

Received April 27, 2022, accepted May 14, 2022, date of publication May 27, 2022, date of current version June 8, 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3178588

A Gradient Guided Architecture Coupled With Filter Fused Representations for Micro-Crack Detection in Photovoltaic Cell Surfaces

MUHAMMAD HUSSAIN¹, TIANHUA CHEN¹, SOFYA TITRENKO², PAN SU^{3,4},
AND MUFTI MAHMUD⁵, (Senior Member, IEEE)

¹School of Computing and Engineering, University of Huddersfield, Huddersfield HD1 3DH, U.K.

²School of Mathematics, University of Leeds, Leeds LS2 9JT, U.K.

³School of Control and Computer Engineering, North China Electric Power University, Baoding 071003, China

⁴Hebei Key Laboratory of Knowledge Computing for Energy and Power, Baoding 071002, China

⁵Department of Computer Science, Nottingham Trent University, Nottingham NG1 4FQ, U.K.

Corresponding author: Tianhua Chen (t.chen@hud.ac.uk)

ABSTRACT This paper presents a shallow architecture based on Convolutional Neural Networks (CNN) for detecting Micro-cracks in Photovoltaic (PV) cells within the manufacturing environment. Based on Electro Luminescence (EL) imaging principles, this research presents a mechanism for determining the number of filters within the convolutional blocks, gradient guided filter tuning (GGFT). Observing the similarity between the original EL images and the filter output images obtained via GGFT, the research further introduces a mechanism for generating PV cell images based on EL Modelling, termed Filter Fused Data Scaling (FFDS). The effectiveness of both techniques is presented by benchmarking our developed architecture against ‘off the shelf’ augmentations and State-of-the-Art (SOTA) networks. The performance criteria was widened to include accuracy, computational, architectural, and post-deployment metrics. The high performance of our architecture in an intensive and wide-scoped evaluation demonstrates the high efficacy of our proposed mechanisms for developing PV-specific architectures and addressing the issue of data scarcity, particularly the difficulty in the procurement of quality EL images from the manufacturing site.

INDEX TERMS Architectural complexities, electroluminescence modelling, filter fused augmentations, filter tuning, micro-crack, photovoltaics, convolutional neural network, deep learning.

I. INTRODUCTION

The reduction of global emissions is amongst a handful of objectives accepted by the majority of nations across the World. The emergence and continuous growth of solar-powered installations are widely accepted as an alternative to conventional power generation sources like coal [1]. To signify the effectiveness of solar power and mitigation of CO₂ emissions, an example is provided from a solar installation initiative in California where 113,533 home-based solar installations have reduced 696,544 metric tons of CO₂ emissions [1].

Along with the sun, solar cells are one of the primary components found within a Photovoltaic (PV) installation. The purpose of these cells is to transform light energy into electrical energy. The cells effectiveness to serve their

purpose of energy conversion is highly dependent on the production quality of the solar cells. As with many production industries, quality control is a key part of the process. Like many other production processes, PV cell production is also exposed to the generation of various defects such as scratches, material defects, dirt, and the infamous cracks. Cracks, more commonly known as Micro-cracks, are one of the most common types of defects originating from mechanical or thermal stress during fabrication [2]. This type of defect can result in the breakdown of electrodes leading to the obstruction of current collection and transmission, developing fragments or hotspots on the cell surface that impact the cell’s performance [3].

The trend of automation and lessening human workload can also be observed in the PV manufacturing industry. However, specific tasks within the process of solar cell manufacturing are still heavily dependent on human interaction. One of these tasks is the quality control of cells at

The associate editor coordinating the review of this manuscript and approving it for publication was Alon Kuperman¹.

production lines, i.e., detecting and rejecting defective solar cells. The subtle nature of various cracks can make identifying these defects through human inspection challenging. Furthermore, defect detection of more complex and minute damage requires domain expertise which can be expensive and time-consuming. The human-led inspection also has an element of human error. The bias and over reliance on human-oriented quality control can lead to a higher rate of defective PV cells making their way into installations, resulting in poor performance. Based on this premise, we observe an increase in active research into developing intelligent systems to detect defective cells, requiring no or minimal human involvement [4].

Micro-cracks are one of the most challenging when it comes to defect detection. This is due to the inability of the human eye to directly observe micro-cracks without requiring assistance from other means [5]. Therefore, Electroluminescence (EL) imaging is one of the most widely used techniques for the detection of micro-cracks along with various types of defects in multi-crystalline PV cells in the present times [6], [7].

Although the detection of Micro-cracks via deep learning in particular computer vision is an active field of interest as evident from the subsequent section on literature review, there is a lack of a ‘systematic design approach’ for developing and justifying architectures that can not only provide a high degree of accuracy but also can be deployed onto the production floor through an edge device. By a systematic design approach, we refer to the justification for the selection of components within the designed architecture and how it impacts the overall computational and inference efficacy of the network. For example, convolutional filters are a key component within the Convolutional Neural Network (CNN), however in most cases as evident from the literature section, the selection of filters seems to be arbitrary hence it’s impact on the networks computational and architectural efficiency cannot be maximized. We address this issue by presenting a novel filter determination process named Gradient Guided Filter Tuning (GGFT), for assisting PV researchers with the development of internal convolutional blocks for their respective architectures. We showcase how the implementation of the GGFT process can assist with the selection of an appropriate number of filters for each convolutional block hence keeping a check on the overall computational load of the network. We also present an additional mechanism for the generation of representative PV cell augmentations named Filter Fused Data Scaling (FFDS) as an alternative to the use of generic augmentations such as flipping, rotating. We showcase how the implementation of our novel FFDS process for data augmentation outperforms generic data scaling. A high-level comparison of the conventional CNN design approach and our contribution is provided in Figure 1.

A. LITERATURE

To address the shortcomings attached to human inspection and improve the production output efficiency of solar

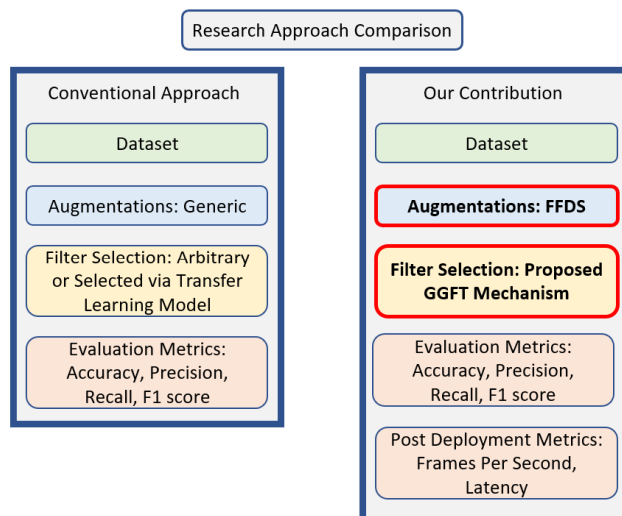


FIGURE 1. Research approach comparison.

cells, researchers have been exploring the use of Artificial Intelligence (AI), especially Computer Vision (CV). Akram *et al.* [8] propose a convolutional neural network (CNN) architecture to detect defective PV cells. The authors share the results of an ‘isolated-model’ (98.67%), i.e., trained on only EL based images and then use transfer learning to tune the ‘isolated-model’ for Infrared (IR) based images achieving an overall accuracy of 99.23%. The dataset consists of ~800 images, hence the authors argue that a deep architecture would result in overfitting. The research is split into two distinct phases.

The first phase is based on developing a CNN using EL images. In the second phase, the EL trained CNN is used as a pre-trained model that is fine-tuned on Infrared Images (IR) of defective cells. The pre-trained model (EL-based) achieved an accuracy of 98.67%. The reported accuracy is impressive, however, looking deeper into the methodology primarily focusing on data collection and pre-processing, we find that defects were artificially placed onto the cell images rather than actual defects. Initially, our research also contemplated the use of artificially generated ‘cracks’ placed onto images of solar cells. However, we found the ‘generated cracks’ when placed next to an actual cell containing a ‘micro-crack’; a significant difference was observed due to the complexity of various cracks (discussed in the methodology section). Therefore, a dedicated ‘defect-generator’ would be required for this type of approach to effectively capture the underlying features of the cracks that correspond to real cracks found within solar cells. We could remove this step by implementing selective data augmentation techniques paired with specific regularization methods. We achieved a recall rate of 99.20%. At the same time, we achieved this without manually creating a ‘crack-generator’ to train an isolated model before fine-tuning on the data of interest. The authors also opt for data augmentation to focus on increasing the scale and variance of the dataset. Understandably, this leads to an increase

in the model's accuracy by 6.5%. Looking at the CNN architecture itself, the authors, through empirical testing, decide on a four-block CNN network with a fully connected layer feeding into a SoftMax function.

Deitsch *et al.* [9] propose an SVM and a CNN network for various defect detection in EL based solar cell images. The author claims that both models provide high accuracy, (SVM; 82.44% and CNN; 88.42%). Before scrutinizing the methodological approach for the top performer (CNN), it's worth mentioning that the advancements in deep learning along with data scaling, transformation techniques and regularization means that models performing under 90% are not seen as ground-breaking. Investigating the CNN itself, we find that the authors subscribe to the transfer learning domain through the VGG-19 architecture. The authors carry out the fine-tuning in two stages. The first stage consists of randomly initialising the weights of the fully connected layer using ADAM as the optimizer for weight updation. The second stage involves the random weight initialization of the fully connected layer and the preceding convolutional layers. The author mentions Stochastic Gradient Descent (SGD) for the second stage and mentions that the 'Momentum' parameter was set to 0.9. It is essential to mention that 'Momentum' is a hyperparameter used in SGD-M, which is different from SGD. Furthermore, it is unclear why replacing the ADAM as the optimizer was required for the second stage.

Ahmad *et al.* [10] propose a CNN architecture for detecting defects in EL based solar cell images with an accuracy of 91.58%. Before discussing the architectural considerations, the authors highlight the importance of data augmentations in scaling the dataset and adding variance. The selected CNN architecture is initiated with the input image, followed by 4 convolutional blocks containing 32 filters each. The next two convolutional blocks increase the number of filters to 64, while the final two contain 128 filters. In total, the developed CNN architecture has 8 convolutional blocks, followed by a single fully connected layer, feeding into the output. Although it is a general rule of thumb to increase the number of kernels (filters) as the model gets deeper, the rationale for 8 convolutional blocks is unclear. An intuitive explanation is also lacking; for example, was the number of convolutional blocks driven by the significant variance in the nature of the faults?

Furthermore, the selection of the learning rate as 0.001 seems to be selected as the default choice rather than through optimization. By experimenting, for example, with the learning rate, the authors may have found the model is able to maintain its accuracy with an increased learning rate, i.e., 0.02, resulting in faster training time. Contrary, our research shows the importance of selecting hyper-parameters based on the specific dataset in question rather than empirically proven default parameters.

Tang *et al.* [11] propose a CNN to detect defects in EL based cell images. The authors claim their first key contribution is implementing a Generative Adversarial Network (GAN) network for the data augmentation. It is unclear

as to the rationale for using a GAN network for scaling and injecting more variance into the dataset. A comparison of accuracy between GAN and the use of standard data augmentation techniques already presented in deep learning frameworks such as TensorFlow, Pytorch, and Keras may have helped justify the use of GAN. However, the fact that the overall accuracy after using GAN for data augmentation was 83% shows the ineffectiveness of the technique for this particular case.

Furthermore, as GAN is a network used for the generation of new images, it is much more computationally demanding than using standard data augmentation techniques, and hence an unnecessary allocation of resources are required. On the positive side, the authors provide adequate findings on their experimentations for selecting the number of kernels. Based on their findings, the authors explain how increasing the number of kernels can improve the model accuracy significantly to a certain extent. After which, the increase in kernels will not positively impact the model but can instead lead to overfitting.

Dunderdale *et al.* [12] demonstrate a feature-based and deep learning approach to detect defective solar cells. As we are interested in deep learning, we will focus on critiquing the methodology of developing the deep learning models. For comparison, the authors train the PV dataset on the VGG-16 [13] and Mobilenet [14] architectures. It is appreciated that the authors not only use ADAM as the 'off-the-shelf' optimizer but rather compare results for both SGD and ADAM. Looking in detail at the VGG-16 trained architecture, we find the best performance (85.8%) was through implementing an SGD optimizer with data augmentations; Horizontal-vertical flipping and rotations. The ADAM optimizer provided an unacceptable accuracy of 27.4% for the same settings.

On the other hand, Mobilenet architecture achieved the highest accuracy (89.5%) with the data augmentations applied as 'Horizontal-vertical' flip and ADAM as the optimizer. The authors do not explain the 'paradigm-shift' in results after changing between the two optimizers and architectures. It is understood that Mobilenet is more computationally effective and lightweight because depth-wise convolutions are applied as opposed to the standard convolutions, reducing computations by as much as 9-folds [15].

Pierdicca *et al.* [16] propose a CNN based on the VGG-16 architecture to detect defective PV cells. The authors provide simplicity of implementation as one of the reasons for justifying the selection of VGG-16. However, many SOTA pre-trained models are now facilitated in a user-friendly manner by many frameworks such as Pytorch and TensorFlow developed by Facebook and Google, respectively. Therefore, rather than the simplicity of implementation, the selection of the architecture should be based on the characteristics of the dataset. The author confesses the implications of the VGG-16 network selection as the lack of batch normalisation within the convolutional layers of the network. To our understanding the authors don't provide any computational information, if

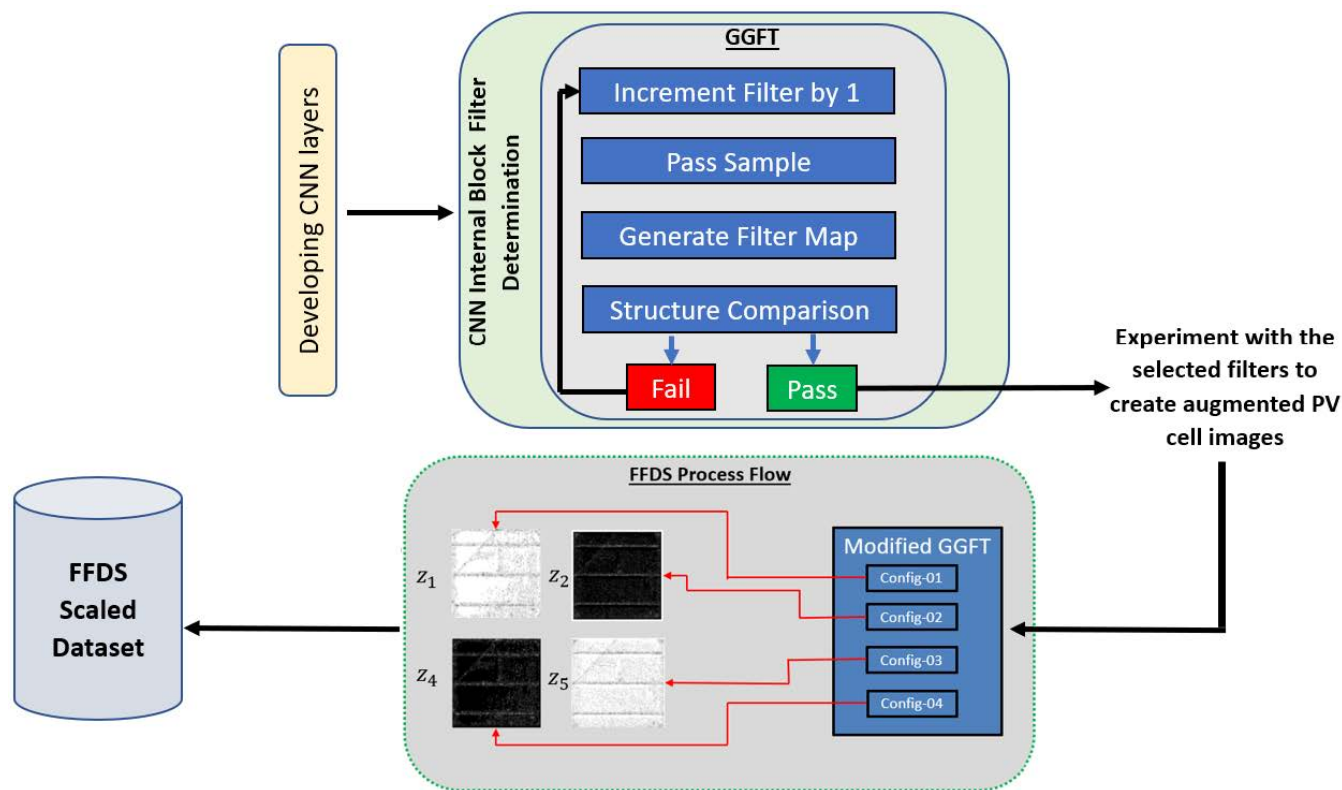


FIGURE 2. GGFT and FFDS coupling overview.

this was provided, we feel the model training and convergence time would be significantly higher as compared to models implementing batch normalisation [17]. The author mentions that data augmentation improves the model performance; however, the results are accepted as modest, and the explanation for this is given as the significant imbalance in the defective images. The work could have further explored tuning the model for better accuracy, i.e., by adjusting the learning rate rather than using 0.001 only.

B. CONTRIBUTION & PAPER ORGANISATION

Our first contribution is developing a shallow CNN architecture for early detection of Micro-cracks within a PV manufacturing complex. We present a mechanism, Gradient Guided Filter Tuning (GGFT), for determining the number of filters within each convolutional block to achieve high performance with limited infrastructure. The process can be seen as an intersection between saliency mapping and the process of EL modelling. We demonstrate how by developing PV domain logic around the concept of gradient mapping, we can obtain a highly efficient architecture in capturing the underlying characteristics of the dataset. The development of the configurable parameters, defined with the GGFT flow, will enable PV developers to tune their respective architecture designs based on the type of PV surface faults they are factoring for.

Inspired by the similarity between EL inputs and filter outputs, dictated by the defined parameters within the GGFT process, we introduce an additional mechanism, Filter Fused Data Scaling (FFDS), for the generation of EL based

augmentations of PV cells. These augmentations are dictated by the same configurable parameters presented in GGFT, allowing PV developers to scale and inject more variance within their datasets. The performance comparison between the FFDS and generic ‘off the shelf’ augmentation options such as Random Erasing demonstrates the effectiveness of our proposed mechanism, outperforming the latter in all metrics.

Another advantage of the proposed FFDS process is its ability to address the issue of data acquisition. Data is the most basic requirement and the inception point for developing image classification architectures. It is not always possible for PV developers to directly gain access to PV manufacturing sites for collecting cell images to an acceptable level of quantity and quality. By implementing our proposed FFDS process, PV researchers would have the luxury of generating representative PV cell images by experimenting with the filter configurations of the architecture developed via the GGFT process.

The developed architecture based on GGFT and data scaling via FFDS is benchmarked against SOTA networks on computational, architectural, and post-deployment metrics, performing highly in all settings. Hence, we feel the presented mechanisms will allow PV developers to develop and tune architectures with respect to the type of PV cell fault(s) they are researching.

Figure 2 presents a high-level process flow for the two proposed methodologies. Notice how the FFDS process is an extension of the GGFT process enabling the generation

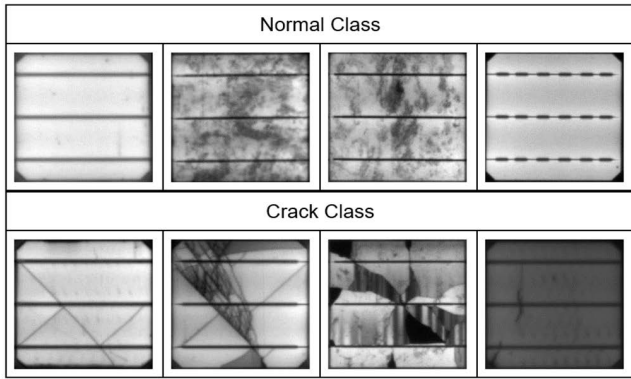


FIGURE 3. Internal variance found within the classes.

TABLE 1. Complete dataset.

Normal	307
Crack	470

TABLE 2. Dataset status post cross-validation split.

Class	Training	Validation
Normal	246	61
Crack	376	94

of representative samples post network defining via GGFT.

II. METHODOLOGY

A. DATASET

The environmental context of the images is based within a PV manufacturing factory. The original data consists of two types of PV cells: normal and defective, shown in Figure 3. The normal cells have no defects and can be expected to operate as per their specification. Contrary, the defective cells contain crack(s) of varying size and characteristics that, if signed off through the quality control process, will negatively impact the PV system’s performance post-deployment. It is clearly observable that both classes contain PV cells that significantly differ in their visual appearance but belong to the same class. Taking the normal class as an example, we observe that the first PV cell (far right) can intuitively be distinguished as a normal cell. However, the next two PV cells also belong to the normal class; but due to internal shading, can be misclassified as defective ones when lacking domain expertise. Another key finding was the variance in the busbar structure. It can be observed in Figure 3 that we have three types of busbar configurations; solid lines across the cell face, solid-line cut-off at cell ends and periodic cut-off lines represented in the normal class (far right).

The status of the dataset before splitting into training and validation sets is shown in Table 1.

The dataset was split into five folds containing training and validation sets for both normal and crack class. The training class contained 376 samples of ‘crack’ cells and 246 of ‘normal’ cells. Similarly, the validation set contained 94 ‘crack’ cells and 61 ‘normal’ cells for each fold. The status of the dataset post cross-validation is shown in Table 2.

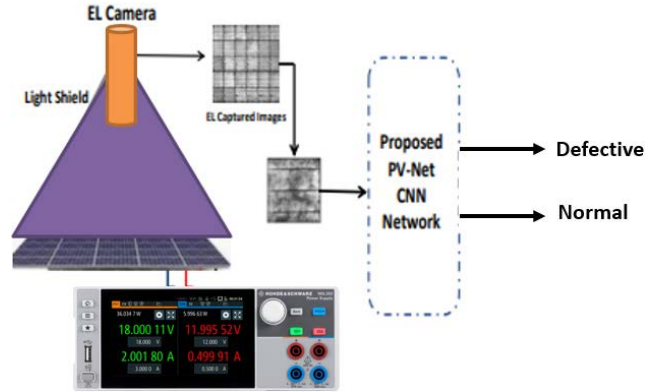


FIGURE 4. Experimental setup for EL cell extraction.

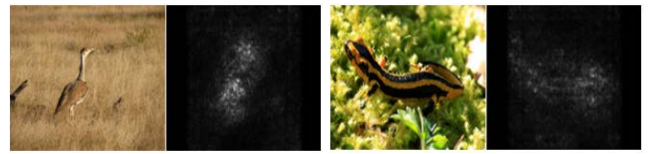


FIGURE 5. Saliency mapping from [18].

B. EL EXTRACTION PROCESS

The EL imaging process is essentially a measurement technique applied to analyze solar cells. Applying current via an external power source forces the cell to reverse its operation. The introduction of current leads the cell to emit light that is not within the visible spectrum, residing around the range of 1100nm. As a result, EL tuned cameras such as charge-couple devices (CCD) are commissioned for capturing the emitted light. The dark shield plays a critical role in minimizing reflections caused by external light sources. When inspecting the EL images, quality inspection personnel look for darker regions within the image, indicating potentially faulty segments.

C. GRADIENT GUIDED FILTER TUNING

With the above premise, we aim to model the EL process through the backpropagation of gradients, without weight optimization to assist with the development of the architecture, in particular the number of filters required. Simonyan et al. [18] proposed gradient-based saliency mapping for initiating GraphCut-based object segmentation models. The technique essentially enables some level of internal layer interpretability by visualising the learning of filters for a given class. As shown in Figure 5, it can be understood that the regions within the image containing the class of interest have a higher pixel magnitude. However, as mentioned by the authors, the process is termed as ‘weakly supervised’ as it looks at the filter output by simply projecting the gradients back onto the image without any optimization.

Furthermore, by observing Figure 5, it can be concluded that the interpretability is of very high abstraction, and by simply following the filter output in the absence of the actual

input image, it is very difficult for determining what the object may be or even a high-level abstraction concept of the type of object due to the complexity and the high dimensionality that comes with real-world objects.

However, the level of complexity within the produced images is reduced significantly due to the EL process. Hence the challenge is to differentiate between the class of interest and limited variance, namely busbar configurations, shading and light intensity.

Therefore, the aim was to develop a model flow mechanism that would effectively model the input to the filter gradients enabling filter tuning to determine how many filters should be used within each convolutional block.

A filter is the most fundamental component of the convolutional block. It's the first architectural component to interact with the input images via element-wise multiplication for generating transformations and initiating the process of learning key characteristics.

The importance of this component (filter) begs the question, is there a framework or a mathematical model generally used to determine the number of filters required within each convolutional block? This is answered by pointing out that the determination of filters and many other components is based heavily on the assumptions derived from the inspection of the dataset and prior domain knowledge. However, in our case, we demonstrate that by modelling the EL process into our filter design, we can obtain strategic interpretability that allows us to tune the number of filters we require. Our Gradient Guided Filter Tuning (GGFT) process enabled us to analyze pixel attribution through domain-based logical parameters for determining the number of filters required within the convolutional process. The GGFT process is shown in Figure 6.

The process is initiated by taking a sample image D_1 and setting the filter parameter f_{1n} , to one, dictating the number of filters. The second convolutional block contains $2 * f_1$ filters (doubling of the filters with respect to the first block) as the feature extraction level becomes more focused with the increased architectural depth. A forward pass is carried out initiated by an input image. Once the forward pass is complete, the obtained gradients are not sent to the optimizer for weight optimization, but rather the gradients with respect to the scoring class are backpropagated unchanged through the network providing the same input matrix D_{1E} , as the original image D_1 . The two images are then processed through a 'Structure Comparator', referring to a set criterion for input and output evaluation. In our case, the Structure Comparator had two objectives that D_{1E} must fulfil before accepting the filter specification. Firstly, the defined number of filters must have the capacity to reproduce the basic cell structure and secondly the pixel gradients of potential false activations must be diminished.

The Structure Comparator can be seen as an AND gate, so the filters would be adjusted if any of the criteria were not fulfilled. The process would repeat until both criteria are reached and the filters are frozen. It's important to mention that the set criteria for the structure comparator is domain

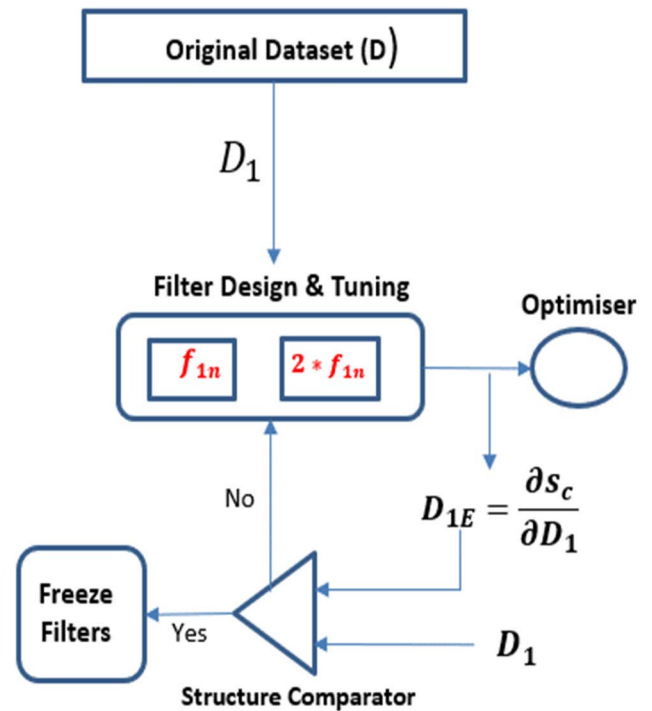


FIGURE 6. Proposed GGFT process flow.

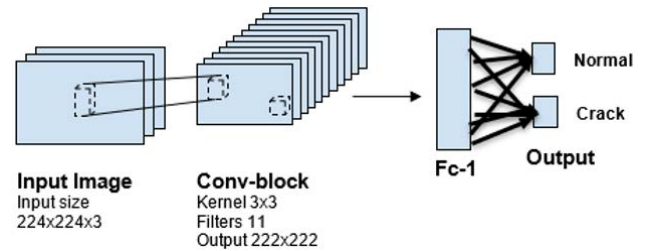


FIGURE 7. Initial model block design.

specific. Hence, an adequate level of domain knowledge is required for setting the objectives needed for the structure comparator.

The raw gradients are a fundamental component of the process proposed above and are obtained via the extraction of the first-order derivative from the Taylors Series Representation Theorem. The theorem states that by knowing the function and its derivative at point 'a', we can estimate the function at another point. Looking at the Taylors estimation in terms of CNN:

$$f(a) + \frac{f'(a)}{1!}(x-a) + \frac{f''(a)}{2!}(x-a)^2 + \dots$$

Here $f(a)$ represents the trained CNN, and the goal is to obtain the salient features within each filter, showing what part of the image was learnt by each filter resulting in the final classification. As we aimed to use this concept to unearth local importance approximations, we decided to focus on the

first part of the expression.

$$f(a) + \frac{f'(a)}{1!} (x - a)$$

Furthermore, we wanted to implement the expression for assisting with the development of the network feature maps rather than assist with model training, hence:

$$\frac{f'(a)}{1!}$$

So essentially, we take the first-order derivatives, i.e., gradients for a single forward pass. Where $f'(a) = \frac{\partial S_c}{\partial I}$, $a =$ selected image and $x =$ image variable (I):

$$\left. \frac{\partial S_c}{\partial I} \right|_I = I_0$$

D. INITIAL ARCHITECTURE DESIGN

The first iteration of the architecture consisted of only one convolutional block comprising of 11 filters, followed by a fully connected layer consisting of 40 neurons. As mentioned before, there is no general rule for initiating with a specific number of filters; however, our proposed GGFT mechanism coupled with the domain knowledge would assist with evaluating the capacity of the selected filters and dictate lowering or increasing the number of filters.

Reiterating the point that GGFT does not require weight optimization; hence, an optimizer was not required at this stage. Two images from the defective class were selected with high dissimilarity in the degree of damage. The purpose was to perform a forward pass on the selected image, obtain gradients with respect to the class of interest, backpropagate the gradients for the selected class onto the original image. This would reveal the regions of importance determined by the architecture.

The resultant filter output for each filter iteration based on GGFT is presented in Figure 8. Initiating the process with only a single filter, it can be observed that the network was partially successful in reproducing the basic structure of the busbars and considered these pixels as the most important in influencing the scoring class. Clearly, an increase in the number of filters was required to unearth at least the basic structure of the cell. As the filter capacity increased, the network was able to reproduce the busbar configuration. However, it was still unable to provide any importance to the pixels within the upper-left region containing the Micro-crack. By the sixth iteration containing 11 filters, the actual busbar configuration was also diminishing in starkness, hinting towards increasing the overall network capacity through the advent of another hidden layer. Furthermore, the fluctuations in the shading indicated that the model was struggling with grasping the underlying structure of the PV cell surface.

Before increasing the network capacity, we decided to test with an input image containing a higher degree of surface deflection. It can be seen from Figure 9 that even after providing a cell with major defects present on the surface, the network was unable to interpret these defects through the scoring function.

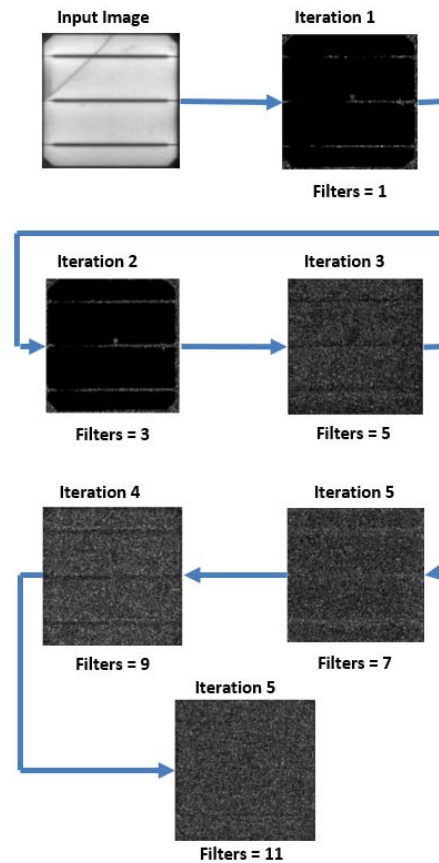


FIGURE 8. GGFT output maps.

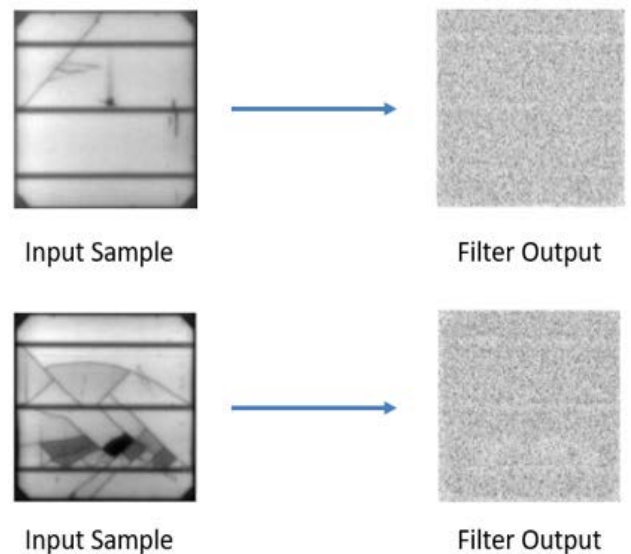


FIGURE 9. GGFT output comparison against increased surface defect.

E. MODIFIED ARCHITECTURE

We decided to double the number of filters in the second convolutional block with respect to the number of filters in the preceding convolutional layer. The rationale for this was that the deepening of the network would enable the architecture to

Layer	Output Shape	Calculate Learnable Parameters	Total learnable Parameters
Input	3,224x224	N/A	N/A
Conv1	11,222x222	(3x3x3+1)x11	308
ReLu	11,222x222	N/A	N/A
Max-pool	11,111x111	N/A	N/A
Conv2	22,109x109	(3x3x11+1)x22	2,200
ReLu	22,109x109	N/A	N/A
Max-pool	22,54x54	N/A	N/A
Fc1	100 neurons	(54x54x22+1)x100	6,415,300
ReLu	100 neurons	N/A	N/A
Fc2	50 neurons	(100+1)x2	102
Total Learnable Parameters			6,422,960

FIGURE 10. Architecture block diagram.

learn key characteristics for unearthing the overall cell structure and then determining the micro-crack characteristics. Each convolutional block consisted of a convolutional layer, activation function and max pooling. The initial convolutional layer included 11 filters, followed by 22 in the next, based on the GGFT process, with spatial dimensions of 3×3 pixels.

The number of resultant feature-maps were attained through the equation below. A term introduced in the formula is 'P', referring to 'padding. Padding is the implementation of 'zero' rows and columns near the margins of the image in order to compensate for the reduction of the original image, as a result of the convolutional operation and for the conservation of pixel data at the borders. The rationale for not implementing padding was primarily due to the fact we only utilized a stride of 'one'. This would limit the reduction of the resultant feature maps after the convolution process as presented in the equation. Furthermore, as our aim was to attain high accuracy with a shallow architecture, we felt that the limited convolutional operations mitigate the requirement for padding. Where n_{out} = Nu. of output features, n_{in} = Nu. of input features, p =padding size, k =kernel size, s =stride

$$n_{out} = \left[\frac{n_{in} - 2p - k}{s} \right] + 1$$

The justification for the selection of odd dimensional filters was on the basis that it offers an anchor pixel for encoding the results, post filtering. This cannot be achieved with the use of symmetrical dimensional filters as there is no anchor pixel with respect to the symmetrical nature of the kernel, resulting

in aliasing errors. The activation function was selected as ReLu, due to its effectiveness in preventing the vanishing gradient problem compared to Sigmoid or Tanh, owing to its mathematical function being fundamentally a 'max' operation. The simplicity of the ReLu function also made it more efficient for implementation on GPU enabled processors.

Max-pooling was initiated after each convolutional process to eliminate any positional dependency which may tilt the network towards overfitting whilst in the training phase. Max-pooling takes the highest value from within a feature map. The basis for selecting max-pooling rather than average-pooling was due to our findings through data inspection. It was noted that 'Micro-cracks' within a cell were generally stark and, therefore, could be distinguished from normal PV cells in most cases. Therefore, when removing positional dependency, it was justified to implement max-pooling with the aim to maintain the stark difference between the normal and defective PV cells.

The final component of our architecture was the fully connected layers. Fully connected layer is the intermediary entity residing between the convolutional layers and the output layer. They provide the output layer with access to the amassed image information acquired through the convolution layers facilitating the final classification of the input to be made on a wide range of factors.

The justification for introducing two fully connected layers post convolutional blocks was to enhance the capability of the network to further develop the responses received from the final convolutional block, for the classification process.

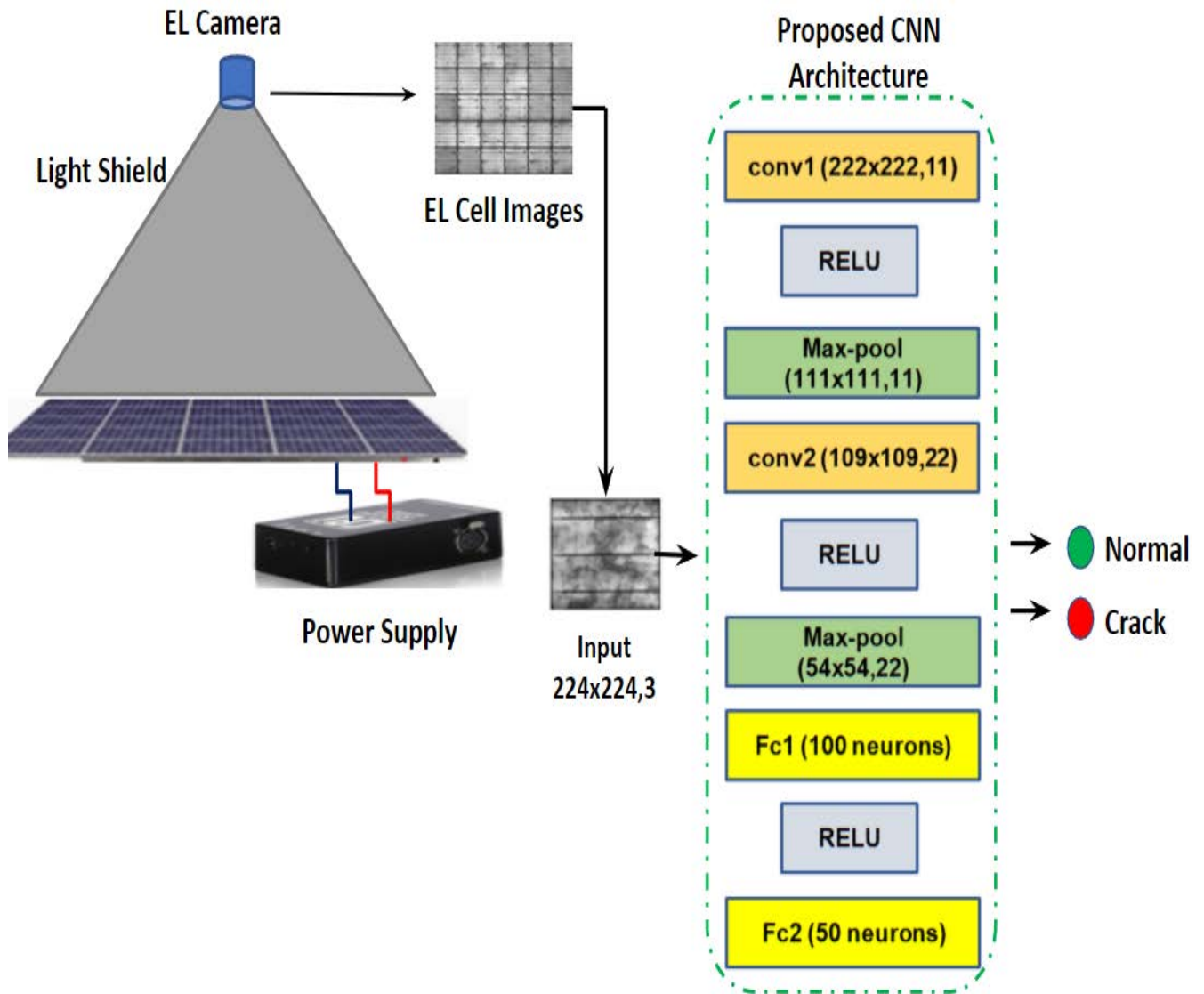


FIGURE 11. Architecture block diagram.

It may be questioned as to why the convolutional layer was not connected directly to the output. Although, this is theoretically feasible, doing so, would suppress the amount of detail available for making the final classification as convolutions are established on segments of the image rather than the whole image itself i.e., local representations. While fully connected layers take the output from the convolutional layer via a fully connected structure providing global representations and maximizing the networks classification ability.

After designing the raw architecture of the network, the next part was the selection of a loss function. There are various loss functions that can be used for regression and classification tasks. The starting point in the selection of the loss function is to know what type of output is expected from the model. We expect our network, given an input image of a PV cell to output the status i.e., normal, or defective.

Rather than using an activation function like ReLU or Tanh at the very last layer of the network for making the predictions, we used a ‘SoftMax’ function. As a result, this would convert the output of the last layer into what is essentially known as probability distribution with the values summing to one. By having two output distributions, one for the one-hot encoded output classes (0-normal,1-defective) and the second for the predicted output distribution we can feed this into the loss function, where p =probability of class, q = class label.

$$H(p, q) = - \sum_x p(x) \log q(x)$$

The selection of ‘cross-entropy’ for the loss function was not only due to its wide use for classification tasks but rather cross-entropy can be used for measuring the difference between two probability distributions (obtained through one-hot encoding and applying of SoftMax at the output

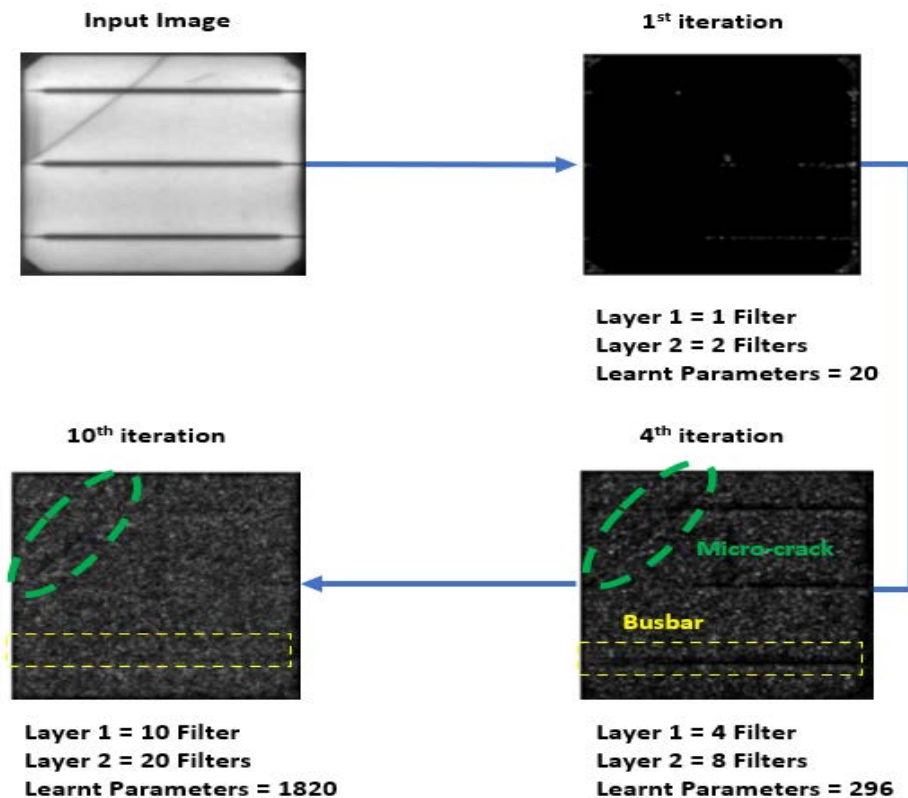


FIGURE 12. GGFT maps for modified architecture.

layer). The cross-entropy between the actual distribution and the predicted distribution is a scalar measure of the difference between the two. This is exactly what’s required for a cost function to be initialized with the aim to get the predicted distribution close to the actual distribution. The proposed architecture is shown in Figure 11.

Figure 12 shows a more abstract view of the filter output evolution. We wanted to achieve two specific objectives (Structure Comparator) from this process. The first providing the base capacity for capturing the fundamental structure of the PV cell. It is evident from Figure 12 that the first iteration lacked the required capacity for meeting the stated objective. By the time we reached the fourth iteration, the network had achieved the necessary appreciation for the overall structure. However, it can be seen from the fourth iteration that the busbar structure was a strong indicator for impacting the scoring function, with similar starkness to that of the Micro-crack (upper left region of the cell). Therefore, our next objective was to diminish the level of starkness for the busbar configuration. This could be a potential factor for misclassification due to its similarity between certain Micro-crack structures.

By the 10th iteration, we were able to reduce the impact of the busbar on the scoring function; however, the starkness in the Micro-crack had also decreased. This was not a major concern, as mentioned earlier, the process for the filter design had only involved backpropagating of gradients without any optimization. After the filter design had been approved, the

weight updation would be enabled, allowing the filter weights to be fine-tuned with respect to the scoring function.

F. FILTER FUSED DATA SCALING

The impetus that led to Filter Fused Data Scaling (FFDS) proposal was two-fold. Firstly, data scarcity within the PV industry was a major factor. This is further aggregated when the data required is after performing EL testing of PV cells within the manufacturing complex. Secondly and most importantly, the development of the Gradient Guided Filter Tuning (GGFT) mechanism for determining the number of filters provided a major breakthrough. That is, when projecting the gradient outputs for various filter configurations, we found that specific configurations projected an output that was similar to the actual input image with a level of distortion that was similar to what may be found within EL samples due to variations in manufacturing processes, EL filter specifications, shielding etc. Hence, by modifying the GGFT process, filter configurations that resulted in practically feasible samples could be used as augmentations, enabling representative scaling of the dataset.

G. MODIFYING GGFT FLOW

The repurposed GGFT process flow is presented in Figure 13. The process was initiated with a sample from the original dataset D_n , passed through the CNN architecture, with the values for f_1, f_2, f_{c1} and f_{c2} being configurable. The gradients

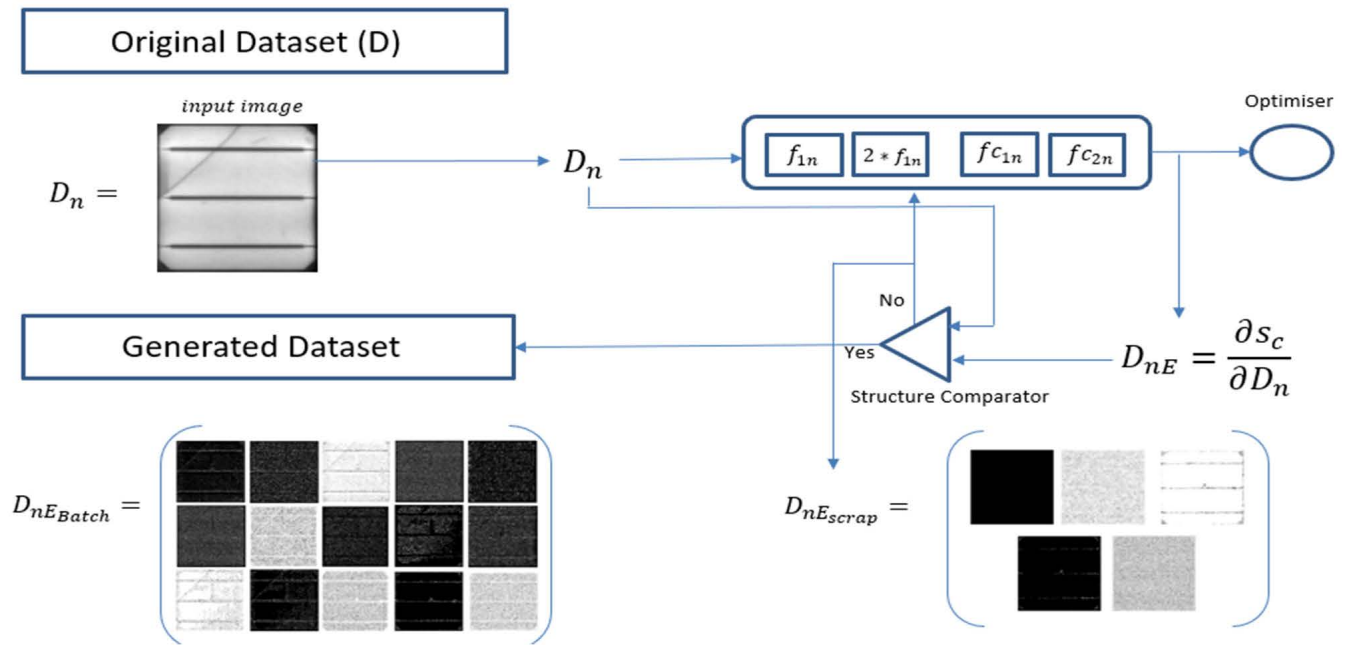


FIGURE 13. FFDS process flow.

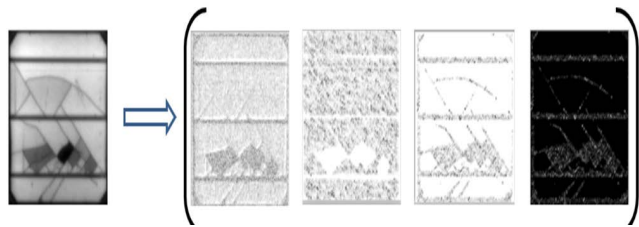


FIGURE 14. FFDS accepted batch sample set.

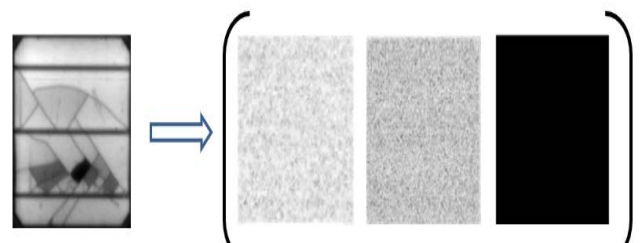


FIGURE 15. FFDS rejected batch sample set.

with respect to the scoring class S_c are extracted without passing through the optimizer, backpropagating onto the input image. The fact that the gradients linked to the scoring class are only backpropagated allows the resultant image to bring forth pixel-based regions that had the most impact on the scoring class S_c . The resulting image D_{nE} and the original image D_1 are passed through the structure comparator. The role of the structure comparator is different from that of GGFT. Here, the pass criteria's objective is whether D_{nE} is likely to be produced through the EL process due to production floor variations.

Generated images D_{nE} that are passed through the structure comparator as containing practical variations with respect to the domain are included in the augmented dataset $D_{nE_{Batch}}$, as shown in Figure 14.

FFDS generated images that were labelled by the structure comparator as incapable of manifesting any aspect of the actual EL obtained cell images were designated as not useable $D_{nE_{Scrap}}$, as shown in Figure 15. As evident from the generated images due to the configurations for the FFDS parameters,

these particular cases were not sufficient in capturing any useful representation of the EL processed image.

Figures 14-15, manifest the importance of empirically tuning the configurable parameters (f_1, f_2, f_{c1} and f_{c2}) for obtaining the relevant variations. The proposed concept (FFDS) provides a suitable mechanism for addressing the issue of acquiring large amounts of EL data and transforming the raw dataset to include variances that may be found with PV manufacturing facilities located in different countries.

III. RESULTS

A. HYPER-PARAMETERS

To evaluate the performance of each technique, we decided to implement K-Fold cross-validation, with $K=5$ for each technique. The dataset was split into 5 folds, each fold containing training and validation set for both normal and crack class. The training class contained 376 samples of 'crack' cells and 246 samples of 'normal' cells. Similarly, the validation set contained 94 samples of 'crack' cells and 61 samples of

TABLE 3. Hyperparameters.

Batch Size	32
Epochs	95
Learning Rate	0.02-0.001
Loss	Cross Entropy
Optimizer	SGD-M

TABLE 4. Network performance comparison FFDS.

FFDS	
Precision	98.8%
Recall	99.2%
F1 score	98.8%
Random Erasing	
Precision	96.6%
Recall	94.6%
F1 Score	95.4%

‘normal’ cells for each fold. The hyperparameters used for training our proposed architecture are provided in Table 3.

B. ARCHITECTURAL COMPLEXITY

Firstly, the effectiveness of our network architecture design via GGFT can be gauged by observing the performance of the architecture across all metrics. To show the effectiveness of the FFDS mechanism, we compare the architecture trained on a standard augmentation technique Random Erasing and FFDS.

It can be observed that FFDS outperformed in all three metrics. The results highlight the importance of selective and domain-specific augmentations. Random Erasing was selected as an ‘off the shelf’ augmentation that was justified with respect to our domain, albeit to a certain degree. The justification was that EL images taken at different manufacturing facilities might have a degree of occlusion in the final image due to variations in EL setup, production line configurations, shading etc. However, when observing Random Erasing output augmentations from Figure 16, we observe that the augmentation places a random black square on top of the cell surface. Looking at the original image and applying domain logic, we know that such periodic, stark square placed randomly does not relate to the practical variations found within manufacturing facilities.

Conversely, augmentations extracted from tuning the configurable parameters within the proposed FFDS process introduced augmented images of PV cells related to variations found with PV manufacturing facilities. As a result, the ability of the architecture to generalize and provide high performance was increased. The ability of the FFDS to provide highly relevant augmentations is illustrated in Figure 16.

C. SOTA COMPARISON

This section of the research presents the performance of our designed architecture against state-of-the-art models used for image classification. In addition to the traditional performance metrics the performance of the models is further

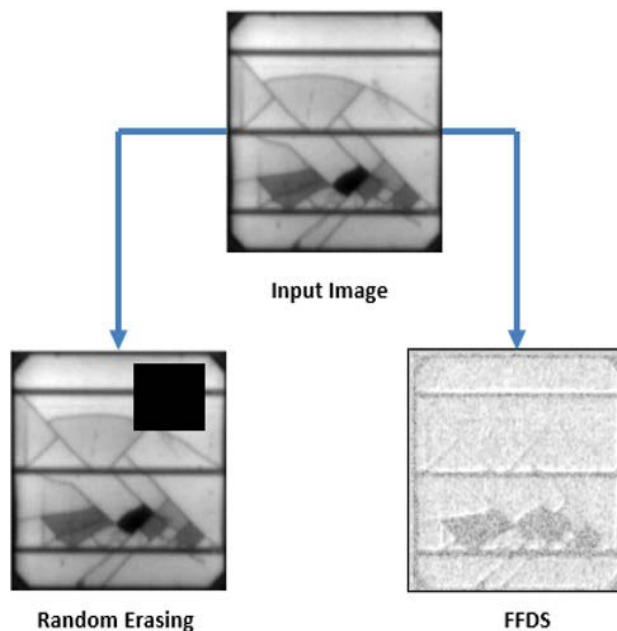


FIGURE 16. Sample comparison (random erasing vs FFDS).

TABLE 5. Performance comparison vs SOTA models.

Model	Precision	Recall	F1
	%	%	%
Our Architecture	98.8	99.2	98.8
ResNet-18	98.6	97.8	97.8
AlexNet	61.0	100	76.0
GoogleNet	99.4	97.8	98.4
MobileNetv2	98	100	98.8

benchmarked based on broader metrics; computational complexity (GMAC’s), number of learnable parameters, Frames Per Second (FPS) and latency. The metrics selected are based on the overall theme of the research i.e., simplicity of network architecture and high computational efficacy.

Table 5 presents the performance of each architecture based on precision, recall and F1-score. It can be observed that all models performed highly across each metric except for AlexNet. Critical analysis reveals that while our architecture in general offered impressive results it was not the top performer across all metrics. However, before passing any conclusions, it is essential to circle back to the purpose of the architecture. The architecture is intended to be deployed within a PV manufacturing factory for detecting Micro-cracks in PV cells. Therefore, we intentionally selected cross validation over standard accuracy for evaluating model performance. The justification behind this was that it would permit us to further tune our network for a specific metric from precision, recall and F1-score to suit our application. Expanding on this further, the architecture post deployment, would maximise the number of true positives (Micro-cracks),

TABLE 6. Architectural performance.

Model	GMAC (G)	Parameters (M)
Our Architecture	0.05	6.42
ResNet-18	1.82	11.69
AlexNet	0.72	61.1
GoogleNet	1.51	13
MobileNetv2	2.88	7.98

TABLE 7. Post-deployment performance comparison.

Model	FPS (s)	Latency (s)
Our Architecture	2.37	0.42
ReSNet-18	0.41	2.42
AlexNet	1.18	0.84
GoogleNet	0.27	3.71
MobileNetv2	0.53	1.91

due to this, there may be some normal cells that are wrongly designated as containing Micro-cracks, however this would not be detrimental as compared to classifying a faulty PV cell as normal and validating it for deployment. Hence, the metric we are most interested in is 'Recall' as this would maximise the number of true positives. Circling back to the results, and focusing on the recall metric, we observe that our model gave a recall rate of 99.2%, a difference of 0.8% from the optimal performance.

To appreciate the effectiveness of our model and its ability to perform highly compared to other SOTA models, we present the computational complexity of the evaluated architectures in Table 6. Multiply-accumulate operations (GMAC's) is one of the metrics used for measuring the model's speed based on the number of computations involved within the network. This can also be measured via Floating Point Operations Per Second (FLOPS). The rationale for selecting multiply-accumulate is that most computations inside a neural network are dot products. This is an important parameter for model evaluation as it provides insights into the feasibility of edge deployment of the model. From Table 6, we observe that our model was the top performer by a high margin, with AlexNet coming in second place.

Furthermore, addressing the lack of literature concentrating on deployment performance, we focused on capturing two vital performance markers whilst the network sought to predict the class of the images in the test batch, latency, and Frames per second (FPS).

We observe our model inference speed was by far the highest from among the models, validating our hypothesis of creating a shallow but at the same time highly performing network. The latency for performing inference on a test image was recorded at 0.42 seconds, also the highest performance across all evaluated models, shown in Table 7.

D. PERFORMANCE EVALUATION

Summarizing the overall evaluation process, it can be said with a high degree of confidence, that our developed

architecture via GGFT and scaled via FFDS provided a well-rounded architecture, highly performant across a broad range of metrics.

The developed architecture secured top position for the architectural, computational and post deployment metrics whilst achieving an impressive recall rate of 99.2%. The results reiterate the effectiveness of the proposed GGFT process for providing a framework that guided the level of complexity required in determining the filters within the convolutional blocks. By tuning the number of filters within each convolutional block as well as the number of internal layers, we were able to effectively suppress the architectural complexity of the proposed system. Furthermore, the FFDS process, designed specifically for generating PV cell samples, and controlled through the defining of the configurable parameters, enabled the model to highly generalize during the various training stages.

It can be argued that the number of learnable parameters are only relevant during the training phase and frozen at an optimum stage for deployment, hence presenting it as a comparison metric is baseless. We endorse the fact that the learnable parameters don't have an impact post-deployment due to the freezing of weights, however as we are providing an architectural comparison coupled with the post deployment metrics, this metric shows the effectiveness of our GGFT mechanism for keeping a check on the complexity of the architecture.

Also, the architectural complexity has a proportional effect on the training time of the architecture. This doesn't impact post deployment, however when we look at the bigger picture it does come into effect due to the concept of data drift. As with Machine learning, computer vision networks are also affected by data distributions, distortions due to external factors; environmental or productional. Therefore, when these changes affect the type of data that is being introduced to the architecture for inference, the architectures need to be trained again on the new 'representative data', hence bringing the architectural complexity back into the fray.

For example, after deploying our architecture within a production facility, if radical production level changes occur, that produce EL images significantly different from the original dataset, then re-training of the architecture would be required for tuning the network in accordance with the drifted data. Depending on the complexity of the network this may not be possible on standard hardware specifications. Taking VGG-19 as a test case model, if re-training was required, this would take weeks on a CPU device due to its significant number of learnable parameters (143.67 Million) as compared to training our architecture within a single day.

IV. CONCLUSION

In conclusion, we were successful in developing a lightweight CNN architecture for the detection of Micro-cracks within a PV Manufacturing complex. The impetus of our research was derived from the inspection of the dataset and studying the modelling of the EL process for photovoltaic cell.

Our proposed GGFT process allowed us to determine the number of filters within each convolutional block with more stability. Furthermore, whilst reviewing the output samples from the GGFT process, we felt that, in fact, by tuning the configurable parameters we had defined within GGFT, we could acquire new samples that presented real variance found within PV manufacturing facilities. Hence FFDS was introduced as an extension of GGFT.

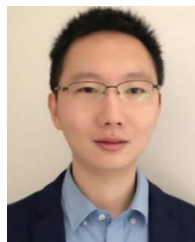
Benchmarking the performance of the FFDS generated images against of-the-shelf augmentations such as Random Erasing, the former outperformed the latter in all metrics. Furthermore, when compared with SOTA architectures on various metrics consisting of computational, architectural and post deployment, our network performed better in the majority of metrics, especially in post-deployment. We are confident that the FFDS process will provide an effective mechanism for PV researchers to address the issue of data scarcity, especially when it comes to EL imaging, enabling developers to create a more robust fault detection network for managing defective PV cells at an early stage of manufacturing. For future work the proposed technique can be employed in sectors utilizing X-ray data such as healthcare sector [19], [20].

REFERENCES

- [1] E. Kabir, P. Kumar, S. Kumar, A. A. Adelodun, and K.-H. Kim, "Solar energy: Potential and future prospects," *Renew. Sustain. Energy Rev.*, vol. 82, pp. 894–900, Feb. 2018.
- [2] M. Dhimsih and P. Mather, "Development of novel solar cell micro crack detection technique," *IEEE Trans. Semicond. Manuf.*, vol. 32, no. 3, pp. 277–285, Aug. 2019.
- [3] M. Dhimish, V. Holmes, B. Mehrdadi, and M. Dales, "The impact of cracks on photovoltaic power performance," *J. Sci., Adv. Mater. Devices*, vol. 2, no. 2, pp. 199–209, Jun. 2017.
- [4] Y. Fu, X. Ma, and H. Zhou, "Automatic detection of multi-crossing crack defects in multi-crystalline solar cells based on machine vision," *Mach. Vis. Appl.*, vol. 32, no. 3, pp. 1–14, May 2021.
- [5] B. Doll, T. Pickel, O. Schreer, C. Zetzmann, J. Hepp, J. Teubner, C. Buerhop, J. A. Hauch, C. Camus, and C. J. Brabec, "High-throughput, outdoor characterization of photovoltaic modules by moving electroluminescence measurements," *Opt. Eng.*, vol. 58, no. 8, p. 1, Aug. 2019.
- [6] M. Dhimish, V. Holmes, M. Dales, and B. Mehrdadi, "Effect of micro cracks on photovoltaic output power: Case study based on real time long term data measurements," *Micro Nano Lett.*, vol. 12, no. 10, pp. 803–807, Oct. 2017.
- [7] M. Frazão, J. A. Silva, K. Lobato, and J. M. Serra, "Electroluminescence of silicon solar cells using a consumer grade digital camera," *Measurement*, vol. 99, no. 10, pp. 7–12, Mar. 2017.
- [8] M. W. Akram, G. Li, Y. Jin, X. Chen, C. Zhu, and A. Ahmad, "Automatic detection of photovoltaic module defects in infrared images with isolated and develop-model transfer deep learning," *Sol. Energy*, vol. 198, pp. 175–186, Mar. 2020.
- [9] S. Deitsch, V. Christlein, S. Berger, C. Buerhop-Lutz, A. Maier, F. Gallwitz, and C. Riess, "Automatic classification of defective photovoltaic module cells in electroluminescence images," 2018, *arXiv:1807.02894*.
- [10] A. Ahmad, Y. Jin, C. Zhu, I. Javed, A. Maqsood, and M. W. Akram, "Photovoltaic cell defect classification using convolutional neural network and support vector machine," *IET Renew. Power Gener.*, vol. 14, no. 14, pp. 2693–2702, Oct. 2020.
- [11] W. Tang, Q. Yang, K. Xiong, and W. Yan, "Deep learning based automatic defect identification of photovoltaic module using electroluminescence images," *Sol. Energy*, vol. 201, pp. 453–460, May 2020.
- [12] C. Dunderdale, W. Brettenny, C. Clohessy, and E. E. Dyk, "Photovoltaic defect classification through thermal infrared imaging using a machine learning approach," *Prog. Photovoltaics, Res. Appl.*, vol. 28, no. 3, pp. 177–188, Mar. 2020.
- [13] N. L. Tun, A. Gavrilov, N. M. Tun, D. M. Trieu, and H. Aung, "Remote sensing data classification using a hybrid pre-trained VGG16 CNN- SVM classifier," in *Proc. IEEE Conf. Russian Young Researchers Electr. Electron. Eng. (ElConRus)*, Jan. 2021, pp. 2171–2175.
- [14] J. Zhang and Y. Feng, "Advanced Chinese character detection for natural scene based on EAST," *J. Phys., Conf.*, vol. 1550, no. 3, May 2020, Art. no. 032050.
- [15] Z. Y. Khan and Z. Niu, "CNN with depthwise separable convolutions and combined kernels for rating prediction," *Expert Syst. Appl.*, vol. 170, May 2021, Art. no. 114528.
- [16] R. Pierdicca, E. S. Malinverni, F. Piccinini, M. Paolanti, A. Felicetti, and P. Zingaretti, "Deep convolutional neural network for automatic detection of damaged photovoltaic cells," *Int. Arch. Photogramm., Remote Sens. Spatial Inf. Sci.*, vol. 42, no. 2, pp. 1–8, 2018.
- [17] P. Napoletano, F. Piccoli, and R. Schettini, "Anomaly detection in nanofibrous materials by CNN-based self-similarity," *Sensors*, vol. 18, no. 2, p. 209, Jan. 2018.
- [18] K. Simonyan, A. Vedaldi, and A. Zisserman, "Deep inside convolutional networks: Visualising image classification models and saliency maps," *Vis. Geom. Group, Univ. Oxford, Oxford, U.K., Tech. Rep.*, 2013.
- [19] T. Chen, C. Shang, P. Su, E. Keravnou-Papailiou, Y. Zhao, G. Antoniou, and Q. Shen, "A decision tree-initialised neuro-fuzzy approach for clinical decision support," *Artif. Intell. Med.*, vol. 111, Jan. 2021, Art. no. 101986.
- [20] T. Chen, E. Keravnou-Papailiou, and G. Antoniou, "Medical analytics for healthcare intelligence-Recent advances and future directions," *Artif. Intell. Med.*, vol. 112, pp. 1–5, Feb. 2021.



MUHAMMAD HUSSAIN was born in Dewsbury, West Yorkshire, U.K., in 1995. He received the B.Eng. degree in electrical and electronic engineering and the M.S. degree in the Internet of Things from the University of Huddersfield, Charlottesville, in 2019, where he is currently pursuing the Ph.D. degree in artificial intelligence for defect identification. His research interest includes detection of various faults in particular micro-cracks forming on the surface of photovoltaic (PV) cells because of mechanical and thermal stress. He has a particular interest in the field of machine vision, focusing on the development of light-weight architectures that can be optimized for deployment on edge devices and ultimately on the production floor. He is also researching into design-level architectural interpretability, with a focus on explainable AI for sensitive fields, such as medicine and healthcare.



TIANHUA CHEN received the Ph.D. degree in computer science from Aberystwyth University, Aberystwyth, U.K., in 2017. He is currently a Senior Lecturer in artificial intelligence with the Department of Computer Science, School of Computing and Engineering, University of Huddersfield, Huddersfield, U.K. He has published over 50 peer-reviewed papers in leading international journals and conferences, including a lead-authored paper selected by the IEEE Computational Intelligence Society as one of the two IEEE TRANSACTIONS ON FUZZY SYSTEM Publication Spotlight articles, as introduced in the 2021 May issue of *IEEE Computational Intelligence Magazine*. His research interests include computational intelligence and data analytics with a keen focus on mental health and brain informatics. He is an Editorial Board Member of *Artificial Intelligence in Medicine* journal.



SOFYA TITRENKO received the Ph.D. degree in applied mathematics from Moscow State University (MSU), in 2011. She joined the University of Leeds, Leeds, U.K., in 2011, and worked on the problems of computational geophysics at the School of Earth and Environment. She joined the School of Mathematics, University of Leeds, in 2017, and worked on the development of pattern mining algorithms. Since April 2022, she has been working as a Lecturer in statistics with the

Department of Statistics, Faculty of Engineering and Physical Sciences, University of Leeds. She currently works as a Senior Lecturer in mathematics with the School of Computing and Engineering, University of Huddersfield, Huddersfield, U.K. She is the author or coauthor of 34 journals and conference publications. Her current research interests include mathematical algorithms and their applications to the fields of biology, geoscience, engineering, and medicine.



PAN SU received the M.Sc. degree in computer science and technology from Hebei University, China, in 2011, and the Ph.D. degree in computer science from Aberystwyth University, U.K., in 2015. He is currently an Associate Professor with the School of Control and Computer Engineering, North China Electric Power University, China. His research interests include medical image analysis, fuzzy systems, and machine learning.



MUFTI MAHMUD (Senior Member, IEEE) received the Ph.D. degree in information engineering from the University of Padova, Italy, in 2011. He is currently working as an Associate Professor of cognitive computing at Nottingham Trent University (NTU), U.K. He has been listed among the top 2% scientists in the world. He is the Co-ordinator of the Computer Science and Informatics Research Excellence Framework Unit of Assessment, NTU, and the Deputy Group Leader

of the Cognitive Computing & Brain Informatics and the Interactive Systems Research Groups. With over 18 years of experience in the industry and academia in India, Bangladesh, Italy, Belgium, and U.K., he is an expert in computational intelligence, applied data analysis, and big data technologies with a keen focus on healthcare applications. As of February 2022, he has published over 190 peer-reviewed articles and papers in leading journals and conferences and (co-)edited six volumes and many journal special issues on those domains. He has secured research grants totaling > £3.3 million and has supervised over 50 researchers and students (postdoctorals, Ph.D., master's, and bachelor's). He is a Senior Member of ACM, a Professional Member of the British Computer Society, and a fellow of the Higher Education Academy, U.K. He was a recipient of the NTU VC Outstanding Research Award 2021 and the Marie-Curie Postdoctoral Fellowship. During the year 2021–2022, he has been serving as the Vice-Chair for the Intelligent System Application and Brain Informatics Technical Committees of the IEEE Computational Intelligence Society (CIS), a member of the IEEE CIS Task Force on Intelligence Systems for Health, an Advisor of the IEEE R8 Humanitarian Activities Subcommittee, the Publications Chair of the IEEE U.K., and Ireland Industry Applications Chapter, and the Project Liaison Officer of the IEEE U.K., and Ireland SIGHT Committee. He has also served as the Co-ordinating Chair for the local organization of the IEEE-WCCI2020; the General Chair of Brain Informatics Conference, in 2020, 2021, 2022, and the Applied Intelligence and Informatics Conference, in 2021 and 2022; and the Symposium Chair of IEEE-CICARE 2017, 2018, 2019, 2020, 2021, and 2022. He serves as a Section Editor (Big Data Analytics) for the *Cognitive Computation* journal, an Associate Editor for the *Frontiers in Neuroscience*, and a Regional Editor (Europe) for the *Brain Informatics* journal. He also serves as the Lead Editor-in-Chief for *Applied Intelligence and Informatics* (Springer) and *Smart Healthcare Systems: From Data to Knowledge* (Routledge) book series.

...