Advanced Machine Learning in PV Cell Defect Classification

Abstract — **This report investigates the use of machine learning and deep learning for classifying defects in photovoltaic (PV) cells using electroluminescence (EL) imaging, analysing the ELPV dataset. Focusing on the challenges faced by PV panels due to environmental and manufacturing factors, the study assesses the effectiveness of various algorithms like CNN, Random Forest, ViT, ResNet, and VGG in detecting defects. It highlights the strengths and limitations of these models, especially in dealing with imbalanced datasets and achieving a balance between model simplicity and accuracy. The findings underscore the importance of model optimization and suggest future research directions in model fusion for more accurate defect classification in PV cells.**

I. Introduction

Solar energy systems, especially photovoltaic (PV) panels, are key for generating electricity in remote areas. These panels, made durable with aluminium frames and glass lamination, convert solar energy into electricity. However, they can still be damaged by external elements like debris or thermal stress, and production flaws, which reduces their efficiency..

Traditional methods of visual inspection are labour-intensive and often inadequate, as many defects are not discernible to the naked eye. Moreover, not all visible irregularities correlate with decreased panel efficiency. To overcome these challenges, advanced diagnostic tools are needed to facilitate swift and thorough inspections. Electroluminescence (EL) imaging emerges as a powerful non-invasive technique that enables detailed examination of solar panels. By inducing EL emission via an electrical current, this method allows the capture of high-resolution images that reveal a wide spectrum of cell defects, with malfunctioning areas appearing darker due to their inability to emit light.

Our research aims to explore the effectiveness of machine learning and deep learning technologies in classifying these defects. By applying EL imaging to the ELPV dataset, we have tested various algorithms, including CNN, Random Forest, ViT, ResNet, and VGG, to evaluate their accuracy and efficiency in defect detection. The results show that these technologies improve the precision of defect identification. Furthermore, our study assesses the strengths and weaknesses of these models in processing image data, particularly in terms of accuracy, stability, and their ability to handle imbalanced datasets. Each algorithm demonstrates unique advantages and limitations.

II. Literature Review

This literature review examines pivotal studies in the field of photovoltaic (PV) cell defect detection using machine learning (ML) and deep learning (DL) techniques. Focusing on electroluminescence (EL) imaging, these studies contribute significantly to the accuracy and efficiency of identifying defects in solar cells. They explore various models and approaches, ranging from Deep Neural Networks (DNNs) to Convolutional Neural Networks (CNNs), highlighting their effectiveness in diverse and complex scenarios.

Efficient deep feature extraction and classification in the study focusing on identifying defective photovoltaic module cells in electroluminescence images, the integration of Deep Neural Networks (DNNs) and Machine Learning (ML) techniques plays a pivotal role. The process begins with feature extraction, where significant patterns in the EL images are identified and quantified, highlighting attributes like crack

patterns or discolorations indicative of defects. Following this, feature selection is applied to isolate the most informative features, reducing data dimensionality and enhancing model efficiency. The heart of the study lies in the classification phase, where DNNs, particularly suited for image classification due to their hierarchical feature learning capabilities, are combined with traditional ML algorithms. This combination capitalises on the strengths of both: the deep, nuanced understanding of image features by DNNs, and the diverse, robust decision-making frameworks of ML models. The result is an efficient classification framework that excels in accuracy and computational efficiency, crucial for handling large datasets and enabling real-time defect detection in solar cells. This approach demonstrates how blending advanced neural network architectures with established ML techniques can lead to significant improvements in automated defect classification systems.[1]

Deep learning system for defect classification the research extends the scope of defect detection in photovoltaic (PV) cells by applying various models to the ELPV benchmark images, encompassing both binary (defective vs. non-defective) and more granular four-class classifications. Recognizing the limitations of the ELPV dataset in terms of defect diversity, this study enriches its data pool by incorporating EL images from publicly available datasets. This expanded dataset includes a substantial collection of 18,347 PV cell images, which feature a comprehensive range of 11 different defect types, along with images of non-defective cells. Such an extensive dataset provides a more robust platform for the deep learning models to learn and distinguish between a wider variety of defect types. By training on this diversified dataset, the models are better equipped to accurately classify and identify defects in PV cells, thereby enhancing the reliability and applicability of the classification system in real-world scenarios. This approach demonstrates the importance of diverse and extensive datasets in training deep learning models for accurate and comprehensive defect classification in solar cells.[2]

Large scale challenging dataset In another significant study, researchers utilised a large-scale, challenging dataset comprising 2,624 electroluminescence (EL) images from the ELPV dataset to evaluate the performance of their classification models. This dataset is particularly notable for its complexity and size, providing a comprehensive basis for rigorous testing. The study's approach involved both binary classification, distinguishing between functional and defective solar cells, and multi-class classification, which delves deeper by categorising the cells into functional, mild, moderate, and severe defect classes. The use of such a diverse range of classification tasks demonstrates the study's commitment to nuanced and detailed defect analysis, going beyond mere identification of defects to assess the severity of the damage. This level of detailed classification is vital for practical applications, where understanding the extent of damage can inform maintenance and replacement strategies. By successfully navigating the complexities of this large and varied dataset, the study showcases the potential of advanced classification models in accurately identifying and categorising defects in solar cells, thereby contributing to more efficient and reliable solar energy production.[3]

Deep learning based automated defect classification represents a significant advancement in the field of photovoltaic (PV) cell defect detection. In this research, the publicly available ELPV dataset was utilised, with a particular emphasis on meticulous preprocessing and class categorization to optimise the data for analysis. A key aspect of this study was the implementation of the Convolutional Neural Network-based Image Labelling Detector (CNN-ILD) model. This model was rigorously tested, and the results were highly promising, showcasing classification accuracies ranging from 88.41% to 98.05% for detecting defects in PV cell EL images. Such high accuracy rates are indicative of the model's robustness and reliability in identifying defects. Moreover, the CNN-ILD model distinguished itself by its computational efficiency and stability, outperforming other pre-trained Deep Neural Networks (DNNs). This aspect is particularly crucial in real-world applications where processing speed and consistency are key to timely and reliable defect detection. The success of this model in achieving high accuracy, coupled with its computational advantages, marks a notable contribution to automated defect classification in solar cells, highlighting the potential for deep learning approaches to significantly enhance quality control in solar energy production.[4]

III. METHODS

A. Dataset

The dataset includes 2642 8-bit grayscale images of 300x300 pixels, among which 1074 are Mono and 1550 are Poly. The image format is PNG. These images display solar cells in varying degrees of degradation, both normal and defective.

In the dataset's README, all images have undergone standardisation in size and perspective. Moreover, any distortions caused by the lens have been eliminated. Additionally, the photographs contain only a single solar cell grid, without including other grid blocks. For this reason, the dataset is considered small and clean.

The dataset's annotation files are in CSV format, stored in labels.csv. The metadata in the file includes the following fields: filename, defect probability, and crystal type. The filename format is images/cell0001.png. Defect probabilities are divided into four categories: 0.0, 0.333, 0.666, and 1.0, all as floating-point numbers. The crystal types are categorised into mono and poly, separated by spaces in the columns. The corresponding images are stored in the images directory.

B. Exploratory Data Analysis

The data distribution of this dataset is uneven. Simple statistical processing of the categories shows the number of images per category.

TABLE I: Distribution of Data

Varying degree	0	0.33	0.66	1
Mono	588	117	56	313
Poly	920	178	50	402
Mono&Poly	1508	295	106	715

From the table, the image distribution ratio as shown in Fig. 1.

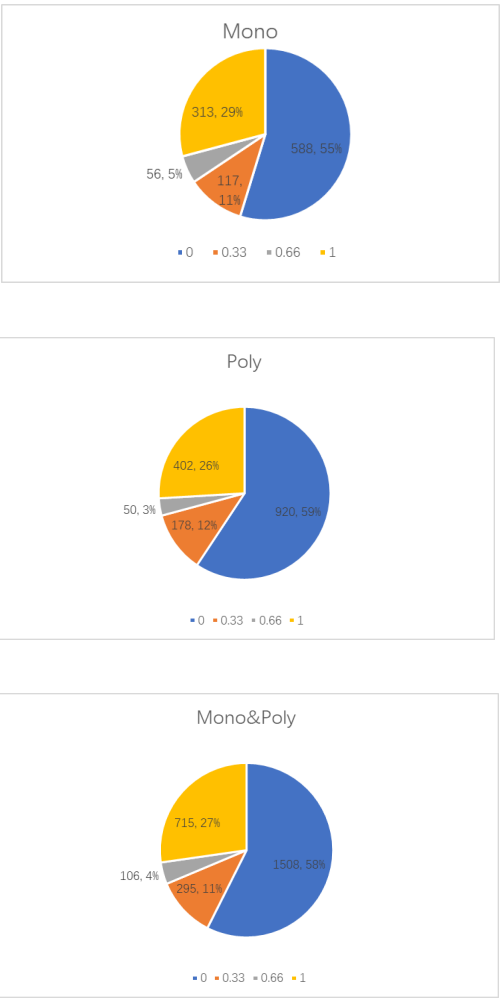

Fig.1: Image Distribution Ratio

Data with a probability of 0.66 accounts for only 4% of the total data size, while data with a probability of 0 represent 58% of the total data volume. Such a data ratio could adversely affect the model's classification training. For some models, they tend to better recognize samples of the majority category. This is because the models are over-optimized for these samples, having seen more of them during the training process. The data with a probability of 0.66 might be underrepresented due to their small quantity, leading to their incorrect recognition during testing.

Similarly, this can also result in inaccurate performance assessments. If the model fails to recognize defects with a probability of 0.66, it

would only affect about 4% of the accuracy, posing higher demands for the model's optimization.

C. Data Augmentation

Addressing the challenge of training deep learning models on small datasets, such as the ELPV dataset, requires innovative solutions to counteract the risk of overfitting. A key strategy used in our study is data augmentation, a process designed to artificially expand the dataset and introduce a greater variety of training examples. In cases of significantly uneven data distribution, selecting appropriate data augmentation methods becomes particularly important. Based on the model's training requirements, we have adopted various data augmentation methods to balance the composition ratio of the data, thus reducing the problem of weak model learning effects due to insufficient training samples. Initially, in the process of dividing the original data, we used two different methods of dataset partitioning. The first method divided the data into eight categories, namely (Mono, Poly) * (0, 0.33, 0.67, 1), resulting in eight folders: '0.0_mono', '0.0_poly', '0.3333333333333333_mono', '0.3333333333333333_poly', '0.6666666666666666_mono', '0.6666666666666666_poly', '1.0_mono', '1.0_poly'. A common split of 70% training, 10% validation, and 20% test set was used for distribution. The second method ignored the difference between mono and poly crystals, retaining only the probability information for folder distinction, resulting in four folders: '0.0', '0.33', '0.67', '1.0'. The divided images were then split into 52% for training, 23% for validation (if necessary), and 25% for testing, as recommended in the literature for a 75% training and 25% testing split. This was done to minimise the interference of the 0.66 category. If the proportion of 0.66 in the test set is too low, it leads to higher randomness in classification results, which is not conducive to further model optimization. Through extensive trials, different models excelled at different data types, as detailed later. Overall, this approach helps maximise the training outcomes of the model.After completing the data allocation, the training set data underwent data augmentation processing. Since the images are single-channel grayscale, the processing method differs from multi-channel. Our data augmentation method involved a series of transformations on the original images, including rotation, flipping, and cropping. Each newly generated image underwent scrutiny to ensure it retained its original label relevance while offering distinct variations. This meticulous process was crucial for extracting valuable information from the augmented images, thus enriching the dataset's diversity.

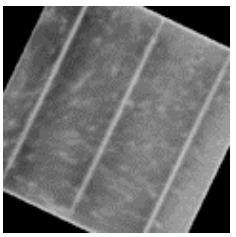

Fig.2:Rotated Images

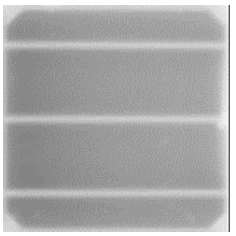

Fig.3: Images after solarization

Taking the four-class dataset as an example, to further enhance the dataset's robustness, we implemented a comprehensive augmentation plan aimed at ensuring each category contained at least 780 images. This included a combination of random rotations, solarization, and AutoAugment policies specifically tailored for similar datasets. Each image was augmented using these diverse transformations until the category directory reached the threshold. The augmented images were systematically saved with unique filenames that encapsulated the nature of the augmentation and the policy applied. The effect of the AutoAugment policy is shown in Fig. 4.

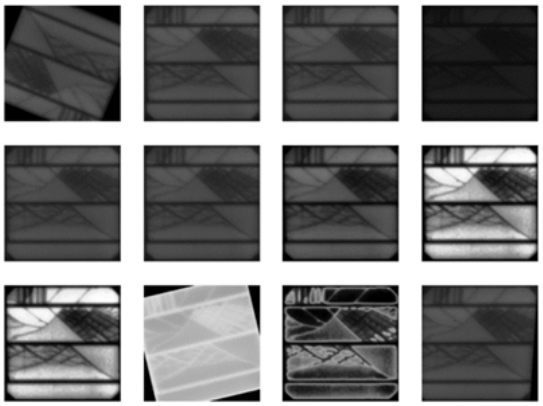

Fig.4:Images after AutoAugment policy

From the figures, it is evident that correct data processing can significantly enrich the training set and enhance model performance. Overall, this process serves two primary purposes. Firstly, it balances the dataset, ensuring a more equitable representation across classes. Secondly, it significantly enhances the generalisation capabilities of the machine learning models trained on this enriched dataset. By introducing a broader spectrum of image variations, the models are less prone to overfitting and better equipped for robust feature extraction. The result of these augmentation efforts is a notable improvement in recognition accuracy, highlighting the effectiveness of our chosen augmentation techniques.

After augmentation, taking the training set divided at a 52% ratio as an example: the total number of photos in the train set is close to 3200. Compared to the original train set of 1364 photos, this is an increase of 2.35 times.

TABLE II : Number of Images

Varying degree	0	0.33	0.67	1
Mono& Poly	784	153	55	372
Augmentation	784	800	800	800

The table shows the change in the number of photos. To maximise the use of existing training photos, no image augmentation was performed for the Mono&Poly 0 scenario, and all its samples were retained. Conversely, the 0.67 category was expanded nearly 14.5 times its original number using augmentation algorithms, thereby levelling the difference in the number of training images. This lays a solid foundation for subsequent improvements in model performance. The data augmentation methods for the eight-category dataset were essentially the same, with only slight differences in the final numbers, which are not elaborated further due to space constraints.

D. *Classification*

CNN Model:

CNNs are the cornerstone of most modern image processing and computer vision applications. They are particularly well-suited for image classification tasks because they can automatically and adaptively learn spatial hierarchies of features from input images [5].CNNs use convolutional layers to process data in a grid-like topology, such as pixels in an image. This allows the network to focus on local input patterns, making translation invariant and reducing the number of parameters compared to fully connected networks. This aspect is crucial for EL images of solar cells, which may have localised defects [6].

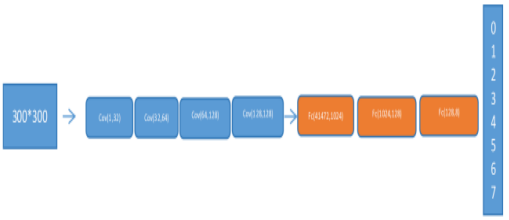

Fig.5: CNN Architecture

This CNN is built using the PyTorch library. It features three convolutional layers with 32, 64, 128, and 128 channels respectively, all using 3x3 convolutional kernels. Each layer is followed by a pooling layer. After the final flattening, it connects to three fully connected layers, with output sizes of 1024, 128, and 8, respectively, the last of which is used for classification. Each layer also uses the ReLU activation function to prevent gradient explosion.

Random Forest Model:

Random Forest is a versatile machine learning algorithm capable of performing both regression and classification tasks. It is an ensemble method that combines the predictions of multiple decision trees to improve generalizability and robustness against overfitting [7].

For the EL images, features such as pixel intensity distributions, gradients, and textures can be critical in determining the cell's health. Random Forest can capture these features effectively, and its ensemble nature makes it less prone to the overfitting that might occur if individual decision trees were used.

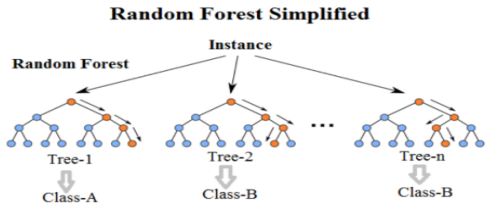

Fig.6: Random Forest Architecture

When using traditional machine learning, we flatten the preprocessed data and then feed it into a random forest model for training. We achieved satisfactory accuracy after adjusting the hyperparameters.

Model parameters: 1000 decision trees Maximum depth: 11"

Vision Transformer Model:

Vision Transformer (ViT) is a recent innovation that applies the transformer architecture, originally designed for natural language processing, to image classification tasks. It treats images as sequences of patches and can capture long-range dependencies across the image [8].In the case of PV cell defect classification, defects may be subtle and distributed across the cell. ViT can potentially capture such complex patterns through its attention mechanism, which weighs the importance of different patches in making a classification decision.

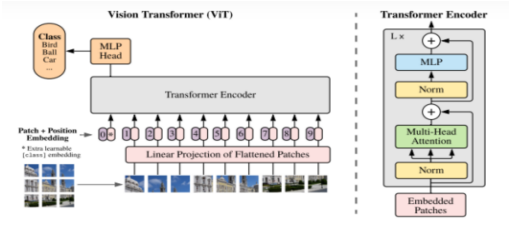

Fig.7: VIT Architecture

We are using the ViT (Vision Transformer) model, which has around 80 million parameters. This model replaces the convolutional layers in traditional CNNs with a Transformer structure. When using this pretrained model, we employed two strategies:

1) Update all parameters during training.

2) During training, add two fully connected layers at the end and only update the last two layers.

Comparisons revealed that the first strategy was more effective. Consequently, we obtained more objective results through multiple training sessions and tuning.

Resnet18 Model:

In our evaluation, we tried to use the ResNet model developed by He and colleagues in 2015[9]. This design draws inspiration from the VGG19 model by Simonyan and Zisserman, established in the same year [10], while innovating through the integration of residual units via shortcuts. Notably, ResNet diverges from the traditional approach by employing convolutions with a stride of 2 for the purpose of downsampling, and by substituting the conventional fully connected layers with a global average pooling layer. A key architectural principle of ResNet is its strategy to balance the network's computational complexity: whenever the spatial dimensions of the feature maps are halved, the model compensates by doubling the number of feature maps. This approach, along with the incorporation of shortcut connections, underpins the model's capacity for residual learning, which is instrumental in its ability to perform image classification tasks effectively.

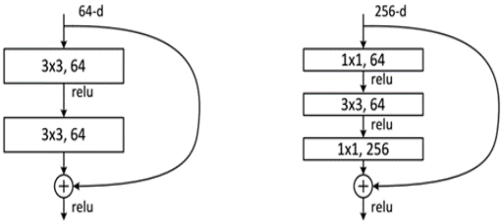

Fig.8: Resnet Architecture

VGG11 Model:

We also proceeded to evaluate the VGG11 model, a variant within the VGG network family, which was introduced by Simonyan and Zisserman in 2014[11]. The VGG11 model is characterised by its simplicity, using only 3x3 convolutional layers stacked on top of each other in increasing depth.

Reduction in volume size is handled by max pooling. Compared to its more complex counterparts in the VGG family, VGG11 has fewer layers, yet it still maintains a remarkable performance in image classification tasks. The architectural design follows a uniform pattern, which has been proven to be highly effective for visual feature extraction in various image recognition challenges. This model serves as a reference point for demonstrating the effectiveness of deep convolutional networks, illustrating that depth is a critical component for achieving advanced levels of performance in image classification tasks (Simonyan & Zisserman, 2014). The graph below shows the VGG11 architecture.[12]

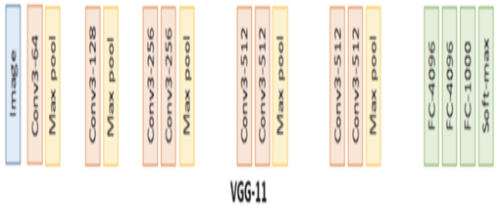

Fig.9: VGG11 Architecture

III.EXPERIMENT RESULTS

CNN Result:

Parameter Setting:
Optimizer: AdamW
Learning Rate: 5e-5
Loss Function: CrossEntropyLoss"

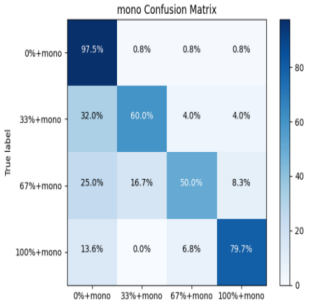

Fig.10: Mono Confusion Matrix

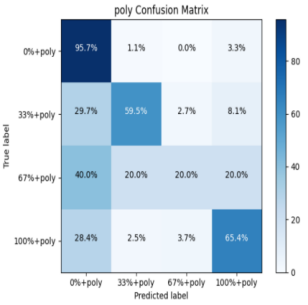

Fig.11: Poly Confusion Matrix

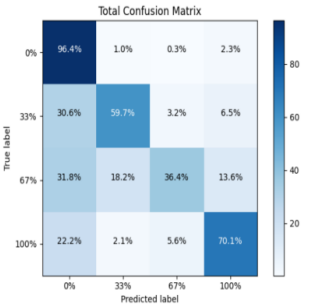

Fig.12: Total Confusion Matrix

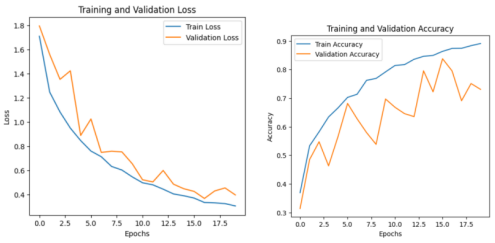

Fig.13: Loss and Accuracy

Random Forest:

Parameter Setting:
1000 decision trees
Maximum depth: 11"

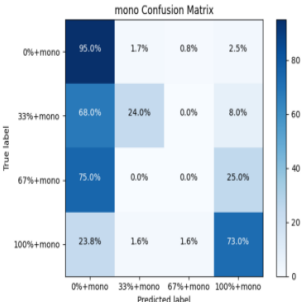

Fig.14: Mono Confusion Matrix

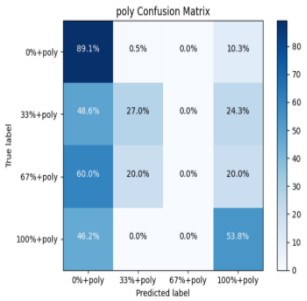

Fig.15: Poly Confusion Matrix

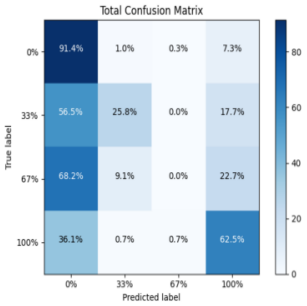

Fig.16: Total Confusion Matrix

Resnet Result:

Parameter Setting:

modellr = 2e-4

BATCH_SIZE = 64

EPOCHS = 16

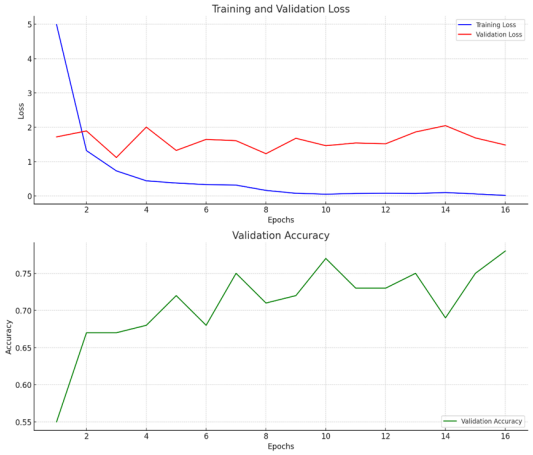

Fig.17: Loss and Accuracy

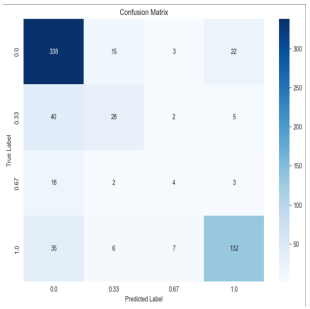

Fig.18: Confusion Matrix

Training and Validation Loss Graph (Top):

The training loss (blue) shows a steep decline in the beginning, indicating rapid learning, and then gradually decreases, suggesting the model is approaching a minimum in the loss function.

The validation loss (red), after initial improvement, fluctuates and even increases at certain points. This could imply overfitting or the model's sensitivity to the validation sets variance.

Validation Accuracy Graph (Bottom):

The validation accuracy (green) increases from the first epoch, demonstrating an overall upward trend with some fluctuations. The highest accuracy is observed in the final epoch.

The fluctuations may indicate that the model's ability to generalise is not stable across epochs, and the validation set may contain challenging or diverse examples that affect performance differently at each epoch.

For the 4x4 confusion matrix.The model predicts class with high accuracy, with 338 correct predictions, but also misclassifies 15 as , 3 as , and 22 as ‘0.0’,’0.33’,’0.67’,’1.0’

For class, the model shows considerable confusion, with 40 instances correctly predicted, but a significant number (28) are misclassified as class ’0.33’,’0.0’

Class also has low correct predictions (2), with more instances misclassified as class (18) and (3)’0.67’,’0.0’,’1.0’

Class shows a better prediction rate with 132 correct classifications, but there are still misclassifications where 35 instances are predicted as class ‘1.0’,’0.0’

Vit Result

Parameter Setting:

Optimizer: AdamW

Learning Rate: 5e-5

Loss Function: CrossEntropyLoss"

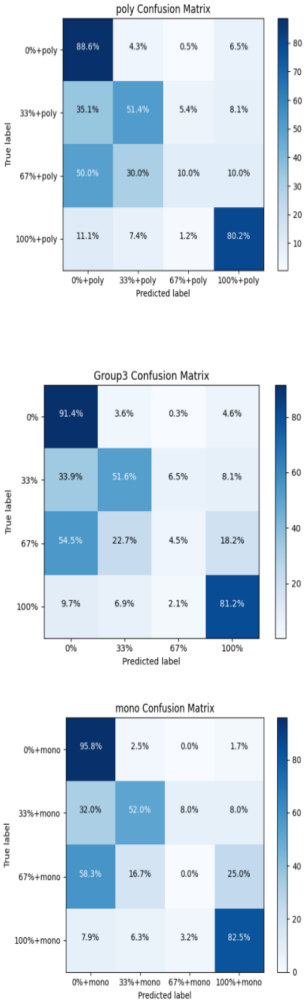

Fig.19: Confusion Matrices of ViT

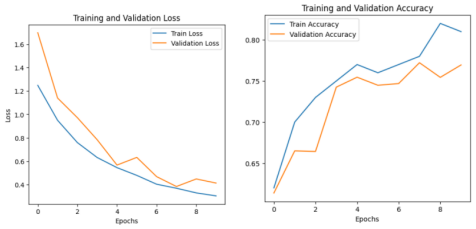

Fig.20: Loss and Accuracy

VGG Result

Parameter Setting:
batch_size=16
lr=0.001
num_epochs=15

Fig.21: Loss and Accuracy

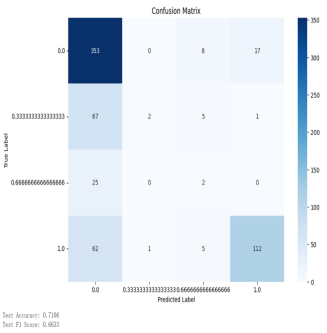

Fig.22: Confusion Matrix

Above graphs plot the changes in loss and accuracy for VGG11 model across 15 epochs of training.

Loss Graph: The training loss (blue) and validation loss (red) both generally trend downwards, indicating the model is learning. However, fluctuations in validation loss suggest variability in the model's performance on unseen data.

Accuracy Graph: Both training accuracy (blue) and validation accuracy (red) increase over time, a sign of improvement. Notably, validation accuracy surpasses training accuracy at certain points, which may indicate the model's good generalization or possibly an effect of regularization techniques.

For the 4x4 confusion matrix.The majority class (true label 0.0) has a high number of correct predictions (353), which shows that the model is effective at identifying this class.

However, for the other classes (true labels approximated to 0.33 and 0.67), there are fewer correct predictions (67 and 25 respectively), and a significant number of misclassifications.

The class corresponding to true label 1.0 shows a mix of correct predictions (112) and misclassifications.

TABLE III: Results of Models

Model	Accuracy	F1 Score
Random Forest	71%	0.45
CNN	88%	0.80
ViT	81%	0.59
ResNet	76%	0.56
VGG	72%	0.66

Regarding Preprocessing:

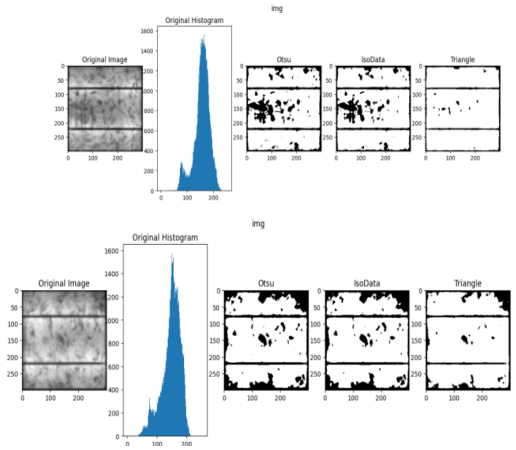

Fig.23: Preprocessing

In terms of image processing, we selected a few images with higher misclassification rates for analysis and found that image preprocessing is crucial for classification. After selectively extracting features from the images, the accuracy of the model we trained also improved. However, this improvement is more pronounced compared to traditional machine learning methods. For neural network models, the improvement in image processing is not significant (one reason is that the prediction results are inherently better than traditional machine learning methods, and secondly, models like ViT and RestNet, their convolutional layers are essentially an abstract form of feature extraction).

When dealing with images with 66% completeness, they are easily misclassified as 0%.

Regarding Model Training:

In our model training, it's not hard to notice that traditional machine learning, such as the random forest we used, is quite sensitive to the settings of hyperparameters. The accuracy can vary greatly, for example, we need to limit the depth to prevent overfitting, and set a safe number of decision trees and their depth.

Secondly, among the above models, the accuracy of processing data at 100% and 0% is quite high. However, for the classification of images at 33% and 66%, it is not as clear, especially the 66% data which is more likely to be misclassified as 0%. This poses a challenge in tuning the models, particularly when dealing with datasets that have diverse features and imbalanced distributions. For instance, the ViT model may require a large amount of data for effective training, while random forests might be more susceptible to the effects of imbalanced data.

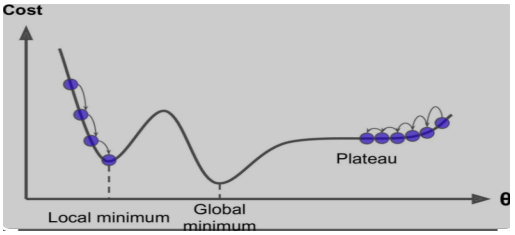

Fig.24: Global Minimum

Regarding Accuracy:

In our model training process, with suitable hyperparameters, optimizers, and loss functions, the accuracy of the ViT model was the most stable (around 80%), while the accuracy of the random forest was about 71%, and RestNet was around 76%.

The accuracy of the CNN model was around 78%, which was relatively unstable. At times, the training set accuracy could reach as high as 88%! After group discussion, we believe the reasons are as follows:

Appropriate model parameters: Since ViT is a large model with up to 80 million parameters, the adjustments during backpropagation are not significant. Therefore, it's likely to fall into local optima during training and cannot "jump out". For our CNN model, with about 40 million parameters, it has a better chance of reaching the global optimum, so the accuracy fluctuates more.

Model Architecture: Since ViT is a pretrained model, it has strong generalization capabilities and is not very specific, so the results obtained by updating parameters on an already trained base during backpropagation are very stable.

In contrast, all parameters of a CNN are randomly initialized, which might be more easily

adjusted to suit the dataset, thus achieving higher accuracy in certain cases. Therefore, it sometimes falls into an optimal solution.

IV. CONCLUSION

This report delves into the feasibility of using machine learning and deep learning technologies for effective classification of defects in photovoltaic cells. Utilizing electroluminescence imaging technology to analyze the ELPV dataset, we successfully applied various algorithms, including CNN, random forest, ViT, RestNet, and VGG. The experimental results of these methods indicate that these advanced technologies can significantly enhance the accuracy and efficiency of defect detection.

Additionally, our study discusses the strengths and limitations of different models in processing image data and how these factors affect the ultimate classification accuracy. We focused on discussing the differences in accuracy and stability between ViT and CNN models, as well as the capability and efficiency of other models, including random forests, ResNet, and VGG networks, in handling imbalanced datasets. Each model has its unique advantages and limitations.

Further research on how to optimise models to handle specific types of defects, especially in situations of data imbalance.

Therefore, we believe that more exploration into model fusion or integration methods is warranted to combine the strengths of different models. For example, an attempt could be made to combine the features of CNN and random forests to achieve more accurate defect classification.

REFERENCES

[1] Efficient Deep Feature Extraction and Classification for Identifying Defective Photovoltaic Module Cells in Electroluminescence Images. "Expert Systems with Applications," vol. 175, Aug. 2021, Art. no. 114810.

[2] Deep Learning System for Defect Classification of Solar Cells. "IEEE Xplore." Available: https://ieeexplore.ieee.org/document/10008277. DOI: 10.1109/CICN56167.2022.10008277.

[3] Another Study Using a Large-Scale Dataset of EL Images for Binary and Multi-Class Classification. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC9138174/.

[4] H. M. Al-Otum, "Deep Learning-Based Automated Defect Classification in PV Cells," vol. 58, Oct. 2023, 102147.

[5] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in "Advances in Neural Information Processing Systems (NIPS)," 2012.

[6] I. Goodfellow, Y. Bengio, A. Courville, and Y. Bengio, "Deep Learning," vol. 1, MIT Press, Cambridge, 2016.

[7] L. Breiman, "Random forests," "Machine Learning," vol. 45, no. 1, pp. 5-32, 2001.

[8] A. Dosovitskiy et al., "An image is worth 16x16 words: Transformers for image recognition at scale," arXiv preprint arXiv:2010.11929, 2020.

[9] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition, 2015.

[10] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition,2015.

[11] Simonyan, K., & Zisserman, A. (2014). Very Deep Convolutional Networks for Large-Scale Image Recognition. arXiv:1409.1556.

[12] S. P. Singh and M. Jaggi, "Model Fusion via Optimal Transport," arXiv:1910.05653v1 [cs.LG], Oct. 2019.