9517 Project

Linxun Lyu *z5430933*
Leyan Gao *z5458808*
Lianqiang Zhao *z5396332*
Ruixi Liu *z5381549*
Shiqi Yin *z5370300*

I. Introduction

In the last decade, in response to the challenges of fossil energy depletion and environmental degradation, renewable energy technologies have garnered increasing interest. Among these, photovoltaic solar energy has experienced rapid development and has emerged as a mainstream technology in the renewable energy market, boasting significant competitiveness.

A. *The main faults types of PV modules*

Photovoltaic (PV) modules are a crucial component of solar photovoltaic technology, typically boasting a lifespan of over 25 years [8]. However, faults may occur due to the influence of real-world operating conditions, leading to the risks of performance and durability degradation.

The failure types of photovoltaic modules mainly include the following:

- Optical degradation
- Electrical mismatches and degradation
- PID fault

Optical degradation (Figure 1) in photovoltaic modules is primarily caused by external factors such as high temperatures, humidity, and internal factors such as poor encapsulation quality, incorrect lamination. [9] This degradation type typically manifests as discoloration, bubbles, delamination, or glass breakage in the modules. These faults reduce penetrating solar flux, which decreases the module's output current and impacts the overall power output. These faults can be detected through simple optical or visual inspections.

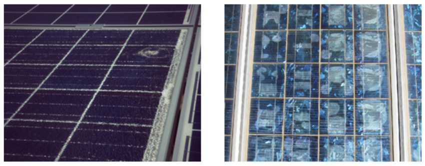


Figure 1: Image of broken module

Electrical mismatches and degradation (Figure 3) encompass various faults such as cell cracks, snail trails, breakages in internal connectors and busbars, shunts, and soldering defects. Cell cracking refers to minute fissures within photovoltaic modules that are often not visible during initial visual inspections and do not immediately result in detectable power loss. However, these cracks may widen and propagate over time, leading to more extensive damage that impacts the module's output. Snail trails are typically associated with the corrosion or degradation of the module's silver paste lines. They frequently occur along the edges of the module and particularly along the paths of microcracks, thus snail trails may indicate the presence of cell cracks within the module.

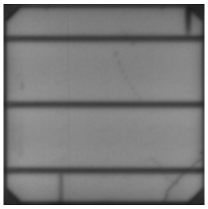


Figure 2: EL image of microcrack and snail trail in PV module

Disconnected cells (Figure 4) are the phenomenon where individual solar cells within a photovoltaic (PV) module become electrically detached from the rest of the array. And broken or degraded string interconnect ribbons refer to the phenomenon where the thin metal strips that form a string of multiple solar cells become fractured or deteriorated. These faults may be caused by physical stress during transportation or installation, thermal cycling, poor soldering, or hotspots arising from the prolonged operation of the PV system. Even a small number of broken interconnection ribbons can result in power losses of up to 35%. As the number of breaks increases, the current may cease to flow through the affected path, routing through adjacent pathways, which can lead to significant local heating and even pose a fire risk.

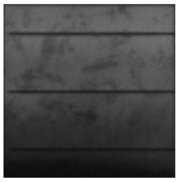


Figure 3: EL image of degradation of cell interconnection in PV module

PID fault, known as potential-induced degradation, is a phenomenon that affects the performance and lifespan of solar cells. It typically occurs in solar modules operating at high voltages, especially those that sustain long-term potential differences with respect to the ground. In outdoor conditions, the power reduction caused by PID can be significant, potentially reaching up to 100% within a few months. IR image (Figure 5) can reveal the lower cells in a module operating at higher temperatures than the upper and middle sections. Similarly, the Electroluminescence (EL) image (Figure 6) can detect the affected areas, which appear darker due to the obstruction of charge transfer.

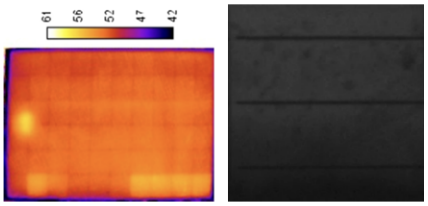


Figure 4: IR
EL image of PID in PV module

B. PV module inspection methods

General methods of outdoor fault diagnosis for solar modules include Infrared Thermography (IRT). This technique initially detects the infrared radiation emitted from the object's surface, then depicts the module in terms of temperature distribution using a thermal imager. Under normal operating conditions, the surface temperature of the module is evenly distributed, and most faults cause anomalies in the thermal distribution on the module's surface, thereby being detected. However, some microcracks may not be detectable as they might not produce abnormal thermal patterns in their initial stages.

Electroluminescence (EL) imaging is another non-destructive technique utilized to detect faults in PV modules. This method gauges the intensity of charge carriers stimulated under forward bias voltage. The resultant imagery displays luminance proportional to the voltage, whereby normally functioning areas are depicted as bright regions, and zones of obstructed electrical flow are rendered as dark regions. EL inspection can precisely locate microcracks and PID faults; however, it is less effective for diagnosing optical degradation faults.

Confronted with the challenges posed by the time-intensive and costly process of Electroluminescence (EL) visual inspection, we have explored the application of machine learning techniques for the automation of defect detection. Support Vector Machine (SVM) and Convolutional Neural Network (CNN) models were employed in our work. The SVM was trained using extracted features from EL images, while the CNN was trained directly with raw image pixels and corresponding labels.

C. Contribution

Leyan Gao: responsible for mainly code of SVM part, using SIFT methods to exract features to train SVM classifier. And Integrating the experiment results of SVM.

Ruixi Liu: I attempted to enhance the generalization capability of a model for image recognition and classification tasks by applying random image augmentation to the training dataset using the ImgAug library. I utilized the VGG16 model for feature extraction. Afterward, I initialized an SVM classifier with a Radial Basis Function (RBF) kernel and the default parameter C=1.0. I then made predictions on the training set, validation set (valid_features), and test set (test_features). Through this process, I was able to achieve an accuracy rate of approximately 76%.

Lianqiang Zhao: For the model part, i implemented CNN using VGG16 and came up with the corresponding experimental results. And i have also brought up some advise for SVM, such as adding up the SIFT feature extraction method to form the comparison. Apart from the model, the corresponding part of powerpoint and presentation also done by me. Lastly, i have been responsible for combining all report component into one.

Shiqi Yin: In the SVM-SIFT and SVM-VGG16 models, I conducted a comprehensive evaluation of their performance using accuracy, F1 score, and confusion matrices. In both the report and the video, I compared the results and analyzed the reasons in Discussion section.

Linxuan Lyu: I took charge of the Resnet in the model part, using the learning rate schedule and early stopping mechanism to enhance the model performance. Also, I reviewed the literature of PV module faults to provide suggestions for analyzing model performance.

D. Outline

The report is organized as follows. The literature review is in Section 2, and Section 3 introduces the methods we used in our work. Section 4 presents the experimental results of our work, and Section 5 evaluates and compares these methods and their results. And finally, we concludes in Section 6.

II. Literature Review

In the research of ”Defect detection of solar cells in electroluminescence images using Fourier image reconstruction”, Tsai (Tsai et al., 2012) [2] came up with a new method of photovoltaic modules defect detection that aiming to find the defective modules based on Fourie image reconstruction. Because the defective area almost always shape as line or bar area with darker colour compared with other area in vicinity, after transforming the original image into

Fourier image, a line shows repaid change of pixel would appear in the middle that represents darker line or bar appear in the original image. And through implementing mask onto the Fourier image we can discard this line. After implementing inverse Fourier Transformation we will get a defect-free image that will be used to subtract the original defective image. If the image after subtraction features with obvious line, we can conclude that this modules is defective. The evaluation shows this methode can effectively detect small cracks and finger interruption and so on even in a uneven background, which means different bunch of area tend to have dark area but not as cracks shape.

Deitsch et al.(2019) [1] conduct a study on comparing difference between traditional machine learning method and deep learning method in terms of running time, accuracy and so on. For traditional machine learning method, they chose to use SVM with six different feature extractors combination to test. Eventually they chose KAZE/VGG as their feature extractor. And for DL, undoubtedly they selected to use CNN with VGG16 as backbone. The result shows SVM has lower accuracy with around 78% accuracy on the overall test set and 83.8% on Monocrystalline and 72.8% on Polycrystalline, but faster training time though,

III. Methods

The overall methods that we have used are SVM and CNN.

A. Convolutional Neural Network

In order to compare with traditional machine learning method in several aspects of model performance, we have utilised CNN with different backbones and corresponding strategies. Given the limited dataset we have got, pre-trained model is absolutely best choice. Here we implemented two pre-trained models to compare their performance.

The first pre-trained model is VGG-19 network architecture (Simonyan and Zisserman, 2015) [3] originally trained on the IMAGENET dataset using 1.28 million images and 1000 classes. Just as its name, VGG-19 originally has 16 convalutional layers with max pooling layer following and 3 fully function layers.

Another pre-trained model is ResNet-18 architecture, which has been similarly trained on IMAGENET dataset. What makes ResNet-18 different to VGG-19 is ResNet-18 implements residual block that is being used to cut off the connection between one layer and another, hence gradient vanishing and exploding mitigated.

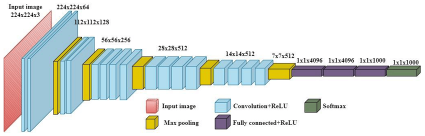


Figure 5: VGG19

1) VGG-19 structure:

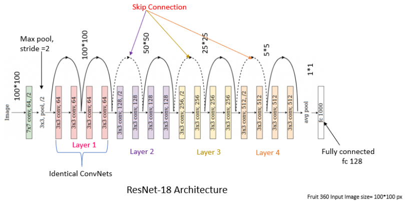


Figure 6: ResNet-18

2) ResNet-18 structure:

B. Support Vector Machine

Many traditional, machine learning, and deep learning-based computer vision methods could be used for this task. The dataset contains 2,624 EL images of functional and defective PV cells with varying degrees of degradation extracted from 44 different solar modules.The support-vector network is a new learning machine for two-group classification problems. The machine conceptually implements the following idea: input vectors are non-linearly mapped to a very high-dimension feature space. In this feature space a linear decision surface is constructed. (Cortes and Vapnik, 1995) [4]. In addition, even in cases where differences are not obvious in EL images, SVM can effectively distinguish between functional and defective PV cells. Therefore, when handling with datasets that are not too large, with high feature dimensions, and require good generalization capabilities, we first consider using the SVM method to complete the task.

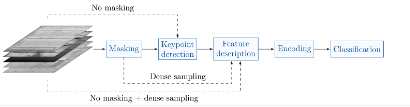
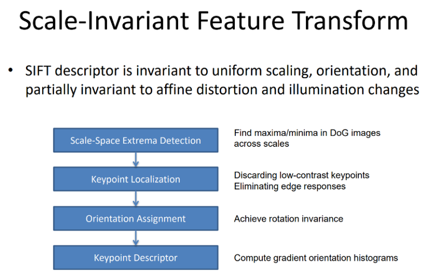


Figure 7: SVM

1) SIFT:

Feature Transform (SIFT) (Lowe, 1999) [6], which detects and describes features at multiple scales. SIFT is invariant to rotation, translation, and scaling, and partially resilient to varying illumination conditions.



Scale-Invariant Feature Transform

- SIFT descriptor is invariant to uniform scaling, orientation, and partially invariant to affine distortion and illumination changes

Scale-Space Extrema Detection
Find maxima/minima in DoG images across scales
Keypoint Localization
Discarding low-contrast keypoints
Eliminating edge responses
Orientation Assignment
Achieve rotation invariance
Keypoint Descriptor
Compute gradient orientation histograms

Figure 8: SIFT

2) Fourier Transform: Fourier transform is mainly being used on signal analysis. What it does is to decompose signal into its constituent frequencies based on the theory that every signal can be reconstructed as a sum of sinusoidal functions, each with a certain amplitude and phase. The Fourier spectrum is noise-insensitive and is ideally suited for representing the directionality of lines in a complicated spatial image.(Tsai et al., 2012) [1]

Downbelow i will introduce the steps of how fourier transformation extract image features:

- Transform the input image to get its constituent frequencies.
- Cover different masks to maintain the highest or lowest frequencies.
- Fed the selected frequency components as feature inputs into SVM,

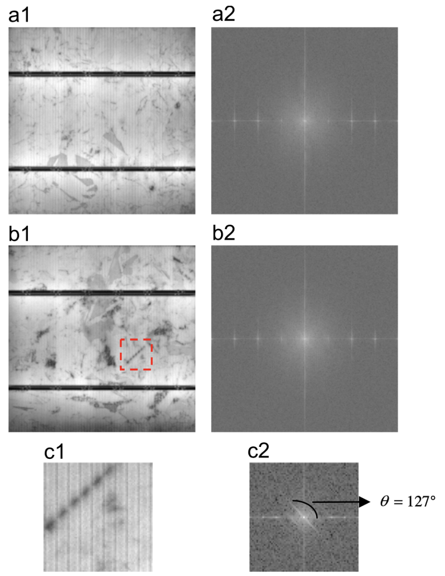


Figure 9: Fourier spectrum images of solar cells: (a1) defect-free EL image; (a2) spectrum image of (a1); (b1) full-sized defective EL image of 550 x 550 pixels; (b2) spectrum image of (b1) where no lines can be visually observed; (c1) small crack in a 75 x 75 EL subimage; (c2) spectrum image of (c1), where a 1271 -line can be distinctly observed

3) Neural Network: VGG16 is widely used in image processing. Its greatest advantage lies in its deep and uniform network structure, make it possible to extract complex features ranging from low to high levels effectively, and let it suitable for various image recognition tasks. VGG16 uses a uniform 3x3 convolution kernel, and after multiple convolution and pooling layers, it further processes features through several fully connected layers.

In the image processing tasks, although the VGG model performs very well, we found that ResNet (Residual Network) is more efficient. Proposed by Kaiming He in 2015, ResNet extracts image features effectively through its innovative residual learning framework. The key to this architecture is the inclusion of skip connections in each residual block to allow data to bypass certain layers, thereby alleviating the problem of vanishing gradients and enabling the network to learn deeper and more complex features. ResNet extracts features through convolution and pooling layers, then deepens the detail and complexity of these features through residual blocks. These skip connections ensure that even in deep networks, the original information can be transmitted completely.

IV. Experimental Results

Based on what project sepicification required, i have used F1 score, overall accuracy and 4x4 confusion matrix to test our models' performance. To clarify before start, models we have implemented include CNN and SVM. For CNN, we have used two feature extraction methods, VGG16 and ResNet34 respectively. And for SVM, we have used SIFT, Fourier Transform and VGG16 as our feature extraction methods. Downbelow i will introduce the experimental setup that we used to get the results and test model performance.

A. Convolution Neural Network

1) VGG16:

- Data Transform: In order to get more robust results, image augmentation or data transformation is important as we can get different shapes or intensity of images. Here is the procedures of image augementation.
  - transforms.RandomRotation(degrees=15)
  - transforms.RandomHorizontalFlip(p=0.5)
  - transforms.RandomVerticalFlip(p=0.5)
  - transforms.ColorJitter(brightness=0.2, contrast=0.2)
  - transforms.RandomResizedCrop(size=(224, 224), scale=(0.8, 1.0))
  - transforms.ToTensor()
  - transforms.Normalize(mean=[0.485, 0.456, 0.406], std=[0.229, 0.224, 0.225])

- Load Pre-trained Model:

The use of pre-trained model can not only speed up the training process, as training a model from scratch requires significant computational resources and time, but also improve our model performance on this dataset as it has already been trained on large datasets with a vast variety of features.

- Customize our pre-trained model

As we have only got 4 classes, wheares pre-trained VGG model aiming to put images into 1000 classes. Therefore customizing the pre-trained model to fit in our task is curcial. Here in order to keep the depth of model but fit in our task, i have only modified the last layer, which is the classifier layer with 1000 outputs, into one classifier layer with 4 outputs and one LogSoftmax layer for computing the loss.

- Test model performance before training

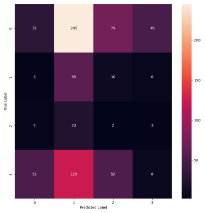


Figure 10: Confusion Matrix on Both Module Before Training

- Test model performance after training

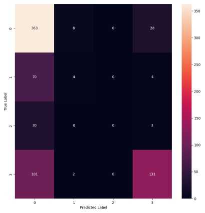


Figure 11: Confusion Matrix on Both Module After Training

- Unbalanced Dataset:

In the overall training dataset, classes {1.0}, {0.66}, {0.33}, {0.0} have 417, 60, 185 and 879 images respectively. In order to conquer class imbalance, i tried to down-weight the categories with large numbers of samples, which means to attach higher weight to classes with fewer images and lower weight to classes with more images. Downbelow is the confusion matrix i have got after considering class imbalance:

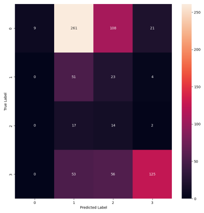


Figure 12: Confusion Matrix of Considering Class Imbalance on Both Module

From above we can observe that the result confusion matrix is not desired, and we can even say model has performed really bad after dealing with class imbalance. To analyse the reason, i reckon that our model originally did not work well on recognizing 0.666 and 0.333 these two classes. After attaching higher weights onto these two classes, model is not learning enough from 1.0 and 0.0 because of the lower weights and is hard to learn from 0.666 and 0.333, hence resulting in higher loss and lower accuracy.

- Test model performance after training on Monocrystalline

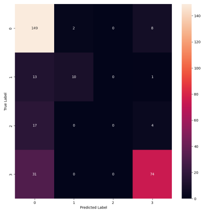


Figure 13: Confusion Matrix on Mono

- Test model performance after training on Polycrystalline

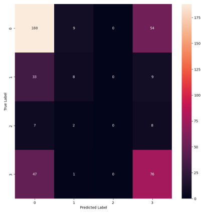


Figure 14: Confusion Matrix on Poly

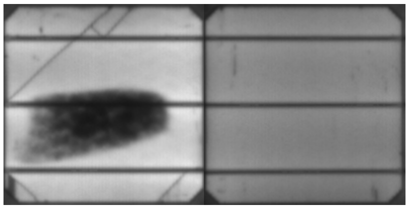
- Number metrics on modules

Table I: Metrics

Cell Modules	**Metrics**			
	Loss	***Accuracy***	***Recall***	***F1 Score***
Overall	0.858	0.670	0.380	0.374
Mono	0.771	0.756	0.515	0.533
Poly	0.914	0.613	0.380	0.376

[a] Keep only the first three digits.

- Representative examples of classifications



(a) Successful Classification (b) Failed Classification

Figure 15: Representative classification

In order to correctly classify images, obvious features are crucial. From the subfigure (a) we can tell 4 or 5 cracks and a large block of black appear obviously on the surface, therefore these features are easily to be recognized by our models. But for the subfigure (b), not only just two vaguely small cracks showing up in the surface, but also their pixel intensity are similar to the background, hence our model might be confused about this image class.

2) ResNet18:

- Data Augmentation: To enhance the generalization capabilities of our model, we have experimented with various data augmentation techniques. These include random image cropping and stochastic horizontal and vertical flipping, complemented by affine transformations.
  - transforms.RandomHorizontalFlip(p=0.5)
  - transforms.RandomVerticalFlip(p=0.5)
  - transforms.RandomAffine(degrees=(-3, 3), translate=(0.02, 0.02))
  - transforms.RandomResizedCrop((300, 300), scale=(0.98, 1.0), ratio=(1.0, 1.0))
- Pre-trained Model

Our study employed pre-trained models from the ResNet architecture, specifically exploring the ResNet18 and ResNet34 variants. Comparative analyses revealed that the ResNet18 model demonstrated better prediction ability. This can be rationalized by considering the architecture's shallower depth in weight layers alongside a reduced number of residual blocks compared to ResNet18. Such a configuration endows ResNet18 with enhanced generalization capabilities when handling smaller datasets, whereas the deeper ResNet18 is more prone to overfitting under similar datasets.

- Customize Pre-trained Model

Similar to VGG, we modified the last layer of the model to accommodate the requirements of a multi-class classification task.

- Learning Rate Schedule:

The learning rate is a critical hyperparameter in model training that determines the magnitude of the step size when updating the model parameters. In our research, we experimented with two different learning rate schedulers for the Resnet model: ReduceLROnPlateau and MultistepLR.

- Validation Set and Early Stopping Mechanism

In our methodology, the training dataset was partitioned such that 75% was utilized for training purposes, while the remaining 25% was allocated as a validation set. This validation subset was employed to assess the loss after each epoch during the training process. An early stopping mechanism was implemented, whereby training would be prematurely terminated if the validation loss did not exhibit significant improvement within a predefined number of epochs. This approach was designed to mitigate performance degradation and alleviate the issue of model overfitting.

  - patience = 11
  - min_improvement = 0.003
- Class Weight System

Besides, to alleviate the severe class imbalance present in the dataset, our methodology incorporated a system of class weights. Precisely, these weights were inversely related to the number of samples per class, resulting in the minimal weight being attributed to class 0 and the maximum to class 2. This strategy was designed to mitigate the model's fixation on the predominant classes to a certain extent, thereby enhancing the model's generalization ability.

- Result:

Table II: Metrics

Cell Modules	**Metrics**		
	Accuracy	***Recall***	***F1 Score***
Overall	75.71%	74.31%	74.57%
Mono	40.08%	52.21%	33.38%
Poly	77.17%	74.31%	75.46%

[a] Keep only the first three digits.

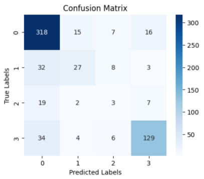


Figure 16: The Confusion Matrix of Resnet Model

From the model outcomes, it is evident that the model predicts highly accurately for the fully functional category (class 0) and the certainly defective category (class 3). However, it performs less effectively for the prediction of the possibly and likely defective categories(classes 1 and 2), which may be attributable to an uneven distribution of data samples.

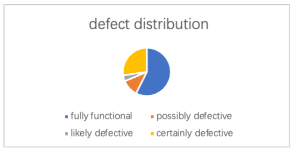
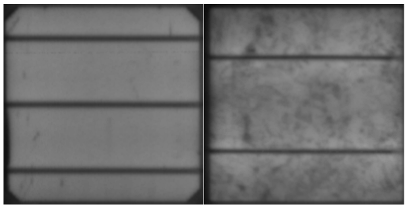


Figure 17: Defect Distribution

The dataset's defect distribution (Figure 17) reveals a lower incidence of the 'possibly defective' and 'likely defective' categories, indicating an insufficiency in the model's training for these two classifications.

It was distinctly observed that the model's predictive accuracy for polycrystalline components surpassed that of monocrystalline components. One potential explanation for this discrepancy could be the uneven class distribution within the dataset, particularly noting that the proportion of fault-free instances in monocrystalline components is lower compared to that within polycrystalline components. Despite the implementation of class weights to address this imbalance, the model still exhibited a propensity to favor the prediction of the categories with a relatively high proportion, which may result in enhanced performance on polycrystalline classifications. Additionally, the intrinsic differences between monocrystalline and polycrystalline components in terms of material structure and performance may be reflected in their EL images. As Figure 3 shown, monocrystalline silicon typically exhibits a more uniform structure, whereas polycrystalline silicon presents a variety of grain structures and boundaries. This could lead to disparities in the model's ability to extract pivotal features from monocrystalline and polycrystalline cells, thereby affecting its learning efficacy and classification performance.



(a) Typical Mono Modules

(b) Typical Poly Modules

- Representative Examples of Classifications and Analysis

From the analysis of misclassified images, it can be found that the model may not have adequately learned the features of microcracks present in categories 1 and 2 (Figure 4). Furthermore, some images, while showing visible traces to the naked eye that are not indicative of actual defects, these features were also not effectively learned by the model.

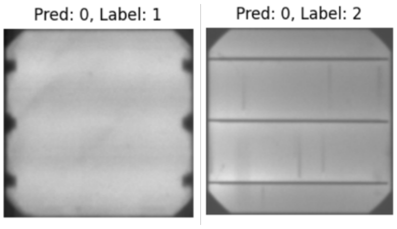


Figure 19: Microcrack that not be detected in Class 1 and 2

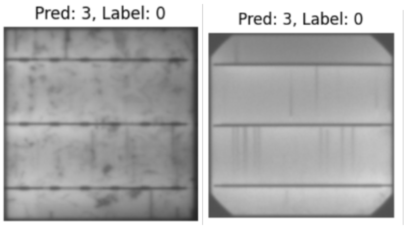


Figure 20: Purely esthetic traces that be predicted as faults

Additionally, it is noteworthy that upon examining the erroneous output images, we discovered that some overall darker images, despite lacking overt cracks on the surface, may harbor significant defects (Figure 6). Such phenomena may be attributed to Potential Induced Degradation (PID), where impeded charge transport results in darker EL images. The model tends to predict a lower likelihood of defects for these types of images, suggesting it may not have effectively learned to recognize this feature.

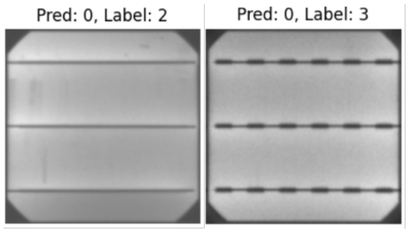


Figure 21: Darker images that be recognized as less likely detective

B. Support Vector Machine

1) SIFT:

- Extracting SIFT Features and Descriptors

Initially, for each image in the training set, we extract SIFT features and descriptors. Because the number of SIFT descriptors from the first step is likely to be different across images, and since SVM requires a fixed-length feature vector whereas SIFT provides local features, it necessitates additional preprocessing to construct feature vectors of uniform length. This is achieved by employing a bag of visual words model, which converts textual data into fixed vector representations. More specifically, we utilize a clustering algorithm, typically k-means, to regroup all SIFT features into clusters (we choose k=120). Each cluster represents one 'word' in the visual vocabulary, and all features are mapped to these words.

- Feature Vectorization and Normalization

After constructing the bag of visual words model, each image's SIFT descriptors are transformed into fixed feature vectors that represent the frequency of each 'word'. In this step, we measure the distance between each image's SIFT descriptors and the central points of every cluster in the k-means model. Each descriptor is then assigned to the nearest cluster centre, and we calculate the frequency of each cluster centre, forming a feature vector. If a cluster centre has no mapped features, its frequency is set to zero. Before deploying the SVM classifier, we normalize the feature vectors to ensure that each dimension contributes equally to the model. This step enhances the model's robustness.

- Training the Classifier

Ultimately, we train the SVM classifier using these feature vectors. It was observed that increasing the C parameter can lead to overfitting. Besides the default SVM with an RBF kernel, we experimented with various types of kernels such as linear, polynomial, and sigmoid. The performance ranking of kernels is RBF, linear, polynomial, and sigmoid, with the RBF kernel performing the best when the C parameter is set to 2.

Table III: Metrics

Cell Modules	**Metrics**	
	Accuracy	***F1 Score***
Overall	72.85%	0.7
Mono	73.46%	0.68
Poly	67.42%	0.65

Downbelow is the confusion matrix of using SIFT as feature extractor:

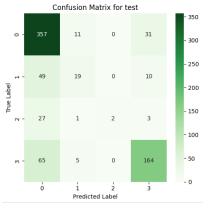


Figure 22: Confusion Matrix on both Module

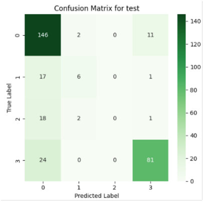


Figure 23: Confusion Matrix on Mono Module

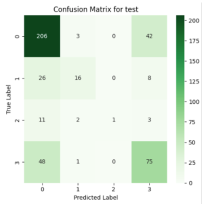


Figure 24: Confusion Matrix on Poly Module

2) VGG16: Aiming to comparing SVM classifier and convolution neural network, we use vgg16 to extract features as well. The step for extracting features is same. And the best svm classifier is rbf kernel with c=2, the result is shown below:

Table IV: Metrics

Cell Modules	**Metrics**	
	Accuracy	***F1 Score***
Overall	68.01%	0.63
Mono	66.99%	0.62
Poly	65.84%	0.6

Downbelow is the confusion matrix of using VGG16 as feature extractor:

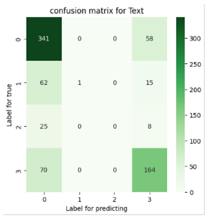


Figure 25: Confusion Matrix on Both Module

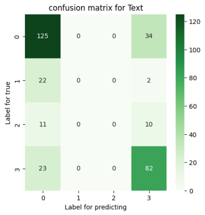


Figure 26: Confusion Matrix on mono Module

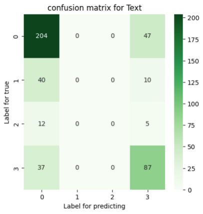


Figure 27: Confusion Matrix on poly Module

V. Discussion

In this section, we will discuss in the performance of different models, focusing on three evaluation metrics which are accuracy, F1 scores, and confusion matrix. These assessment criteria are critical to understanding of the classification performance and weaknesses of the models.

A. Comparison with the creators of the ELPV dataset

1) Accuracy: Benchmark: 0.844

Table V: Accuracy of different models

Models	CNN		SVM	
	VGG16	**ResNet34**	**SIFT**	**VGG16**
Accuracy	0.670	0.757	0.728	0.680

The table shows that all our accuracy score of the models we used are lower that the ELPV Dataset Creators' Methods. That means our methods performed worse than the methods used by the creators of the ELPV dataset.

2) F1 score: Benchmark:0.883

Table VI: Accuracy of different models

Models	CNN		SVM	
	VGG16	**ResNet34**	**SIFT**	**VGG16**
Accuracy	0.374	0.745	0.700	0.630

We can see that the ELPV Dataset Creators' Methods (0.883) has a better general performance of F1 score from the table.

3) Reason: In this part, we will analyze the possible reasons why our models result is worse than the creators of the ELPV dataset. Firstly, we only use part of dataset, and performance of deep learning models is heavily influenced by the size of the training dataset. Because dataset was smaller, less varied, this could have negatively impacted our model's learning process. Moreover, the use of multiple feature extraction in our methods could lead to a situation where irrelevant or redundant features, which can degrade model performance.

B. Comparison between our models

1) CNN:

- Accuracy

Table VII: Model Performance on Different Cell Types

Type	VGG16	ResNet18
Monocrystalline	0.756	0.400
Polycrystalline	0.613	0.771
Overall	0.670	0.757

In this table, we can see that ResNet18 exhibits lower accuracy (0.400) on monocrystalline modules when compare with VGG16, however, ResNet18 show better on polycrystalline and overall modules.

- F1 Score

Table VIII: Model Performance on Different Types of Solar Cells

Model	VGG16	ResNet18
Monocrystalline	0.533	0.333
Polycrystalline	0.376	0.754
Overall	0.374	0.745

According to above data, VGG16 achieves higher F1 score (0.533) on monocrystalline modules than ResNet18, but displays poorer on polycrystalline and overall modules.

- Confusion matrix

After the observation on Figure11-16, it can identify that both CNN models show better performance in identifying class 0 and class3 than other classes. But for ResNet18 on monocrystalline struggles with class 0, having only 9 true positives, and misclassifies several instances as class 2 and 3. On polycrystalline and overall modules, VGG16 show better classification of class0 and class3 than ResNet34, while worse on class1 and class2.

- Reasons

Overall, ResNet18 shows better on polycrystalline and overall modules, but worse on monocrystalline and both categorize class 0 and class3 than other classes. The possible reasons: First of all, VGG16 has a very deep architecture with many layers of convolutional filters of a smaller size, which can be very good at capturing fine-grained details. This could be why VGG16 performs better on monocrystalline modules. Secondly, the representation of class 1 and class 2 defects might be less in the dataset, which could be why both models perform worse on these classes.

2) SVM:

- Accuracy

Table IX: Comparison of SIFT and VGG16 Model Accuracies

Model	SIFT	VGG16
Monocrystalline	0.734	0.669
Polycrystalline	0.674	0.658
Overall	0.728	0.680

It is precise to know that the accuracy of SIFT appear to compare better with the VGG16 on all three areas, which means that SIFT performs a little more efficiently than the other model in this task.

- F1 Score

Table X: Performance Metrics of SIFT and VGG16 Models

Model	SIFT	VGG16
Monocrystalline	0.68	0.62
Polycrystalline	0.65	0.60
Overall	0.70	0.63

Across this evaluated category, SIFT outperforms VGG16 in terms of F1 score. This demonstrates SIFT's superior efficiency in processing and classifying the data for this specific application.

- Confusion matrix

To compare the confusion matrices for SVM with SIFT and SVM with VGG16 feature extractors across, we can analyze Figure22-27, class0 and class3 have significantly higher than class1 and class2 on both SIFT and VGG16 on monocrystalline, polycrystalline, and overall module classification. While SIFT shows better on monocrystalline and overall module, a little worse on polycrystalline than VGG16.

- Reasons

In summary, SVM with SIFT shows a consistent advantage over SVM with VGG16 in classifying solar module defects. The reason is that VGG16 could be overfitting to specific features within the dataset we used, which may not generalize well across all classes, especially for classes 1 and 2.

3) CNN VS SVM:

- Accuracy

With the analysis of two accuracy tables, the best result shows at CNN- ResNet18(0.771) on polycrystalline. It also demonstrates better at CNN- VGG16 on monocrystalline and overall module than two different SVM models.

- F1 Score

According to the tables of F1 score, on polycrystalline and overall module, CNN- ResNet18 appears to be more effective than SVM because of F1 score in excess of 0.7. However, on monocrystalline, SVM-SIFT (0.68) performance better than others.

- Confusion matrix

To identify which confusion matrix is the best, we need to compare Figure22-27 with Figure11-16. For Classification of class0, CNN-VGG16 shows the best performance. For class1 and class2, the results reveals that other models worse than CNN- ResNet18. While SVM-SIFT indicates a good classification ability at class3 with 164 true positives, smaller more than the 131 true positives of CNN-VGG16.

- Reasons

In conclusion, CNN perform better than SVM in our solar module classification. Potential Reason for observed result data: Robustness to Variations of CNNs, which are generally more robust to variations in the input data such as changes in lighting, orientation, or scale. This is essential for solar module classification because defects may appear different depending on the module's angle, lighting conditions during image capture.

C. Future Directions for Research

There is a suggestion exploring the following directions for further improve the levels of solar module classification. Fine-tune existing models tailored to the characteristics of solar modules including the adjustment of parameters and change of data preprocessing methods.

VI. Conclusion

In this project report, we have comprehensively addressed five key components: Introduction, Literature Review, Methods, Experimental Results, and Discussion. In the Introduction, we primarily provided a comprehensive overview of the project's objectives and the dataset utilized. This was succeeded by the Methods section, wherein we primarily delved into the theoretical underpinnings of the two models and the feature extraction methodologies implemented in our study. Substantial effort was dedicated to the development and refinement of experimental setups for evaluating model efficacy. From the obtained data, it was observed that, across all modules, the CNN utilizing ResNet14 architecture exhibited superior performance, achieving an accuracy of 75.71%. This trend persisted in the analysis of polycrystalline cells, with the model attaining an accuracy of 77.17%. Conversely, in the specific context of monocrystalline module identification, the CNN employing the VGG16 architecture demonstrated the most proficient results, recording an accuracy of 75.6%.

Considering the depth of the model, it was evident that the CNN required a considerably deep network architecture to effectively classify images, consequently resulting in an extensive parameter set. In contrast, the SVM classifier, leveraging SIFT for feature extraction, demanded significantly fewer GPU resources, albeit at the expense of reduced accuracy. Thus, in scenarios where there is an abundance of GPU resources, the CNN approach is preferable. However, in resource-constrained environments, the SVM classifier emerges as a practical alternative, offering reasonable accuracy with a substantially lower computational footprint.

References

[1] Tsai D M, Wu S C, Li W C. Defect detection of solar cells in electroluminescence images using Fourier image

reconstruction[J]. Solar Energy Materials and Solar Cells, 2012, 99: 250-262.

[2] Deitsch S, Christlein V, Berger S, et al. Automatic classification of defective photovoltaic module cells in electroluminescence images[J]. Solar Energy, 2019, 185: 455-468.

[3] Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition[J]. arXiv preprint arXiv:1409.1556, 2014.

[4] Cortes C, Vapnik V. Support-vector networks[J]. Machine learning, 1995, 20: 273-297.

[5] Deitsch S, Christlein V, Berger S, et al. Automatic classification of defective photovoltaic module cells in electroluminescence images[J]. Solar Energy, 2019, 185: 455-468.

[6] Lowe, D.G., 1999. Object recognition from local scale-invariant features. In: International Conference on Computer Vision (ICCV), vol. 2. pp. 1150–1157. https://doi.org/10.1109/ICCV.1999.790410. Mair, E., Hager, G.D., Burschka, D

[7] Simonyan, K., Zisserman, A., 2015. Very deep convolutional networks for large-scale image recognition. In: International Conference on Learning Representations. https://doi.org/10.1109/TPAMI.2014.2301163.

[8] Phinikarides A, Kindyni N, Makrides G, Georghiou GE. Review of photovoltaic degradation rate methodologies. Renew Sustain Energy Rev 2014;40:143–52.

[9] Djordjevic S, Parlevliet D, Jennings P. Detectable faults on recently installed solar modules in western Australia. Renew Energy 2014;67:215–21.