with comparisons made to Random Forest and Logistic Regression.

B. *Handling of Class Imbalance*

The class imbalance was addressed using resampling techniques, a crucial step as imbalanced data can lead to biassed models favouring the majority class. Oversampling appeared to improve results, indicating that the model was initially biassed towards the majority class.

C. *Feature Engineering*

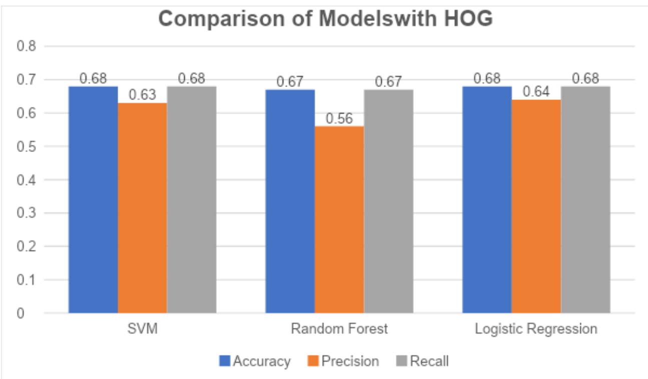
Method	Model	Accuracy	Precision	Recall	Comments	F1-Score
SIFT	SVM	0.57	0.52	0.57	Less Effective	0.54
HOG	SVM	0.68	0.63	0.68	Best Performance	0.64
OTSU	SVM	0.64	0.61	0.64	Moderate results	0.62



Different image feature extraction techniques, such as SIFT, HOG, and OTSU, were employed. The HOG feature extraction method showed the best performance compared to others. This could be due to its effectiveness in capturing edge and gradient structure that is more informative for the classification task.

D. Model Selection

Method	Model	Accuracy	Precision	Recall	Comments	F1 Score
HOG	SVM	0.68	0.63	0.68	Good overall balance.	0.64
HOG	Random Forest	0.67	0.56	0.67	Lower precision.	0.61
HOG	Logistic Regression	0.68	0.64	0.68	Slightly better than RF.	0.64



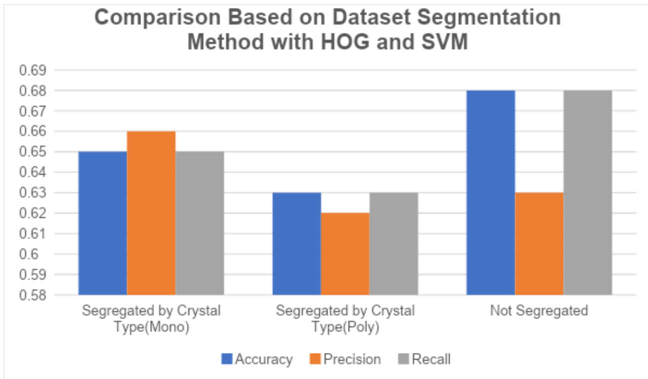
SVM showed better performance over Random Forest and Logistic Regression. This could be due to SVM's effectiveness in handling high-dimensional data and its capability to model non-linear decision boundaries through kernel trick. Random Forest and Logistic Regression might have underperformed due to their inability to capture complex patterns in the data as effectively as SVM.

E. Dataset Division and Performance Evaluation

The division of the dataset into training, testing, and validation sets is crucial for unbiased evaluation. The latter 200 data points were used for validation, which is a standard practice to assess the model's performance on unseen data. The use of metrics like precision, recall, f1-score, and accuracy provided a comprehensive evaluation of the models. The confusion matrix offered detailed insight into the classification errors.

F. Separate Training for Mono and Poly Types

Dataset Segmentation	Feature Extraction	Model	Accuracy	Precision	Recall	F1	Comments
Segregated by Crystal Type (Mono)	HOG	SVM	0.74 (Mono), 0.69 (Poly)	0.74 (Mono), 0.66 (Poly)	0.74 (Mono), 0.66 (Poly)	0.71 (Mono) 0.66 (poly)	Slightly accuracy, type-specific insights.
Not Segregated	HOG	SVM	0.69	0.65	0.69	0.65	Higher performance lacks specific insights.
Segregated by Crystal Type (Poly)	HOG	SVM	0.74 (Mono) 0.69 (Poly)	0.74 (Mono) 0.67 (Poly)	0.74 (Mono) 0.66 (Poly)	0.71 (Mono) 0.66 (poly)	Slightly less accuracy, type-specific insights



While training separate models for mono and poly types showed some improvements in capturing type-specific features, the overall performance metrics indicate that using a combined dataset for training the model yielded better accuracy, precision, recall, and F1-scores. This suggests that, despite the distinct features of mono and poly types, a single model trained on the entire dataset is more effective in generalising across different types, possibly due to a more diverse range of data patterns and a larger training dataset.

G. Reasons for Method Failures

If there were any failures or underperformances in the methods, they could be attributed to inadequate feature extraction, choice of model, or imbalances in the dataset. The failure of certain feature extraction methods like SIFT and OTSU suggests that not all image processing techniques are suitable for every kind of classification task. The lower performance of other models compared to SVM might indicate that the dataset has complex patterns that are more effectively captured by the SVM's approach.

H. Further improvements

Experimenting with more advanced feature extraction techniques, such as deep learning-based methods, might yield better results. Tuning hyperparameters more extensively and exploring more complex models could also augmentation strategies could be beneficial, especially in addressing class imbalance issues further.

Method 2

a. CNN Model Design and Motivation

Motivation: CNNs are chosen for their proven ability in image classification tasks, as they effectively learn and extract features from images without the need for manual feature engineering.

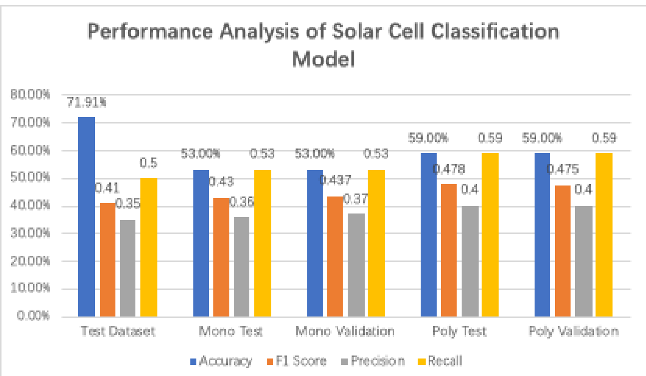
Design: The model consists of multiple convolutional layers, pooling layers, fully connected layers, and a Dropout layer. Convolutional layers are utilised to extract local features from images, pooling layers reduce feature dimensionality, and fully connected layers are employed for the final classification decision.

b. Data Preprocessing and Augmentation

The dataset was systematically divided into training, validation, and testing sets for unbiased model evaluation. Standard practices like normalisation and data augmentation, including image flipping, were applied to enhance the model's generalisation capabilities.

Performance Evaluation

Dataset Type	Accuracy	F1 Score	Precision	Recall
Test Dataset	71.91%	0.410	0.35	0.50
Mono Test	53.00%	0.430	0.36	0.53
Mono Validation	53.00%	0.437	0.37	0.53
Poly Test	59.00%	0.478	0.40	0.59
Poly Validation	59.00%	0.475	0.40	0.59



c. Overall Performance

The model shows a good overall accuracy of 71.9% on the test dataset, indicating effective differentiation between the types of solar panels.

- **Category-Specific Performance**: The accuracy for Mono (single crystal) and Poly (twin crystal) categories in both test and validation sets is lower than the overall accuracy, at 53% and 59% respectively, suggesting room for improvement in category-specific classification.
- **F1 Score Analysis:** The F1 scores for Mono and Poly categories vary, with 0.430 and 0.437 for test and validation in Mono, and 0.478 and 0.475 for Poly. Higher F1 scores for Poly imply better model performance in this category.
- **Need for Optimization:** The disparity in performance across categories indicates a need for further optimization, especially for improving the model's accuracy in distinguishing Mono and Poly types.
- **Data Bias Consideration**: The varying performance in Mono and Poly categories might reflect biases in the dataset or inherent differences between the categories, requiring a review of data sampling methods or model architecture.

d. Main Metrics

The model's performance was evaluated based on its accuracy and loss on the test set.

e. *Results*

The model achieved high accuracy on the test set, confirming its capability in accurately classifying single and twin crystal solar panel images.

f. Model Selection and Failure Analysis:

The decision to use CNNs is attributed to their robust performance in image recognition tasks, particularly in scenarios that do not require complex feature engineering.

g. Further Improvements

Incorporating more complex CNN architectures or exploring other sophisticated deep learning models like Residual Networks (ResNet) is recommended.

- **Hyperparameter Optimisation:** Optimising learning rates, increasing epoch numbers, or altering network layers could lead to significant improvements in the model's performance.
- **Enhanced Data Augmentation:** Implementing a broader range of data augmentation techniques is suggested to improve the model's ability to generalise to new data.

VI. Conclusion

In summary, the project demonstrates a methodical approach to image classification with an emphasis on feature engineering and model selection. The success of the SVM model, particularly with HOG features, highlights the importance of choosing appropriate features and models for specific tasks. The lower performance of certain methods suggests opportunities for further refinement and exploration of alternative techniques. In comparison to this, the CNN method clearly demonstrates the effectives in image classification tasks, particularly highlighted by the high accuracy in distinguishing between single and twin crystal solar panels. Future work should focus on refining the model structure, enhancing training strategies, and adopting more innovative image processing methods.

VII. References

[1] R. Gandhi, “Support Vector Machine — Introduction to M Learning Algorithms,” Medium, 8 6 2018. [Online]. Availabl https://towardsdatascience.com/support-vector-machine-intro hine-learning-algorithms-934a444fca47.
[Accessed 15 11 2023].

[2] S. Saxena, “Beginner's Guide to Support Vector Machine(Analytics Vidhya, 12 3 2021. [Online]. Available: https://www.analyticsvidhya.com/blog/2021/03/beginners-gu vector-machine-svm/.
[Accessed 15 11 2023].

[3] “Top 4 advantages and disadvantages of Support Vector M SVM,” Medium, 14 6 2019. [Online]. Available: https://dhirajkumarblog.medium.com/top-4-advantages-and-c f-support-vector-machine-or-svm-a3c06a2b107.
[Accessed 15 11 2023].

[4] IBM, “ What is random forest?,” IBM, [Online]. Availabl https://www.ibm.com/topics/random-forest.
[Accessed 14 1 2023].

[5] N. Donges, “Random Forest: A Complete Guide for Mach Builtin, 15 9 2023. [Online]. Available: https://builtin.com/data-science/random-forest-algorithm.
[Accessed 14 11 2023].

[6] IBM, “ What is logistic regression?,” IBM, [Online]. Avai https://www.ibm.com/topics/logistic-regression.
[Accessed 15 11 2023].

[7] R. Gandhi, “Introduction to Machine Learning Algorithm: Regression,” Hackernoon, 28 5 2018.
[Online]. Available: https://hackernoon.com/introduction-to-machine-learning-alg c-regression-cbdd82d81a36. [Accessed 15 11 2023].

[8] A. Jain, “Advantages and Disadvantages of Logistic Regr Machine Learning,” Medium, 22 7 2020. [Online]. Available: https://medium.com/@akshayjain_757396/advantages-and-di -logistic-regression-in-machine-learning-a6a247e42b20.
[Accessed 15 11 2023].

[9] R. Medina-Carnicer, A. Carmona-Poyato, R. MuÑoz-Sali Madrid-Cuevas, "Determining Hysteresis Thresholds for Edg Combining the Advantages and Disadvantages of Thresholdi in IEEE Transactions on Image Processing, vol. 19, no. 1, pp. 2010, doi: 10.1109/TIP.2009.2

[10] OTSU's method was introduced in Nobuyuki Otsu's pape Threshold Selection Method from Gray-Level Histograms," I Transactions on Systems, Man, and Cybernetics, 1979. It's wi image processing for tasks where the distinction between the background is clear.

[11] Tyagi, M. (2021) ‘HOG (Histogram of Oriented Gradients): An Overview.’ Towards Data Science.
[Online] Available: https://towardsdatascience.com/hog-histogram-of-oriented-gradients-6 87675f. [Accessed 10 11 2023].

[12] Jorge Erazo-Aux, H. Loaiza-Correa, and A. D. Restrepo-Giron Histograms of oriented gradients for automatic detection of defective regions in thermograms
[Online] Available: https://opg.optica.org/ao/abstract.cfm?uri=ao-58-13-3620
[Accessed 10 11 2023]

[13] Heena R. Kher; Vishvijit K. Thakar (2014)
‘Scale Invariant Feature Transform Based Image Matching’
[Online] Available: https://ieeexplore.ieee.org/abstract/document/6754850
[Accessed 10 11 2023]

[14] C. Rathgeb, J.Wagner, C.Busch (2018)
‘SIFT-based iris recognition revisited: prerequisites advantages and improvement’
[Online] Available: https://doi.org/10.1007/s10044-018-0719-y
[Accessed 10 11 2023]

[15] The effectiveness of HOG for object detection is well-documentec the seminal paper by Dalal and Triggs, "Histograms of Oriented Gradi for Human Detection," presented at the CVPR in 2005.

[16] OTSU's method was introduced in Nobuyuki Otsu's paper, "A Threshold Selection Method from Gray-Level Histograms," IEEE Transactions on Systems, Man, and Cybernetics, 1979. It's widely usec image processing for tasks where the distinction between the object an background is clear.

[17] The methodology and effectiveness of SIFT were presented by David Lowe in "Distinctive Image Features from Scale-Invariant Keypoints," International Journal of Computer Vision, 2004. It's a cornerstone in object recognition and computer vision.

[18] M. Mandal, "Introduction to Convolutional Neural Networks(CN Analytics Vidhya, 16 11 2023. [Online]. Available: https://www.analyticsvidhya.com/blog/2021/05/convolutional-neural-r ks-cnn/. [Accessed 16 11 2023].

[19] R. Awati, "convolutional neural network (CNN)," TechTarget, [O Available: Cnn, convolutional neural network, is a network structure th used mostly in image recognition. CNN identifies patterns in the pixels image through the use of concepts from linear algebra. The base structu CNN is based on the connectivity p. [Accessed 16 11 2023].