sustainability

Article

Performance Analysis of Classification and Detection for PV Panel Motion Blur Images Based on Deblurring and Deep Learning Techniques

Abdullah Ahmed Al-Dulaimi [1], Muhammet Tahir Guneser [1], Alaa Ali Hameed [2], Fausto Pedro García Márquez [3], Norma Latif Fitriyani [4] and Muhammad Syafrudin [5,*]

[1] Department of Electrical and Electronics Engineering, Karabuk University, Karabuk 78050, Turkey
[2] Department of Computer Engineering, Istinye University, Istanbul 34396, Turkey
[3] Ingenium Research Group, University of Castilla-La Mancha, 13071 Ciudad Real, Spain
[4] Department of Data Science, Sejong University, Seoul 05006, Republic of Korea
[5] Department of Artificial Intelligence, Sejong University, Seoul 05006, Republic of Korea
* Correspondence: udin@sejong.ac.kr; Tel.: +82-(34)-081879

Abstract: Detecting snow-covered solar panels is crucial as it allows us to remove snow using heating techniques more efficiently and restores the photovoltaic system to proper operation. This paper presents classification and detection performance analyses for snow-covered solar panel images. The classification analysis consists of two cases, and the detection analysis consists of one case based on three backbones. In this study, five deep learning models, namely visual geometry group-16 (VGG-16), VGG-19, residual neural network-18 (RESNET-18), RESNET-50, and RESNET-101, are used to classify solar panel images. The models are trained, validated, and tested under different conditions. The first case of classification is performed on the original dataset without preprocessing. In the second case, extreme climate conditions are simulated by generating motion noise; furthermore, the dataset is replicated using the upsampling technique to handle the unbalancing issue. For the detection case, a region-based convolutional neural network (RCNN) detector is used to detect the three categories of solar panels, which are all_snow, no_snow, and partial. The dataset of these categories is taken from the second case in the classification approach. Finally, we proposed a blind image deblurring algorithm (BIDA) that can be a preprocessing step before the CNN (BIDA-CNN) model. The accuracy of the models was compared and verified; the accuracy results show that the proposed CNN-based blind image deblurring algorithm (BIDA-CNN) outperformed other models evaluated in this study.

Keywords: deep learning; CNN; image classification; solar panels; photovoltaic (PV); PV image detection

check for updates

Citation: Al-Dulaimi, A.A.; Guneser, M.T.; Hameed, A.A.; Márquez, F.P.G.; Fitriyani, N.L.; Syafrudin, M. Performance Analysis of Classification and Detection for PV Panel Motion Blur Images Based on Deblurring and Deep Learning Techniques. *Sustainability* **2023**, *15*, 1150. https://doi.org/10.3390/su15021150

Academic Editor: Mahyar Silakhori

Received: 29 November 2022
Revised: 19 December 2022
Accepted: 3 January 2023
Published: 7 January 2023

1. Introduction

In recent years, deep learning (DL) and other forms of artificial intelligence (AI) have emerged as mainstays in numerous fields of study, including image classification and recognition. Image classification and recognition are the subjects of many applications in different fields [1], where they make up a sizable field with lots of room for further research and debate [2]. With deep learning, we can obtain greater classification potential by first extracting the relevant features from the fed feature spaces. The final decision is made in the final layer (fully connected) using the output from the convolutional neural network (CNN) model used to extract features and classify the images. The primary goal of feature extraction in images is to streamline the image and classification process by eliminating unnecessary large data and extracting useful features. There are very high similarities in features between classes that cause the models to struggle with identifying them in some image datasets, such as medical X-ray images, thermal images, and animal images of the same breed.

The main challenge is increasing the capabilities of deep learning techniques to distinguish and classify images with high similarity. Deep learning techniques have been developed specifically to deal with the classification of image data. Convolutional neural networks (CNN) have many models, namely visual geometry group-16 (VGG-16), VGG-19 [3], residual neural network-18 (RESNET-18), RESNET-50, and RESNET-101 [4], which we use in this study.

Several studies have demonstrated that various CNN models can be used to solve many real-world issues and applications. Ahsan et al. [5] used six different models of convolutional neural networks to detect COVID-19 using X-ray images. They used two different types of datasets, one balanced and the other unbalanced. The results show that the two models VGG-16 and MobileNetV2 outperformed the rest of the models according to the recognition accuracy. In [6] and [7], the authors used the RESNET model with segmentation on the computed tomography (CT) dataset of COVID-19 patients, and the results verify the performance behavior by giving high accuracy. Ishengoma et al. [8], in their study, relied on the image dataset they obtained through an unmanned aerial vehicle (UAV) to detect the maize leaves that were infected by fall armyworms. The study was conducted on deep learning models, namely VGG-16, VGG-19, InceptionV3, and MobileNetV2; furthermore, the researchers added additional images which were produced by using corner Shi-Tomas detection, and the results show that these models have high accuracy in detecting infected leaves. On the other hand, the image optimization process used for the images has significantly and positively increased the efficiency of the models used. Zhu et al. [9] studied the quality of the appearance of the islands based on twelve (12) models of deep learning, where the researchers added a support vector machine (SVM) to the proposed models. Their results show that the addition of SVM significantly increased the accuracy of the deep learning models.

Renewable energy has recently been in high demand due to environmental and economic factors. Solar panels are essential to harness and transform solar energy into usable energy. Any obstructions block sunlight, cause the panels to stop working, and restrict the access of the photons, so solar panels can only be used in appropriate climatic conditions. In areas with low temperatures, snowfall causes a layer of snow to form on solar panels. This results in an insulating layer on the solar panels and the inability to produce electrical energy.

Several studies have been conducted on snow loss. For instance, researchers [10] presented methods for predicting daily snow losses based on intelligent techniques that can help reduce operational risks. The reliance on solar energy has increased significantly recently [11], as solar energy constitutes the most reliant percentage of renewable energies. Solar panels are the only way to convert solar energy into electrical energy. These panels are affected by the climate, as it is known, in the winter season, as lower temperatures and snowfall have a negative impact on these panels, as snow prevents sunlight photons from reaching the surface of the solar panels; thus, electricity production is reduced at the level of the solar panel system used in the place of snowfall. The overall electricity production losses from solar panels in winter are more than 25% [12] and can be more than 90% if the panels are completely covered with snow [13]. In regions that receive a significant amount of snowfall annually, such as Germany, Canada, Turkey, the U.K., and the U.S., system performance is impacted; consequently, the output power is reduced as a result of the snow that has accumulated on the panels [14,15].

Regarding some related works that depend on mathematical calculations; the state estimation and prediction for the photovoltaic system are critical, as it is very important in avoiding losses due to external influences on the system. The researchers in [16] proposed a new method for calculating the efficiency of the solar panel system in case of snowy weather and low levels of insolation. They clarified that using the Bouguer–Lambert Law, the insolation level of a snow-covered solar panel surface can be estimated. The researchers presented in [17] a state estimation study of two types of solar panels (monofacial and bifacial) for severe winter climates. The study calculated the snow losses in winter for the two types of models, the results achieved show that the snow losses for monofacial and bifacial panels are on average 33% and 16%, respectively, for the winter season and 16% and 2%, respectively, for the annual rate.

The biggest threat to the long-term performance and overall reliability of photovoltaic (PV) modules is faults that occur during operating conditions over time, during transportation and installation, or due to challenging extreme climates. Fast and accurate fault detection in PV modules is essential for the efficient and dependable operation of PV systems and provides sufficient lead time for long-term high performance. In order to locate potential trouble spots, checks of operation and maintenance are conducted; methods of operation and maintenance are costly because they take a long time, interfere with power production, and frequently necessitate specialized laboratory equipment. It is important to remember that PV panels are often situated in inaccessible locations, which makes tampering with them extremely risky [18–21].

Several studies have also been conducted on solar farms to monitor PV panels based on thermal imaging [22], deep learning [23], and Google Earth Engine (GEE) [24] techniques. Many researchers have presented models based on deep learning techniques to detect defects or damage in solar panels. In [25], researchers provided two deep learning models, SVM and CNN multiclasses classification, to automatically detect defects in a single image of PV cells. The researchers depended on the electroluminescence (EL) image dataset in their experiment. The results prove that both models perform well in both training and testing. The researchers in [26] have provided a novel feature descriptor (FD) named CPICS-LBP, which can fuse the CPICS-LBP by thresholding each pixel of the image into binary code. The experiment was conducted on the surface of heterogeneous backgrounds such as polycrystalline silicon PV cells. The results show that the heterogeneous background defect classification was enhanced. The other researchers in [27] proposed a new CNN model for solar cell surface defect classification by fusing a random forest classifier with the CNN model. For more informative features, the spatial attention module is used. Furthermore, a spatial attention class activation map (SA-CAM) was designed, the SA-CAM suppressed the background and highlighted the defect area in the foreground. Finally, the authors applied segmentation to convert complex images to simple images to show surface defects. Finally, the researchers [28] proposed a semantic segmentation model to predict the defects of PV module surfaces based on electroluminescence images (EL). The Deeplabv3 model with a RESNET-50 backbone was used in the experiment with many types of defects in PV cells detected. The authors created ground truth for all datasets used and then compared it with the predicted result to show the performance of the model. Their final result shows that the model has a high performance to detect defects in PV cell images.

Previous studies have demonstrated the performance of deep learning models in image classification. Therefore, in this study, we presented a performance analysis of classification and detection approaches with three cases; the first and second cases classify solar panels based on similarity in images, including size, color, and general appearance of images using the CNN model. The third case detects the panels using the region-based convolutional neural networks (RCNN). Five models are also used in the study, namely VGG-16, VGG-19, RESNET-18, RESNET-50, and RESNET-101, for the classification parts; furthermore, these models are then compared with our proposed blind image deblurring algorithm (BIDA) and CNN (BIDA-CNN) model. The strengths and weaknesses of each model are further investigated, as well as the classification performance of each model considering three splitting datasets scenarios: training, validation, and testing datasets. In the classification, we have two cases, as mentioned before. The first case is conducted with data without any preprocessing, as it uses clear images and an unbalanced dataset. In the second case, extreme climate conditions are simulated by generating motion noise. To balance the data, we replicated the partial class samples using the upsampling technique to improve the performance of the experiment. Snow-covered panels from 1% to 99% are classified as partial and 100% as all_snow. Finally, the solar panels are classified into three categories: “no snow”, “partial”, and “all snow”.

The main contributions of this study can be summarized as follows:

- The study examined the climate challenges of snow-covered solar panels and how to overcome them, as well as the urgent need to apply artificial intelligence methods, such as classification and detection, to deal with these challenges.
- We provided comprehensive experiments and analysis on the roles of upsampling images to solve the issue of insufficient availability of data samples and to significantly enhance the performance of the deep learning models. In this study, the upsampling technique is used for solar panel images.
- We proposed a BIDA-CNN model for images of snow-covered solar panel surface detection and classification.
- The BIDA-CNN-based model for the photovoltaic module classification approach is extensively assessed and validated through a series of experiments using the existing state-of-the-art deep-learning-based solutions such as VGG-16, VGG-19, RESNET-18, RESNET-50, and RESNET-101 as the comparative benchmarks. Moreover, the performance of the different models is examined based on different metrics through a comparative study.
- The paper not only validates the methods based on real-world data available in the synthetic dataset but also has great success in conditions of extreme climate that can be simulated by generating motion blur.
- Finally, the RCNN model based on a series network backbone is applied to detect the solar panel modules and recognize whether they are snow-covered or not.

2. Materials and Methods

2.1. Case Study

In this study, we considered Karabuk University, located in Karabuk province west of the Black Sea region in northern Turkey, as a case study. Karabuk Province is one of the snowiest provinces in winter. Karabuk University is located in the state of Karabuk with Geographical coordinates (41.211242°,32.656032°) [29]. In the Karabuk region, summers are warm and clear and winters are very cold, snowy, and partly cloudy. The temperature normally ranges between −1 °C and 29 °C throughout the year, rarely below −8 °C and above 34 °C. With a temperature of 20.8 °C, August is the hottest month of the year. The average temperature in January is 0.1 °C, which is the lowest average of the year, as we see in Figure 1. Karabuk University is highlighted as a snowy area that causes losses in photovoltaic energy. Karabuk University is characterized by the presence of a large number of solar panels on its roofs and the sides of buildings. As we can see in Figure 2. Karabük University makes use of these solar panels in the production of electrical energy.

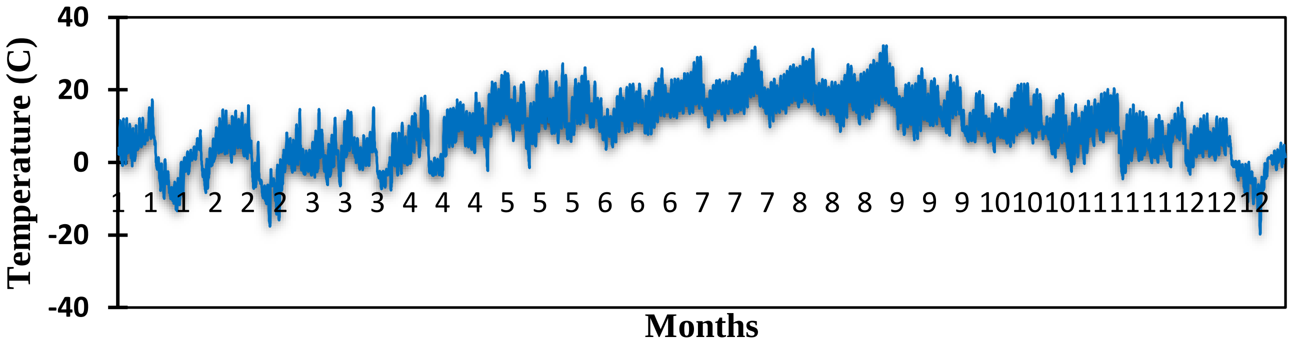


Figure 1. Temperature distribution at 2 m for all months.

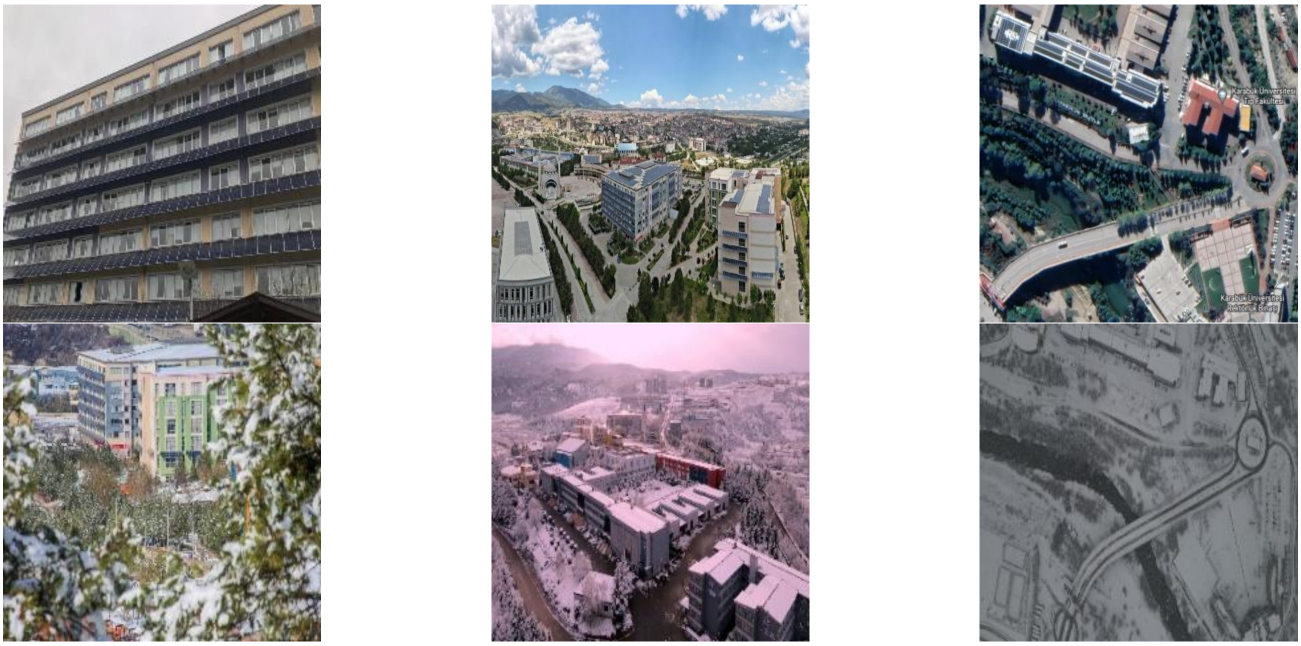


Figure 2. Sample images for Karabuk university solar panels with different side views.

2.2. Irradiation Per Square Meter

Winter has a significant impact on these panels, as low temperatures and snowfall lead to a layer of snow on the surface of the solar panels, as there are sunny days that can be used to produce electrical energy. However, the accumulation of snow that covers the surface of the solar panels reduces the energy production of the panels, making it difficult to utilize the energy on these days. Recently, several methods have been introduced to remove the snow accumulated on solar panel surfaces; one of the recently proposed techniques is the solar panel heating system for photovoltaic systems. Proposing a new method for detecting snow-covered panels is extremely important, since it contributes to accelerating the snow-melting process where detection is handled and then instructs the heating system to carry out the snow-melting process or any other snow-removal process.

The study presents a new methodology based on the historical climatic dataset (2021) that can be found on the NASA research website [30]. Figure 3a,b shows Direct Normal Irradiation (DNI), which is represented by the amount of radiation falling on the surface in an orthogonal manner without calculating the amount of radiation reflected from cloud particles or others; DNI carries the energy that solar panels use to produce electricity. Figure 3a shows the sum of the average amount of energy in units of Wh/m^2 for all months over a 24 h period. The months represented in blue color have lower temperatures and snowfall, while the months represented in black color have moderate temperatures and no snowfall. On the other hand, the months represented in red color mean that the number of sunshine hours is long, which indicates that most solar energy falls on the surface. Snowfall negatively affects the production of electrical energy from solar panels, as snowfall has two main issues: firstly, it prevents the radiation-carrying energy from contacting the solar panels and thus reduces the amount of energy; secondly, it constitutes a layer of snow on the solar panels, causing damage to the solar panels as well as making panels out of service.

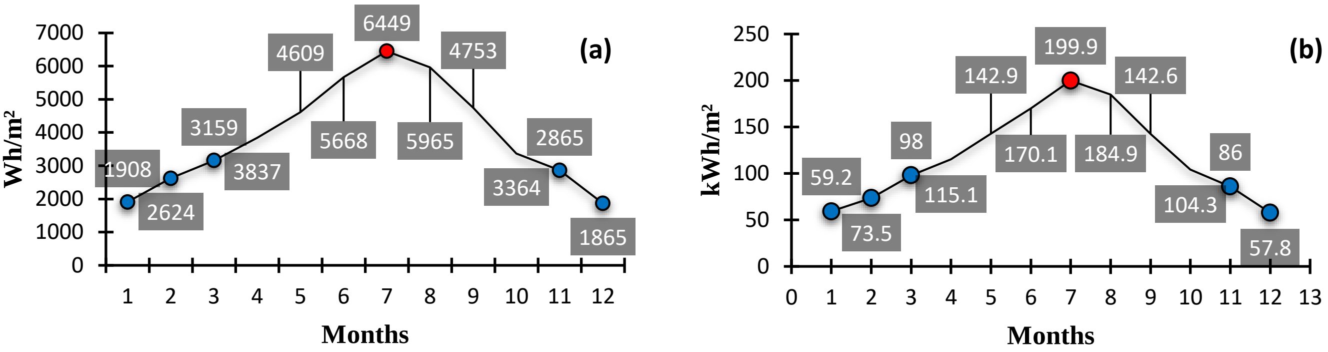


Figure 3. (**a**) The sum of watt hours per meter square for one day. (**b**) Average monthly kilowatt-hours per square meter.

Figure 3a shows the sum of irradiation energy by $\mathrm{Wh/m^2}$ for each month over a 24 h period, as shown in the equation below,

$$\mathrm{Average_{hours}} = \frac{\sum \mathrm{E}}{\mathrm{N}} \tag{1}$$

Figure 3b calculates the irradiation energy in $\mathrm{kWh/m^2}$ for all months as the following:

$$\mathrm{Total\ Average_{month}} = \left(\sum \mathrm{Average_{hours}}\right) \times \mathrm{no.\ month\ days} \tag{2}$$

where E is irradiation energy and N the count of those hours. The $\mathrm{Average_{hours}}$ $\left(\mathrm{Wh/m^2}\right)$ is the E average of the same hour per day throughout the month, while the $\mathrm{Total\ Average_{month}}$ $(\mathrm{kWh/m^2})$ is the total of $\mathrm{Wh/m^2}$ per month.

It is very important to determine the snowfall months to estimate and evaluate the state of the photovoltaic system and forecast the percentage of snow losses to overcome this challenge and solve it using intelligent techniques. Figure 3a,b shows the decrease in the irradiation energy during the first three months and the last two months, which is attributed to a lower temperature and snowfall. Snow losses in these five months are high due to the formation of an insulating layer on the solar panels; the percentage of energy losses is estimated by the time the insulation layer remains on the solar panels. Table 1 shows the specific photovoltaic power output (SPPO) values for the five months based on irradiation energy in relation to the number of sunshine hours.

Table 1. DNI and SPPO rates for the five months that have snowfall.

Months	Rate	kWh/m^2	kWh/kwp
Jan.	4.1%	59.2	56.4
Feb.	5.1%	73.5	70
Mar.	7%	98	93
Nov.	5.9%	86	81.9
Dec.	4%	57.8	55

According to our study, the snowfall days in January are nine days, and the panels need five additional days to melt the remaining snow on the panels. Therefore, the panels will be covered by snow for 14 days. This concludes that the energy produced by the panels during these days is 0%. It has become necessary to find more efficient substitutions using advanced and intelligent technologies that can overcome the obstacles and difficulties facing photovoltaic energy systems, which can contribute to reducing snow losses and improving energy production. This research contributes to the advancement of an advance by directing artificial intelligence techniques to solve the problems of energy systems.

2.3. Dataset

In this experiment, the conducted datasets were examined in two different cases, in the first case, the dataset was conducted on the original size with 395 solar panel images in its original size, then in the second and third cases, the dataset was preprocessed by applying upsampling on the minority sampled classes to be 437 images. This dataset is divided into three categories: all_snow, which represents the images of the panels completely covered by snow; the next category, no_snow, which presents the images of the snow-free panels; and the last category is partial, which presents the partial condensation of snow on the panels. Before the data training process, some preprocessing was performed on the input feature space by resizing the images to 224 × 224 for a regular training process; then, the reconstructed images were fed into the training process. The dataset is divided into 60% training, 20% validation, and 20% testing. Table 2 shows the classification and detection dataset distribution for training, validation, and testing. Figure 4 illustrates samples of the conducted datasets in this paper.

Table 2. Dataset distribution.

Approaches		**Datasets**	**Original Dataset**	**Upsampling Dataset**	**Motion Blur Dataset**
Classification	First case	Training	✓		
		Validation	✓		
		Testing	✓		
	Second case	Training		✓	
		Validation			✓
		Testing			✓
Detection	Third case	Training		✓	
		Validation			✓
		Testing			✓

(**a**) (**b**) (**c**)

Figure 4. Representative samples of solar panel datasets: (**a**) all snow, (**b**) no snow, and (**c**) partial.

2.4. The Models and Features Extraction

The CNN architecture includes three basic layers through which important features are extracted and the classification process is carried out: convolution layers, pooling layers, and fully connected layers [31–33]. The convolution layer is a major component of the CNN structure; it is an input-receiving layer applying two different operations, kernel or filter, and Relu Function. By applying these two currencies, the feature map can be extracted. The pooling layer is usually an appendage with a convolution layer, the purpose of adding a pooling layer is to extract the most important features, thereby reducing data size, as well as accelerating the learning process; there are two types of pooling layers that are the most common: Average Pooling and Max Pooling. The fully connected layers make the last decision, where the features map is extracted in the previous two layers and converted to a one-dimensional array through Flatten Layer (global average pooling) before entering the fully connected layers.

The output value was calculated for each of the convolution layers as described in the Equation below:

$$\mathrm{Y}_{\mathrm{conv,\ 1,2,3,\ldots}} = \left(\sigma\left(\sum_{\mathrm{i}} \mathrm{X_i} \times \mathrm{W_i}\right) + \mathrm{b}\right) \tag{3}$$

where $\mathrm{Y}_{\mathrm{conv,\ 1,2,3,\ldots}}$ is the output, σ mean for each kernel scan, X is the input, W is the filter weight matrix, and b is the bias matrix. The Relu Function is briefly described as follows:

$$\mathrm{f(input\ (x))} = \begin{cases} \mathrm{x}, & \mathrm{x} > 0 \\ 0, & \mathrm{x} \leq 0 \end{cases} \tag{4}$$

Normalization is an essential technique of our analysis, it is a technique used to standardize data; in other words, our network may experience issues, making it significantly more difficult to train it and slowing down its learning rate if the data are not normalized during training. The RESNET model depends on batch normalization, while our proposed method depends on cross-channel normalization. The Equations below describe the batch normalization:

$$\mathrm{x_{normaliz}} = \left(\frac{\mathrm{x^*} - \mu}{\sigma}\right)\gamma + \beta,\ \mathrm{a} = \int(\mathrm{x_{normaliz}}) \tag{5}$$

where $\mathrm{x_{normaliz}}$ is the neurons point value, $\mathrm{x^*}$= (input (x) × weight (w)), $\mu = 0$, $\sigma = 1$, γ, β are batch norm learning parameters, and a is the output.

The Equation for cross-channel normalization can be described as follows:

$$\mathrm{b_i} = \mathrm{a_i}\left(\mathrm{k} + \alpha \sum_{\mathrm{j=j_{bow}}}^{\mathrm{j_{ligh}}} \mathrm{a_j^2}\right)^{-\beta},\ \begin{cases} \mathrm{j_{high}} & = \min\left(\mathrm{i} + \frac{\mathrm{r}}{2}, \mathrm{F_n} - 1\right) \\ \mathrm{j_{low}} & = \max\left(0, \mathrm{i} - \frac{\mathrm{r}}{2}\right) \end{cases} \tag{6}$$

where $\mathrm{b_i}$ is the output of normalization, $\mathrm{a_i}$ activation neurons point value, α, β, k are the hyperparameters of cross-channel normalization, in which $\alpha = 0.0001$, $\beta = 0.75$, and $\mathrm{k} = 1$, and r, $\mathrm{F_n}$ are the depth radius and feature map number. The following Equation is used to calculate the max-pooling layer:

$$\mathrm{Y_{max\ pooling}} = \max(\mathrm{Y_{conv\ ,1,2,3\ldots\ldots}}) \tag{7}$$

The skip connection of the RESNET model can be defined as

$$\mathrm{Y} = \mathrm{F}(\mathrm{X}, \{\mathrm{W_i}\}) + \mathrm{x} \tag{8}$$

where X is the connection to the layers, and x is the skip connection.

Finally, the softmax function is defined as the function that converts a vector of real K values into a vector of real values K whose sum is 1. There are three possible values for the input values: zero, negative, and positive, however, softmax converts them to values between 0 and 1, and as a consequence, they can be considered probabilities. The following Equation can be used to describe this situation:

$$\mathrm{f_j(z)} = \frac{\mathrm{e^{Z_j}}}{\sum_{\mathrm{k}} \mathrm{e^{Z_k}}} \tag{9}$$

$$\mathrm{z_j} = \left(\mathrm{w_j^{\top}} \times \mathrm{Inputs}\right) + \mathrm{Bais} \tag{10}$$

$$w_j^{\top} = \begin{pmatrix} w_{11} & w_{12} & w_{13} & w_{14} \cdots \\ w_{21} & w_{22} & w_{23} & w_{24} \cdots \\ w_{31} & w_{32} & w_{33} & w_{34} \cdots \\ & \vdots & & \end{pmatrix} \tag{11}$$

The visual geometry group (VGG-16, VGG-19) begins with input activations 224 × 224 × 3 for height, width, and depth. For the output activation with 1 × 1 × 1000, in this study, VGG-16 consists of 41 layers, and VGG-19 consists of 47 layers. As a result, both VGG-16 and VGG-19 use only a 3 × 3 convolutional layer with padding, followed by the Relu Function and a padded 2 × 2 pooling layer throughout the network. There are two fully connected layers, the first with 4096 nodes and the second with 1000 nodes, and these layers are preceded by 50% dropout followed by a softmax activation layer that serves as the classifier and probabilities. A visual architectural representation of the VGG-16 and VGG-19 models can be seen in Figures 5a and 5b, respectively.

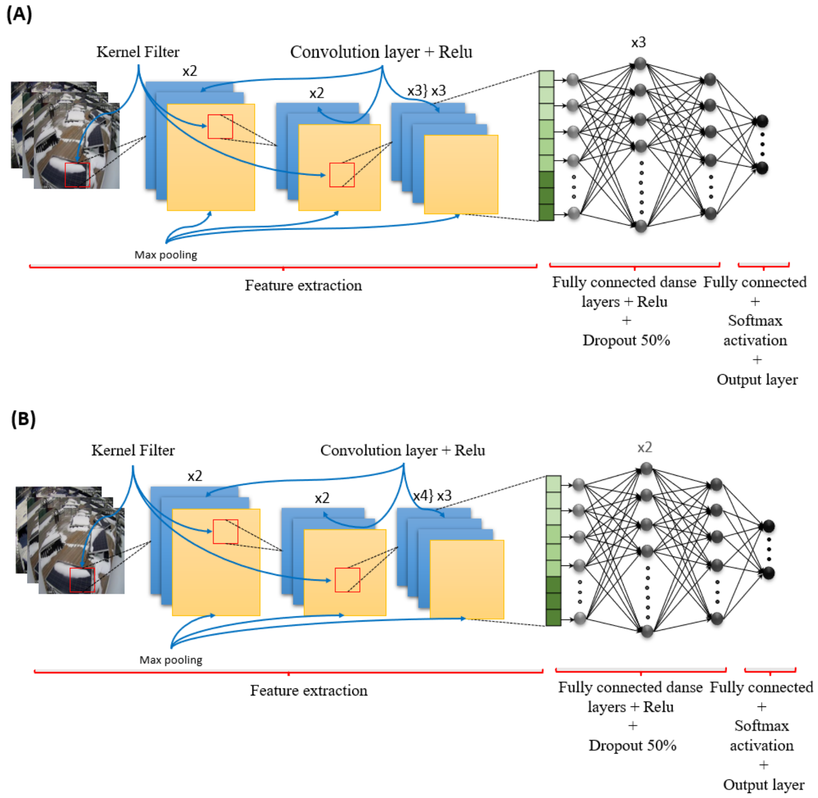


Figure 5. A visual architecture representation of (**A**) visual geometry group-16 (VGG-16) and (**B**) VGG-19.

Deep neural networks have an advantage over shallow neural networks, as they can learn complex functions faster than their shallow counterparts. As deep neural networks are trained, the performance of the model degrades (extracted features) as the depth of the architecture increases. The degradation problem is the term used to describe this issue. To overcome it, skip connections (also known as shortcut connections) are used; as the name implies, some of the CNN layers are skipped and the output of one layer is used as the input to the next layer. The skip connections are used to solve different problems (degradation problems) in different architectures such as RESNET-18, RESNET-50, and RESNET-101, as we present in Figure 6a, Figure 6b, and Figure 6c, respectively.

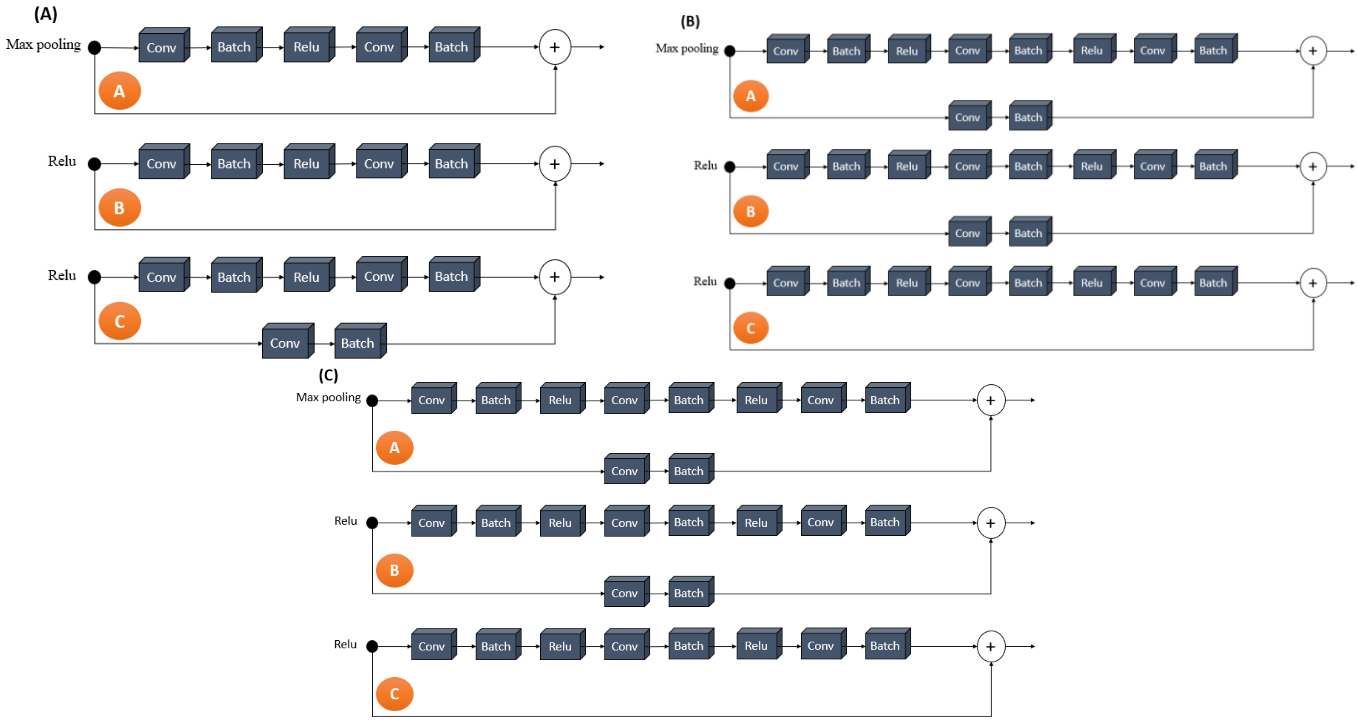


Figure 6. The visual architectural representation of skip connection for (**A**) residual neural network-18 (RESNET-18), (**B**) RESNET-50, and (**C**) RESNET-101.

The residual neural networks (RESNET) that we used in this study, namely RESNET-18, RESNET-50, and RESNET-101, begin with input activations 224 × 224 × 3 for height, width, and depth. For the output activation with 1 × 1 × 1000. In this study, RESNET-18 consists of 71 layers, RESNET-50 consists of 177 layers, and RESNET-101 consists of 347 layers. Consequently, the convolutions of RESNET-18, RESNET-50, and RESNET-101 are preceded by the 7 × 7 inputs images with padding 3 and stride 2, followed by batch normalization of 64 channels and the Relu Function. The pooling layer size is 2 × 2, and padding 1 and stride 2 are used after the first convolution layer. The fully connected layer contains 1000 nodes, preceded by Relu and global average pooling, followed by a softmax activation layer that serves as a classifier and probabilities. A visual architectural representation of these RESNET models RESNET-18, RESNET-50, and RESNET-101 can be seen in Figure 7a, Figure 7b, and Figure 7c, respectively.

In this study, we proposed a new method called BIDA-CNN based on the blind image deblurring algorithm (BIDA) and CNN approach to classifying solar panels images. Originally, BIDA was proposed by Jiangxin Dong et al. [34] as an effective algorithm for blind image deblurring that has outliers. BIDA can also be used to reduce the effects of outliers on the estimation of the blur kernel. The challenge problem in digital imaging is how to restore a clear image without motion blur from an image that has a motion blur due to camera shake; our analysis will rely on BIDA to remove motion blur from the solar panel images. The visual architectural representation of our proposed BIDA-CNN method can be seen in Figure 8.

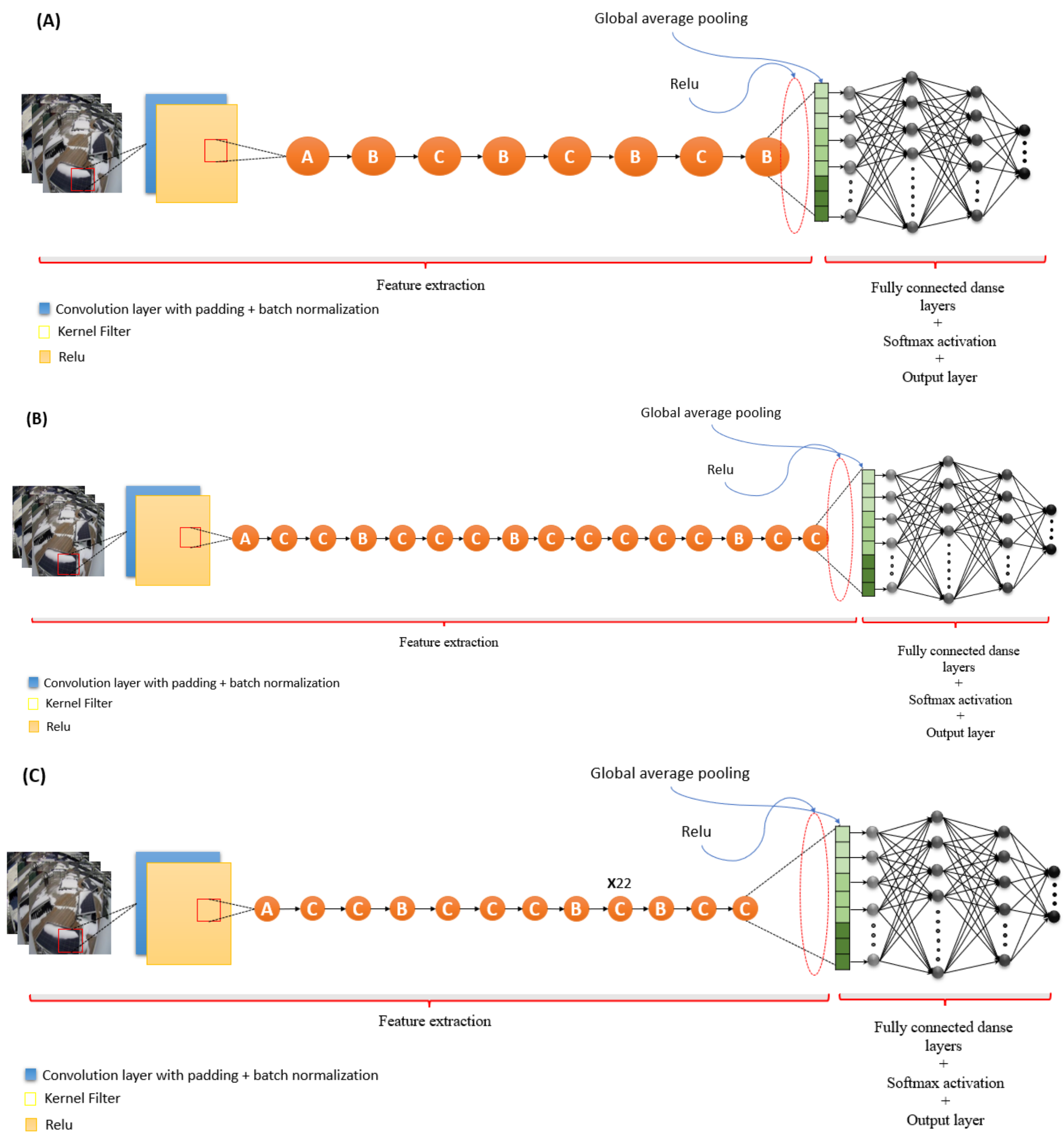


Figure 7. A visual architectural representation of (**A**) residual neural network-18 (RESNET-18) with 71 layers, (**B**) RESNET-50 with 177 layers, and (**C**) RESNET-101 with 347 layers.

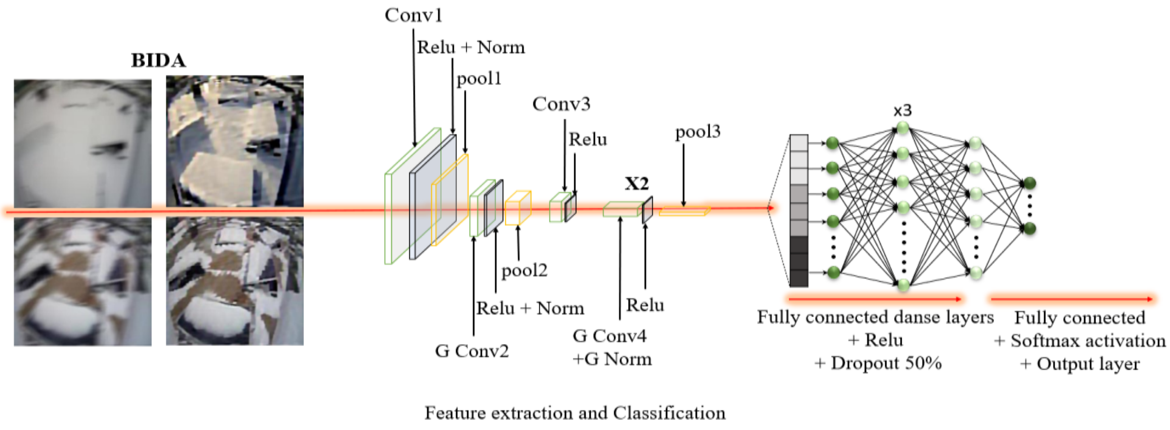


Figure 8. A visual architectural representation of our proposed blind image deblurring algorithm (BIDA) and CNN (BIDA-CNN).

In this study, we simulated the challenges surveillance cameras face while detecting solar panel conditions. To simulate that, we applied motion blur on the original dataset with linear motion across 21 pixels at an angle of 11 degrees. We convoluted between the motion blur kernel to model the motion blur on a digital image. The motion blur can be expressed as the following Equation:

$$\mathrm{y} = \mathrm{x} \ast \mathrm{k} + \mathrm{n} \tag{12}$$

where y is the blurred image, $(\mathrm{x} \ast \mathrm{k})$ convolution operator between the clear image and blur kernel, and n is the noise.

After the original image is exposed to motion blur, the original image becomes a latent image with the blur matrix. BIDA handling preprocesses the used aim to estimate the latent image (x) and blur kernel (k) from the blurred image (z).

The latent image (x) and the blur kernel (k) estimated can be describe by the Equations (13) and (14) below, respectively.

$$\min_{\mathrm{x}} \mathrm{R}(\mathrm{x} \ast \mathrm{k} - \mathrm{y}) + \lambda \parallel \nabla \mathrm{x} \parallel_{\mathrm{x}\ast}, \ \mathrm{x}\ast = 0.8 \tag{13}$$

$$\min_{\mathrm{k}} \mathrm{R}(\nabla \mathrm{x} \ast \mathrm{k} - \nabla \mathrm{y}) + \gamma \parallel \mathrm{k} \parallel_{\mathrm{k}\ast}, \ \mathrm{k}\ast = 1 \tag{14}$$

where $\parallel \nabla \mathrm{x} \parallel_{0.8}$ is the hyper-Laplacian for original priors of x, and $\parallel \mathrm{k} \parallel_1$ is the $\ell 1$ norm for original priors of k and, to balance these two priors, there are weights γ and λ. Iteratively reweighted least squares (IRLS) is used to solve the $R(\cdot)$, as we see in the Equations below for x and k, respectively:

for x:

$$\mathrm{x}^{[\mathrm{t}+1]} = \underset{\mathrm{x}}{\mathrm{argmin}} \sum_{\mathrm{p}} \left\{ \omega^{\mathrm{x}} \left| (\mathrm{x} \ast \mathrm{k} - \mathrm{y})_{\mathrm{p}} \right|^2 + \lambda \left(\omega^{\mathrm{x}}_{\mathrm{h}} \left| (\partial_{\mathrm{h}} \mathrm{x})_{\mathrm{p}} \right|^2 + \omega^{\mathrm{x}}_{\mathrm{v}} \left| (\partial_{\mathrm{v}} \mathrm{x})_{\mathrm{p}} \right|^2 \right) \right\} \tag{15}$$

$$\text{where } \omega^{\mathrm{x}} = \frac{\mathrm{R}'\left(\left(\mathrm{x}^{[\mathrm{t}]} \ast \mathrm{k} - \mathrm{y} \right)_{\mathrm{p}} \right)}{\left(\mathrm{x}^{[\mathrm{t}]} \ast \mathrm{k} - \mathrm{y} \right)_{\mathrm{p}}} \tag{16}$$

$$\text{and } \omega^{\mathrm{x}}_{\mathrm{h}} = \left| \left(\partial_{\mathrm{h}} \mathrm{x}^{[\mathrm{t}]} \right)_{\mathrm{p}} \right|^{-1.2}, \omega^{\mathrm{x}}_{\mathrm{v}} = \left| \left(\partial_{\mathrm{v}} \mathrm{x}^{[\mathrm{t}]} \right)_{\mathrm{p}} \right|^{-1.2} \tag{17}$$

for k:

$$\mathrm{k}^{[\mathrm{t}+1]} = \underset{\mathrm{k}}{\mathrm{argmin}} \sum_{\mathrm{p}} \left\{ \omega^{\mathrm{k}}_{\mathrm{h}} \left| (\partial_{\mathrm{h}} \mathrm{x} \ast \mathrm{k} - \partial_{\mathrm{h}} \mathrm{y})_{\mathrm{p}} \right|^2 + \omega^{\mathrm{k}}_{\mathrm{v}} \left| (\partial_{\mathrm{v}} \mathrm{x} \ast \mathrm{k} - \partial_{\mathrm{v}} \mathrm{y})_{\mathrm{p}} \right|^2 + \gamma \omega^{\mathrm{k}} \left| \mathrm{k}_{\mathrm{p}} \right|^2 \right\} \tag{18}$$

$$\text{where } \omega^{\mathrm{k}}_{\mathrm{h}} = \frac{\mathrm{R}'\left(\left(\partial_{\mathrm{h}} \mathrm{x} \ast \mathrm{k}^{[\mathrm{t}]} - \partial_{\mathrm{h}} \mathrm{y} \right)_{\mathrm{p}} \right)}{\left(\partial_{\mathrm{h}} \mathrm{x} \ast \mathrm{k}^{[\mathrm{t}]} - \partial_{\mathrm{h}} \mathrm{y} \right)_{\mathrm{p}}} \tag{19}$$

$$\text{and } \omega^{\mathrm{k}}_{\mathrm{v}} = \frac{\mathrm{R}'\left(\left(\partial_{\mathrm{v}} \mathrm{x} \ast \mathrm{k}^{[\mathrm{t}]} - \partial_{\mathrm{v}} \mathrm{y} \right)_{\mathrm{p}} \right)}{\left(\partial_{\mathrm{v}} \mathrm{x} \ast \mathrm{k}^{[\mathrm{t}]} - \partial_{\mathrm{v}} \mathrm{y} \right)_{\mathrm{p}}} \tag{20}$$

where $\underset{\mathrm{x,k}}{\mathrm{argmin}}$ mean x, k minimizing the reconstruction error $\left| (\mathrm{x} \ast \mathrm{k} - \mathrm{y})_{\mathrm{p}} \right|^2$, $\left| (\partial_{\mathrm{h}} \mathrm{x} \ast \mathrm{k} - \partial_{\mathrm{h}} \mathrm{y})_{\mathrm{p}} \right|^2$ with the prior preferring x and k to be as smooth as possible, $\mathrm{R}'(\cdot)$ is the derivation of $\mathrm{R}(\cdot)$, p is the spatial pixel location, and t is the iteration index.

Figure 8 also shows the proposed convolutional neural network, which consists of 27 layers. The proposed CNN contains few layers compared with other networks implemented in this paper, which is inspired by Alexnet [35]; we also updated its parameters and added group normalization layers. A new CNN with a few layers is proposed to avoid losing features during the process of feature extraction from convolutional neural networks with a high number of layers; therefore, this network can be expressed as a simple network. The proposed CNN in this experiment begins with input activations 227 × 227 × 3 for height, width, and depth. For the output activation with 1 × 1 × 1000,the learnable weight begins with 11 × 11 × 3 × 96 in Conv1 and ends with 1000 × 4096 in the fully connected layers; furthermore, the learnable bias begins with 1 × 1 × 96 and ends with 1000 × 1 in the fully connected layers. The used convolutional different stride begins with [4 4] and ends with [1 1]; furthermore, padding [0 0 0 0], [1 1 1 1] has been used for the convolutional and pooling layers. The fully connected layers are preceded by a 50% dropout, followed by a softmax activation layer that serves as the classifier and probabilities.

2.5. Overall Proposed Model and Experimental Settings

Our overall proposed model is illustrated in Figure 9. The first part of our study is illustrated in Figure 9a, which involved the dataset with the original size. The compared models were applied to images executed with 224 × 224 × 3 and 227 × 227 × 3 activations. The implementation process went through three steps: data collection, data visualization, and data splitting. Figure 9b illustrates the second part of this study, the imbalance classes with minority samples were handled. To perform that, we applied data upsampling on partial class. Then, the samples of the partial class are increased. This can be accomplished through the use of a variety of methods, such as rotating and inverting the images. The upsampling process on the dataset aims to increase the variability and uniformity of the CNN models. This process helped the models acquire more knowledge about the input space. The Upsampling Equation can be expressed as the following:

$$\mathrm{d_u} = \mathrm{d_o} \times 300\% \tag{21}$$

where $\mathrm{d_u}$ is the upsampling dataset, $\mathrm{d_o}$ is the original dataset, and 300% is the percentage increase.

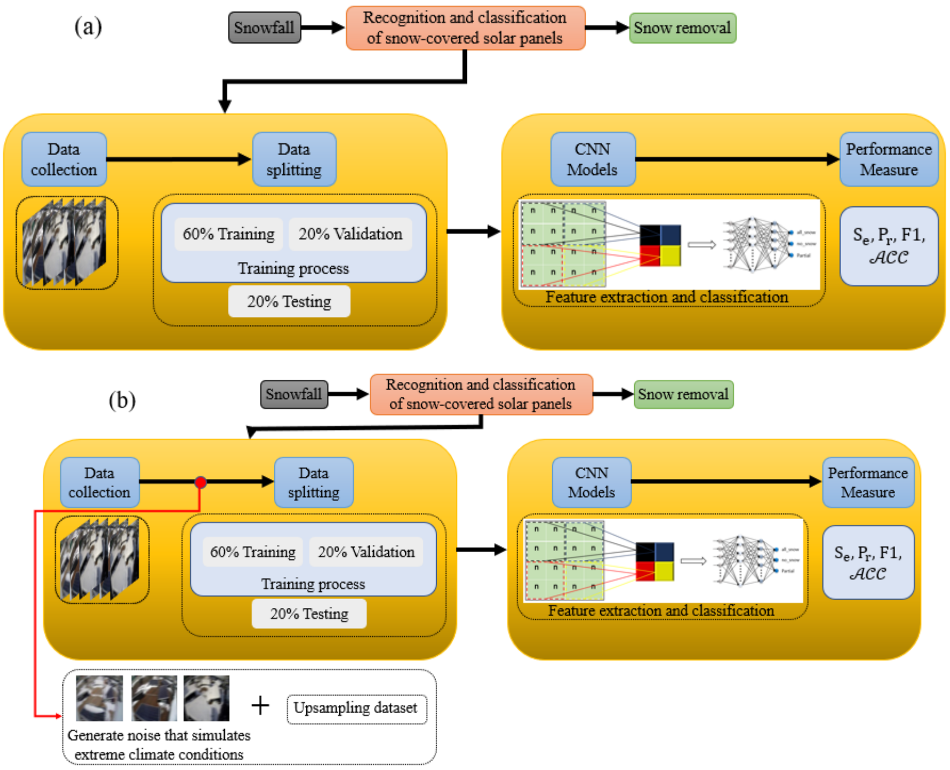


Figure 9. *Cont.*

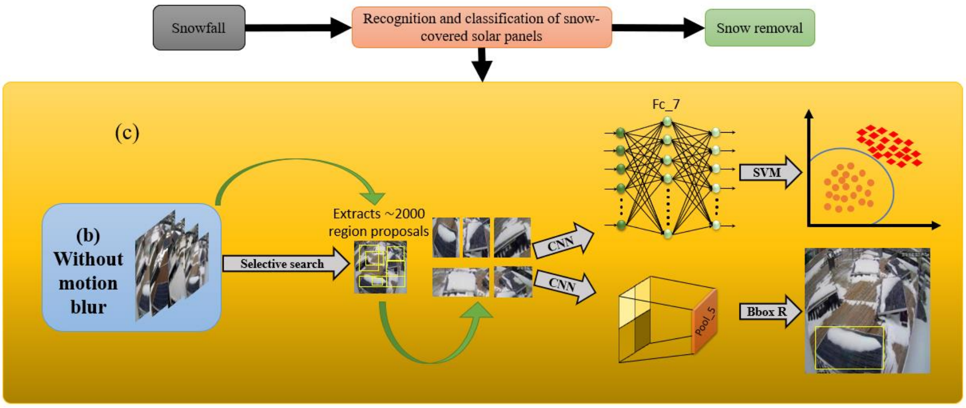


Figure 9. The overall design of our proposed deep-learning-based model for solar panel image classification and detection: (**a**) without preprocessing, (**b**) with simulated motion blur and upsampling preprocessing, and (**c**) RCNN for solar panel image detection.

Finally, training is performed on the bulk of the dataset. Training also includes a validation process, and then testing is conducted to ensure the model's level of accuracy in classification. Figure 9c presents the third part of our study which is the region-based convolutional neural networks (RCNN) [36] that is used to detect the panels.

All the classification and detection experiments are administered in a work environment with an Intel® Core™ i7-9750H CPU @ 2.60GHz and an NVIDIA GeForce RTX 2060 on MATLAB package. RCNN is based on a series network, the ImageNet package provides a variety of pretrained advanced models and is trained on the ImageNet dataset; two of these models are VGG-16 and VGG-19, which were used in this experiment and can accelerate the learning of the network. Furthermore, the above two models are compared with the proposed model. The max epochs are fixed to 80 with the Mini batch size set to 2 due to the GPU memory limitation. The number of iterations per epoch = number of data samples/Mini batch size, which can ensure that the solar panel images training data are looped through completely. The learning rate is set to 0.0001 and the class number is set to 3.

The partial class in the state without upsampling consists of 27 images and with upsampling consists of 80 images; all images in the two states are entered into RCNN. The partial class is important because it contains both cases with snow and without snow, which can contribute to (i) making experiments more complex, and thus giving deep analysis and results, and (ii) to determining the performance of the models used in the detection.

The workflow of RCNN begins with extracting feature maps from input data by a selective search focusing on the region of interest (ROI). Secondly, feature maps are outputted from the first step to the convolution neural network, then the output of the fully connected layer (fc_7) is an input to the support vector machine (SVM). Finally, the ground truth iscompared with the predicted bounding box.

Due to the possibility that the proposed regions by the selective search algorithm may not be able to capture all of the object features, bounding box regression is used to refine the accuracy of the predicted bounding box precision. The ground truth $\left(\mathrm{G} = \left(\mathrm{G_x}, \mathrm{G_y}, \mathrm{G_w}, \mathrm{G_h}\right)\right)$ and the predicted $\mathrm{P} = \left(\left(\mathrm{P_x}, \mathrm{P_y}, \mathrm{P_w}, \mathrm{P_h}\right)\right)$ region are proposed by the selective search algorithm. Parameterizations of the transformation in terms of the four functions are $(\mathrm{dx(P)}, \mathrm{dy(P)}, \mathrm{dw(P)}, \text{ and } \mathrm{dh(P)})$, where x, y are the coordinates of center and w, h are the bounding box width and height.

Transformations are performed to make the prediction of ground truth (G) scale-invariant in the following Equations:

$$\mathrm{T_x} = \frac{(\mathrm{G_x} - \mathrm{P_x})}{\mathrm{P_w}},\ \mathrm{T_x} = \frac{(\mathrm{G_y} - \mathrm{P_y})}{\mathrm{P_h}},\ \mathrm{T_w} = \log\left(\frac{\mathrm{G_w}}{\mathrm{P_w}}\right), \text{ and } \mathrm{T_h} = \log\left(\frac{\mathrm{G_h}}{\mathrm{P_h}}\right) \tag{22}$$

where T_x, T_y, T_w, and T_h are the target for the regression.

The last pooling feature from the CNN (pool5), denoted by φ_5 (P), is fed into the regression model. The four functions of parameterizations are the transformation denoted by $d_\star(P)$:

$$d_\star(P) = w_\star^T \varphi_5\ (P) \tag{23}$$

where ($\star$) is a placeholder for (x, y, w, h).

The learnable model parameters' vector is denoted by ($w_\star$); through optimizing the regularized least squares (RLS) objective (ridge regression), we learn ($w_\star$):

$$w_\star = \underset{\hat{w}_\star}{\operatorname{argmin}} \sum_i^N \left(T_\star^i - \hat{w}_\star^T \varphi_5 \left(P^i\right)\right)^2 + \|\lambda \hat{w}_\star\|^2 \tag{24}$$

When we take the scale-invariant coordinates from the regression model and add them to the coordinates for the proposed regions, we obtain the final set of coordinates.

$\check{G}_x = P_w d_x(P) + P_x$, $\check{G}_y = P_h d_y(P) + P_y$, $\check{G}_w = P_w \exp(d_w(P))$, and

$$\check{G}_h = P_h \exp(d_h(P)) \tag{25}$$

2.6. Evaluation Metrics

In this study, the six different CNN architectures VGG-16), VGG-19, RESNET-18, RESNET-50, RESNET-101, and BIDA-CNN are compared to investigate their performance. Comparisons were conducted using different metrics such as error, accuracy, and convergence behavior. These architectures are characterized by their high ability to understand the patterns and features of the images and the ability to extract them through the structure of the models, thus obtaining appreciated accuracy (ACC):

$$ACC = \frac{T_N + T_P}{T_N + T_P + F_N + F_P} \tag{26}$$

In this context, a model that correctly predicts the positive solar panel classes is commonly known as True Positive (T_P), while a model that correctly predicts the negative solar panel classes is commonly known as a True Negative (T_N). On the other hand, a model that incorrectly predicts the positive solar panel classes is commonly known as a False Positive (F_P), while a model that incorrectly predicts the negative solar panel classes is commonly known as a False Negative (F_N).

The Sensitivity (Recall (S_e)) is defined as the ratio of the overall predicted True Positive (T_P) to the overall predicted True Positive (T_P) and predicted False Negative (F_N). See Equation (7).

$$S_e = \frac{T_P}{T_P + F_N} \tag{27}$$

On the other hand, Precision (P_r) can be defined in a similar manner to sensitivity, except that we consider the predicted False Positive (F_P) in this calculation as the ratio of the overall predicted True Positive (T_P) to the overall predicted True Positive (T_P) and predicted False Positive (F_P) as follows,

$$P_r = \frac{T_P}{T_P + F_P} \tag{28}$$

In addition, $F_1 - \text{score}\ (F_1)$ is the combination of the harmonic mean of precision and recall. This combination comes with a single measure. It provides a more accurate measurement of incorrectly classified cases using the following formula:

$$F_1 = \frac{2 \times S_e \times P_r}{S_e + P_r} \tag{29}$$

Furthermore, these Equations were updated to calculate the overall metrics for each category. The overall sensitivity (Recall (S_e)) is calculated as follows:

$$\text{Overall } S_e = \frac{\sum_{\text{classes}} S_e}{\sum \text{classes}} \tag{30}$$

Overall Precision (P_r) is given as

$$\text{Overall } P_r = \frac{\sum_{\text{classes}} P_r}{\sum \text{classes}} \tag{31}$$

and overall $F_1 - \text{score}\ (F_1)$:

$$\text{Overall } F_1 = \frac{\sum_{\text{classes}} F_1}{\sum \text{classes}} \tag{32}$$

Finally, statistical functions were used, especially measures of central tendency such as median and mode. The value was calculated for each median and mode, respectively, as described in the Equations below:

$$\text{Median}_{\text{odd}} = \left(\frac{n+1}{2}\right)^{\text{th}} \text{observation} \tag{33}$$

$$\text{Median}_{\text{ even}} = \frac{\left(\frac{n}{2}\right)^{\text{th}} + \left(\frac{n}{2}+1\right)^{\text{th}}}{2} \text{observation} \tag{34}$$

$$\text{Mode } = L + h\frac{(f_m - f_1)}{(f_m - f_1) + (f_m - f_2)} \tag{35}$$

where n is the sum of values number, th is the order in which the number is located, and the mode is equal to the most common value.

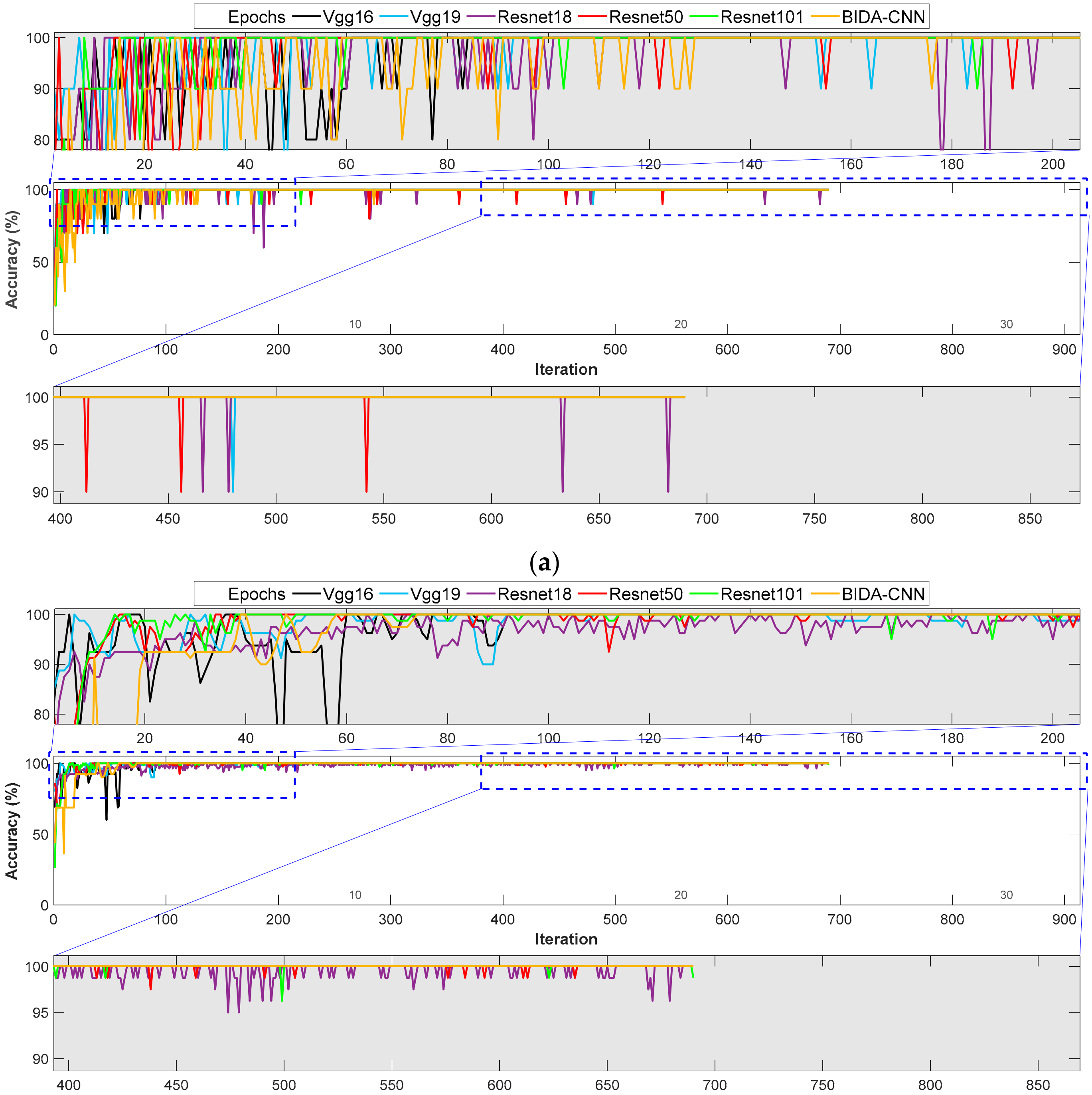
3. Results and Discussions

In this section, the performance of the study is extensively examined and discussed. Due to the diversity of climatic challenges, experiments are designed to simulate surrounding climatic conditions on solar panels depending on Equation (12).

3.1. Classification Performance Results

For the first classification case, we used the original dataset without any type of preprocessing or noise; for the second classification case, we used the motion blur dataset for training with upsampling on partial class; the idea is to use motion blur as training data to indicate whether these datasets are valid for training or not and whether this type of dataset can be relied upon to obtain better test results with extreme climate.

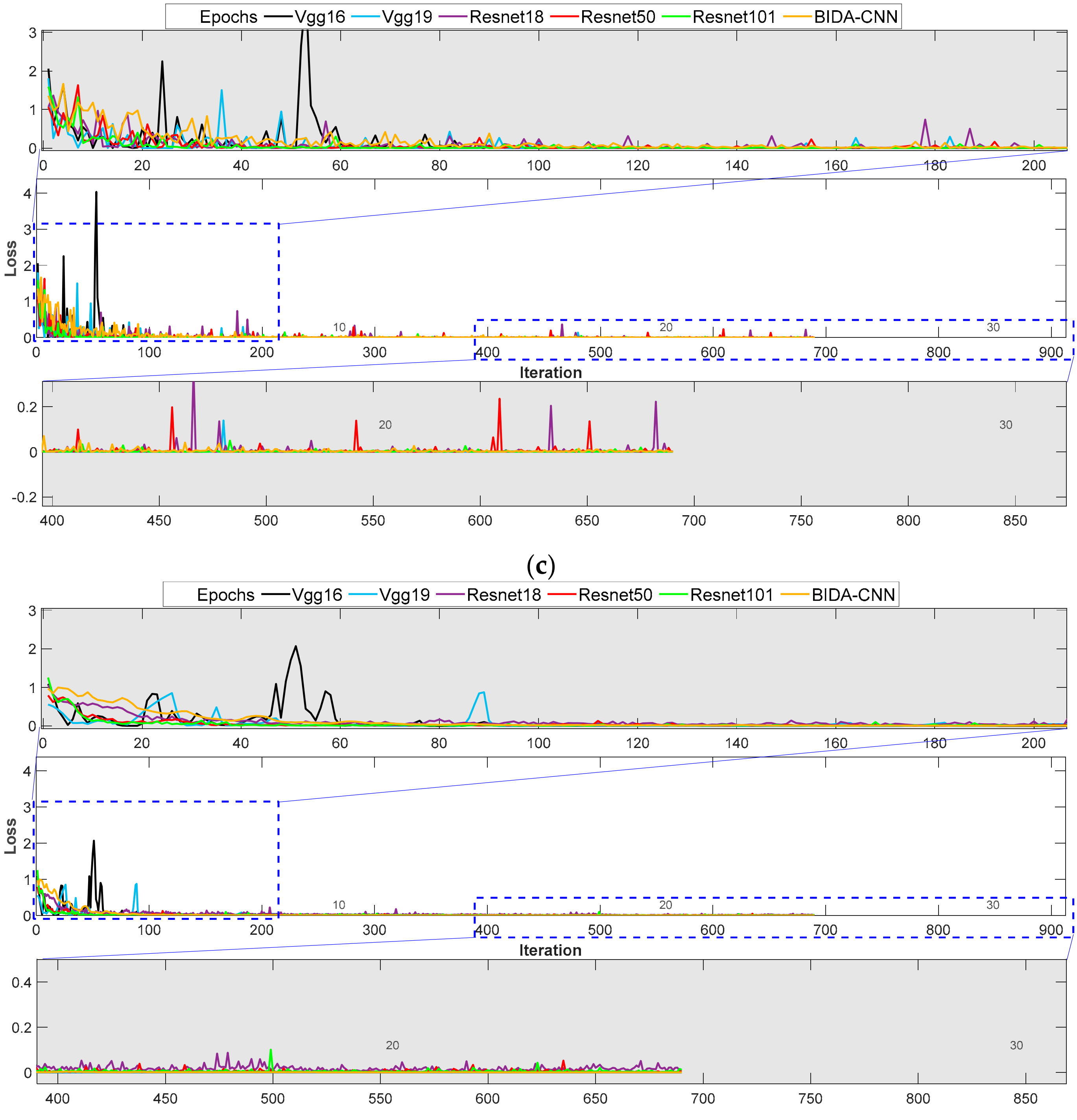
The conducted dataset is divided into 60% of the data utilized for training, while 20% was used for each validation and testing. Figure 10 shows the accuracies and losses of all compared architectures for the first proposed approach in the experimental setup. For the training, Figure 10a,c illustrated the accuracy and loss results of VGG-16, VGG-19, RESNET-18, RESNET-50, and RESNET-101, respectively. The learning behavior of VGG-16 is more stable with less fluctuation than VGG-19: it can reach a steady state before 100 iterations. On the other hand, VGG-19 reaches stability after 183 iterations. The learning behavior of RESNET-101 outperformed RESNET-18 and RESNET-50 architectures by achieving steady-state results with fewer iterations (approximately 60 iterations). The training performance also shows that the BIDA-CNN model with a series network outperforms the other models, which supports the effectiveness of fast learning. The learning behavior of the BIDA-CNN model is nonvolatile and steadier with less fluctuation than RESNET. The performance of the BIDA-CNN model is close to VGG, but it is not possible to rely on VGG because it contains a high number of layers that can be lost in the features of the images in the next cases.



(a)

(b)

Figure 10. *Cont.*



(**c**)

(**d**)

Figure 10. First case comparison results of accuracy and losses for VGG-16, VGG-19, RESNET-18, RESNET-50, RESNET-101, and BIDA-CNN proposed based on (**a**) training accuracy, (**b**) validation accuracy, (**c**) training losses, and (**d**) validation losses.

For the validation, validation is performed after every 1 iteration for all models used in the analysis. The validation process is devoid of underfitting and overfitting. Figure 10b shows the minimum and maximum accuracies for VGG-16 are 60% and 100%, and VGG-19 are 80% and 100%, while the median and mode for VGG-16 are 99.18% and 100% and VGG-19 are 100% and 100%, respectively. The minimum and maximum accuracies for RESNET-18 are 43.75% and 100%, for RESNET-50 68.75% and 100%, and for RESNET-101

26.25% and 100%. In addition, the median and mode for RESNET-18 are 100% and 100%, for RESNET-50 100% and 100%, and for RESNET-101 100% and 100%, respectively. The minimum, maximum, median, and mode for the BIDA-CNN model are 20%, 100%, 100%, and 100%, respectively.

Figure 10d shows that the loss results confirm the accuracy results, where the loss of VGG-16 suffers from a high fluctuation at the beginning of learning. Tables 3 and 4 show the numerical results and the evaluation metrics of all compared architectures for the first proposed approach (original data). The experiment runs with 30 epochs and 690 iterations (for each epoch (E) 27 iterations (I)), and Mini batch accuracy (M^{acc}) results for the last iterations of the last 5 epochs). The best-obtained validation accuracy (V^{acc}) is 100% for all models without RESNET-101. The lowest Mini batch loss (m^{loss}) is 0, obtained by VGG-19, where the lowest loss validation (v^{loss}) obtained from VGG-19 is 1.4901×10^{-8}. The learning rate (LR) for all the architectures is 0.0003. The average of the lowest validation loss for all iterations (v^{loss} mean) is 0.01865, which was achieved by the BIDA-CNN model.

Table 3. Validation comparison results for first case based on the last iterations of the epochs.

Models	**E**	**I**	$\mathbf{M^{acc}}$	$\mathbf{V^{acc}}$	$\mathbf{m^{loss}}$	$\mathbf{v^{loss}}$	$\mathbf{v^{loss}}$ **Mean**	**LR**
VGG-16	26	598	100.00%	100.00%	5.9631×10^{-5}	0.0001	0.03312	0.0003
	27	621	100.00%	100.00%	5.1497×10^{-6}	0.0001		0.0003
	28	644	100.00%	100.00%	1.6808×10^{-6}	8.8603×10^{-5}		0.0003
	29	667	100.00%	100.00%	$\mathbf{1.1682 \times 10^{-6}}$	$\mathbf{8.7807 \times 10^{-5}}$		0.0003
	30	690	100.00%	100.00%	1.4209×10^{-5}	8.8387×10^{-5}		0.0003
VGG-19	26	598	100.00%	100.00%	**0**	$\mathbf{1.4901 \times 10^{-8}}$	0.02139	0.0003
	27	621	100.00%	100.00%	7.6294×10^{-7}	1.4901×10^{-8}		0.0003
	28	644	100.00%	100.00%	7.1526×10^{-8}	1.4901×10^{-8}		0.0003
	29	667	100.00%	100.00%	5.9605×10^{-8}	1.4901×10^{-8}		0.0003
	30	690	100.00%	100.00%	5.4836×10^{-7}	1.4901×10^{-8}		0.0003
RESNET-18	26	598	100.00%	100.00%	**0.0006**	0.0141	0.05048	0.0003
	27	621	100.00%	100.00%	0.0008	**0.0073**		0.0003
	28	644	100.00%	100.00%	0.0009	0.0101		0.0003
	29	667	100.00%	100.00%	0.0007	0.0153		0.0003
	30	690	100.00%	100.00%	0.0014	0.0099		0.0003
RESNET-50	26	598	100.00%	100.00%	0.0007	**0.0038**	0.02047	0.0003
	27	621	100.00%	100.00%	0.0006	0.0067		0.0003
	28	644	100.00%	100.00%	0.0003	0.0100		0.0003
	29	667	100.00%	100.00%	0.0013	0.0049		0.0003
	30	690	100.00%	100.00%	**0.0002**	0.0040		0.0003
RESNET-101	26	598	100.00%	100.00%	**0.0001**	**0.0017**	0.01996	0.0003
	27	621	100.00%	100.00%	0.0003	0.0046		0.0003
	28	644	100.00%	100.00%	0.0011	0.0061		0.0003
	29	667	100.00%	100.00%	0.0010	0.0044		0.0003
	30	690	100.00%	98.75%	0.0002	0.0117		0.0003
Proposed BIDA-CNN	26	598	100.00%	100.00%	0.0003	0.0006	**0.01865**	0.0003
	27	621	100.00%	100.00%	0.0011	0.0005		0.0003
	28	644	100.00%	100.00%	0.0047	0.0005		0.0003
	29	667	100.00%	100.00%	0.0016	0.0005		0.0003
	30	690	100.00%	100.00%	$\mathbf{9.7633 \times 10^{-5}}$	**0.0004**		0.0003

Note: The bold is the minimum losses values, which shows the researchers which model is the best and which model gets the least loss.

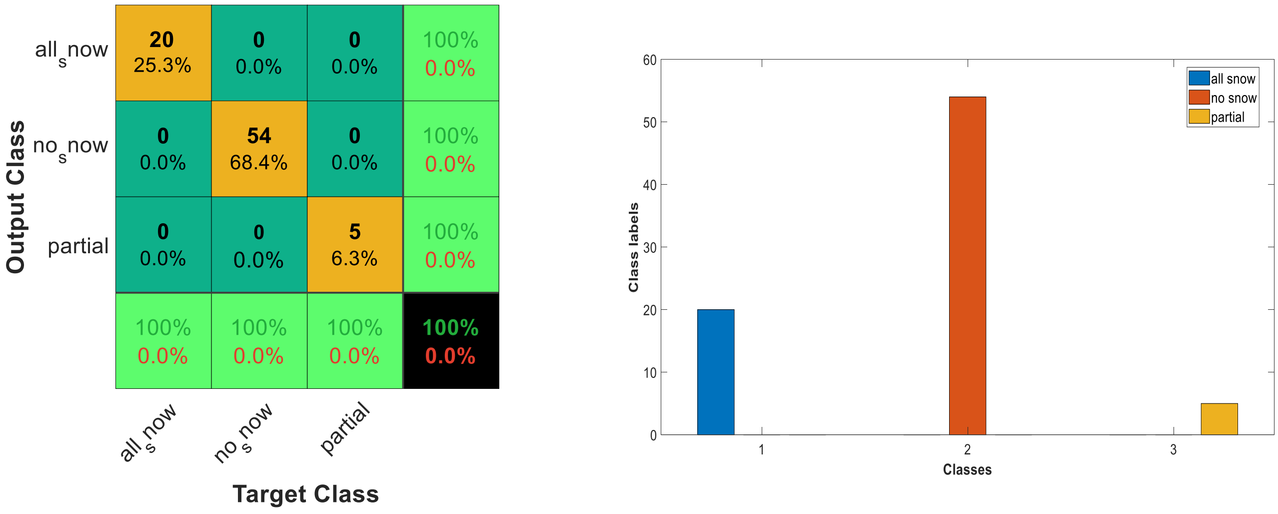
Table 4. Comparison of the testing results based on evaluation metrics for first case.

Models	**Classes**	$\mathbf{S_e}$	$\mathbf{P_r}$	**F1**	**ACC**
All Models	all snow	1	1	1	100%
	no snow	1	1	1	
	partial	1	1	1	

The evaluation metrics and accuracies for the testing used to determine the level of model quality are presented in Table 4. Table 4 shows the sensitivity, precision, F1-score, and final accuracies for testing. The five CNN-based models VGG-16, VGG-19, RESNET-18, RESNET-50, and RESNET-101, as well as the BIDA-CNN model, achieved a testing accuracy of 100% in the first case. We can observe three findings by comparing Table 3 with Figure 10: (i) with the excessive convolutional network layers, the input will lose some important features, affecting the training and testing process; (ii) the skip connection process of RESNET does not always provide the network with informative features that would raise the training accuracy and thus obtain high results; and (iii) this high performance of the models may result from the clean dataset. Thus, motion blur will be applied to simulate extreme climate and complicate the analysis process. In return, the data of the minority class will be increased using upsampling preprocessing to avoid issues resulting in the dataset suffering from inadequate knowledge that can appear due to the complexity of the analysis process after applying the motion blur.

When evaluating the models, it is essential to utilize a variety of different metric evaluations. This is due to the fact that the performance of a model may be satisfactory when using one measurement from one metric of evaluation, but it may be unsatisfactory when using another measurement from another metric of evaluation. It is essential to use evaluation metrics in order to ensure that your model is functioning correctly and to its full potential.

A confusion matrix is a table used to show the performance and effectiveness analysis of the classification model. The evaluation performance of a classification model can be represented graphically and summarized using a confusion matrix. Figure 11a shows that the proposed solar panel classification models are evaluated using confusion-matrix-based performance metrics. The confusion matrix includes actual classes and predicted classes by displaying the values of true positive, true negative, false positive, and false negative. Moreover, through these values, the sensitivity, specificity, precision, and overall accuracy metrics can be calculated. Consequently, the comparison results for our models can be seen in Figure 11a, where the five CNN-based models and BIDA-CNN model metrics achieved high accuracies by providing more accurate diagnostic performance on the solar panels' dataset.



(**a**) (**b**)

Figure 11. (**a**) Confusion matrix results of VGG-16, VGG-19, RESNET-18, RESNET-50, and RESNET-101 and our proposed BIDA-CNN; (**b**) predicted distribution over classes of the compared models, namely VGG-16, VGG-19, RESNET-18, RESNET-50, and RESNET-101, as well as BIDA-CNN.

Figure 11b shows the results obtained by VGG-16, VGG-19, RESNET-18, RESNET-50, and RESNET-101 models, as well as our proposed BIDA-CNN model, where no class missed to classify from the all_snow, no_snow, and partial classes. The RESNET-101 model struggled with the issues in the training process, but there are no labels missed in the testing process (all_snow, no_snow, and partial classes); this is because the probability of classification in the softmax layer was high (approximately higher than 95%), and because of that, no labels were missed in the testing process, as we present in Figure 12, taking into account that the result is compared in the confusion matrix with 20% out of the conducted dataset.

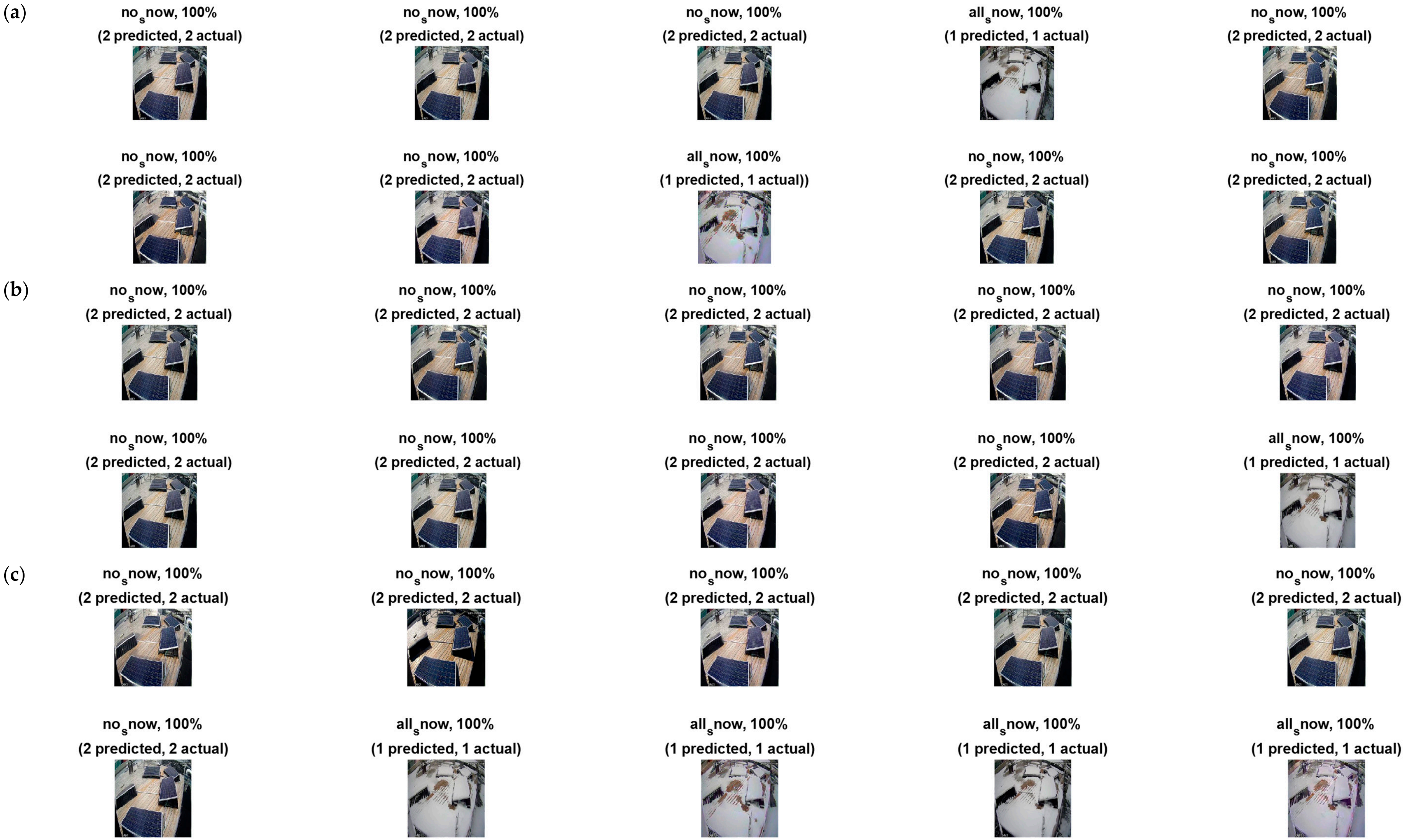


Figure 12. *Cont.*

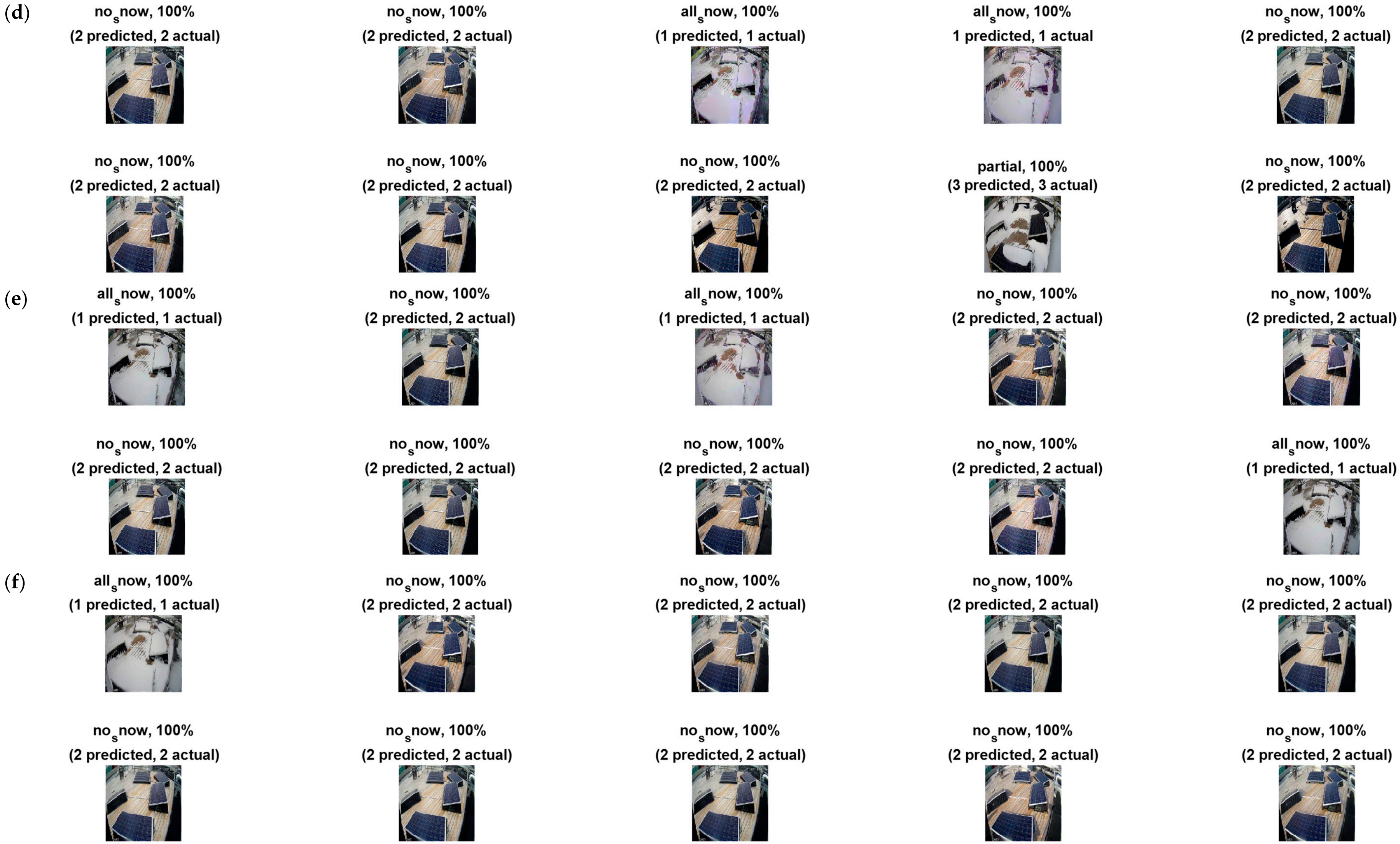


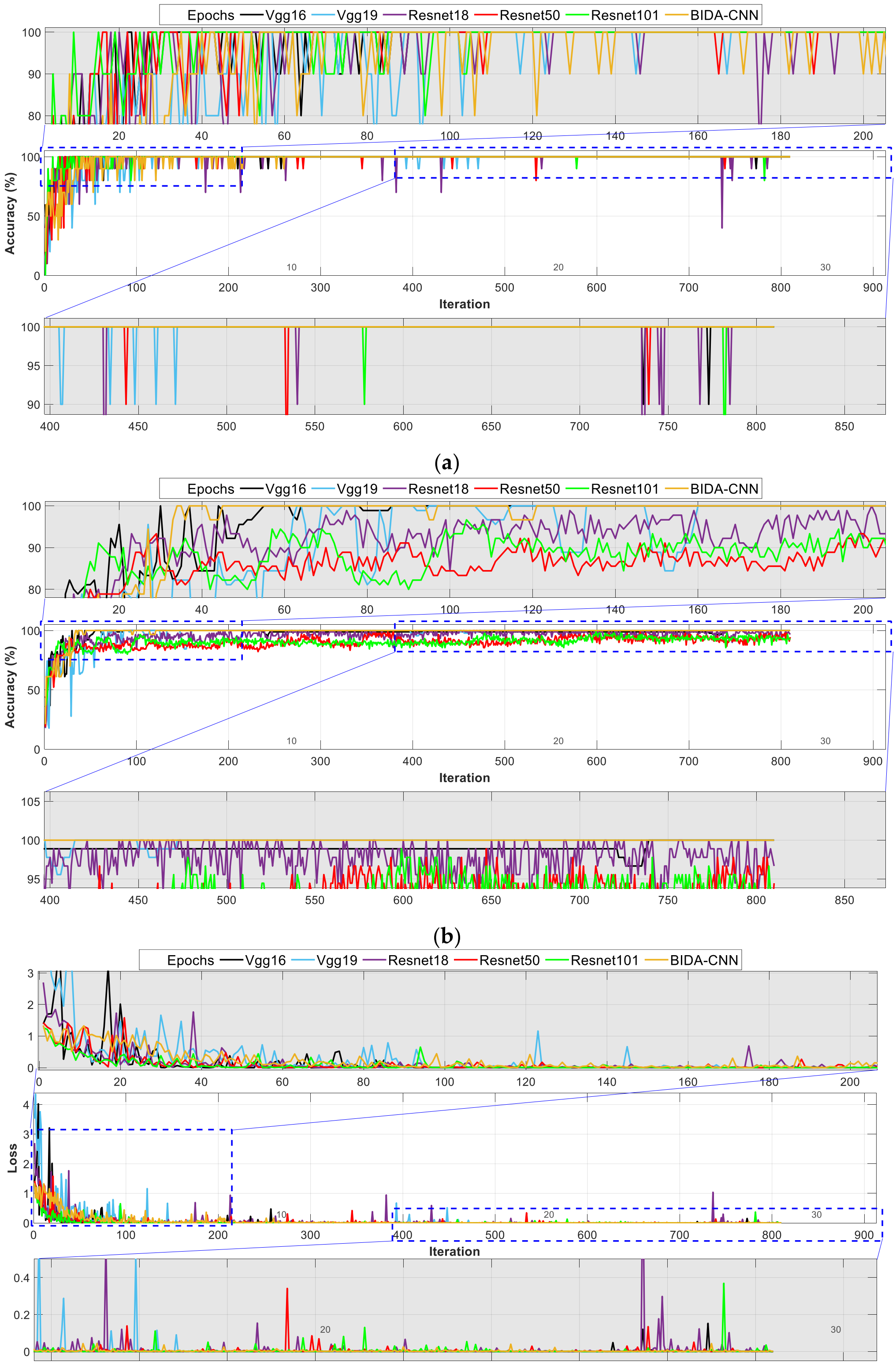
Figure 12. Second case image probabilities with a softmax function for (**a**) VGG-16, (**b**) VGG-19, (**c**) RESNET-18, (**d**) RESNET-50, and (**e**) RESNET-101, as well as (**f**) our proposed BIDA-CNN.

In the second proposed approach, the upsampling method is applied to the imbalance classes with minority samples. The partial class in the conducted dataset suffers from inadequate knowledge that could provide better prediction performance. This can be accomplished by using a variety of techniques, such as rotating and inverting the images. The dataset upsampling process aims to increase the variability and uniformity of the CNN models. This procedure aids the models in learning more about the input space. Moreover, in this approach, we simulate the challenges faced by surveillance cameras while detecting the condition of solar panels to simulate that we applied motion blur on the original dataset with linear motion across 21 pixels at an angle of 11 degrees.

The balance dataset is trained using the five implemented models and the BIDA-CNN model. The accuracy and loss results of VGG-16, VGG-19, RESNET-18, RESNET-50, and RESNET-101 are shown in Figures 13a and 13c, respectively. We can analyze that VGG-16 is more stable with less fluctuation than VGG-19 from a learning behavior perspective, it converges faster and reaches the steady state at an earlier stage of iterations (250 iterations); furthermore, VGG-19 reaches stability after 167 iterations with fluctuation between 390 to 470. The learning behavior of RESNET-101 outperformed RESNET-18 and RESNET-50 architectures by achieving steady-state results with 100 iterations. Our proposed BIDA-CNN model achieved higher fast learning and operation speed than other models. The BIDA-CNN model achieved a steady state with 260 iterations; furthermore, the fluctuations of the BIDA-CNN model were fewer than other models, as observed in Figure 13c.

The other part of the dataset is validated with 20%. The same with the first section of this experiment, the validation is performed after every 1 iteration for all models used in the analysis. As used for the validation process, it does not involve underfitting and overfitting. The obtained accuracies of validation for all models are shown in Figure 13b; the accuracies improve iteratively with time. The minimum and maximum accuracies for VGG-16 are 36.67% and 100%, and for VGG-19 17.78% and 100%, while the median and mode for VGG-16 are 98.89% and 98.89%, and for VGG-19 100% and 100%, respectively. The minimum and maximum accuracies for RESNET-18 are 17.78% and 100%, for RESNET-50 18.89% and 98.89%, and for RESNET-101 35.56% and 98.89%. In addition, the median and mode for RESNET-18 are 96.67% and 98.89%, for RESNET-50 91.11 and 92.22%, and for RESNET-101 91.11% and 93.33%, respectively. Figure 13d shows the loss results confirm the validation accuracy results, where the lowest Mini batch loss $\left(\mathrm{m}^{\mathrm{loss}}\right)$ and the lowest validation loss ($\mathrm{v}^{\mathrm{loss}}$) were obtained from VGG-19. The minimum, maximum, median, and mode for the BIDA-CNN model are 21.11%, 100%, 100%, and 100%, respectively; this indicates that the model outperforms the rest of the models used in this experiment. Furthermore, the BIDA-CNN model outperformed the other five models used in this experiment by an average lowest validation loss ($\mathrm{v}^{\mathrm{loss}}$ mean) of 0.03797.

We can deeply observe three findings by comparing Figures 10 and 13: (i) the motion blur increased the fluctuation in the training process; (ii) the motion blur led to the instability of the validation process accuracy in networks with a high number of layers (RESNET-18, RESNET-50, and RESNET-101), and this led to there being fluctuations until the end of the validation process, however, the fluctuations remained confined between 90% to 100%, which represents the positive probability in the softmax layer, and we observe that all classification categories are correct with a variable probability that always tends to be positive; and (iii) the deblurring model is the ideal solution under extreme climates, which helps to obtain clean and clear data that can be conducted within classification and detection models and achieve high performance results.

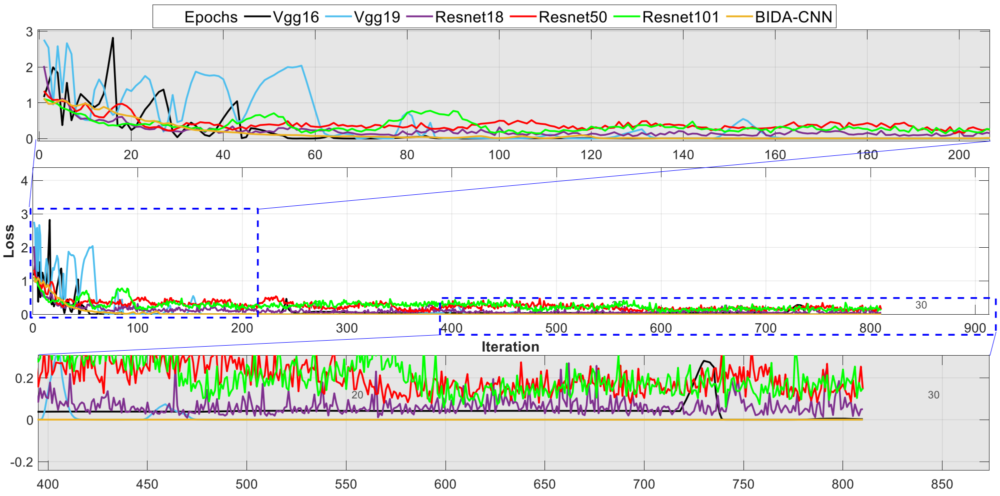


(**a**)

(**b**)

(**c**)

Figure 13. *Cont.*



(**d**)

Figure 13. Second case comparison results of accuracy and losses for VGG-16, VGG-19, RESNET-18, RESNET-50, RESNET-101, and BIDA-CNN proposed based on (**a**) training accuracy, (**b**) validation accuracy, (**c**) training losses, and (**d**) validation losses.

To determine the level of model quality, the evaluations, metrics, and testing accuracies for all models are included in Tables 5 and 6. These are used to determine how well the model is constructed. The quantitative results are well presented in Tables 5 and 6, and the numerical results and the evaluation metrics of all compared architectures for the second proposed approach are clearly discussed. The experiment runs with 30 epochs and 810 iterations (for each epoch (E) 27 iterations (I)), and the Mini batch accuracy (M^{acc}) results for the last 5 epochs are 100%. The best validation accuracy (V^{acc}) is 100%, obtained from the VGG-16, VGG-19, and BIDA-CNN models. The lowest Mini batch loss (m^{loss}) is 3.5763×10^{-8}, obtained by VGG-19, whereas the loss validation (v^{loss}) obtained from VGG-19 is 2.0662×10^{-6}. The learning rate (LR) for all the architectures is 0.0003. Based on v^{loss} mean, the best model in this experiment is the BIDA-CNN model, achieving 0.03797. In addition, all models used in this experiment recorded 100% overall precision, sensitivity, F1-score, and accuracy in the testing. Achieving high accuracy does not mean that the model probability is 100%; the probability ranges from 50% to 99%. We always work on proposing models that increase the probability of being correct through a mathematical model that improves the process of extracting features with important information that is input into the decision layer.

Figure 14 shows that the diagonal matrix represents a true positive that the model has correctly predicted class dataset values; moreover, the values that are biased from the diagonal matrix are false predicted values. In addition, Figure 14a shows there are no objects that were missed in the classes, taking into account that the result is compared in the confusion matrix with 20% out of the conducted dataset. Figure 14b represents the data distribution over the classes; it shows which class gives high-performance prediction and which class struggles with noise and outliers. Each of the tested models achieves different data distribution or classification. In Figure 14b, the classes are presented in the form of a histogram. Each bar contains part of the data (class). Figure 14b shows the predicted classes by all models, where all classes are well-predicted (no missed classification).

In this experiment, the softmax activation function is used in the output hidden layer of tested architectures. The softmax function normalizes the received outputs from the previous layers by translating them from weighted sum values into probabilities that add up to one, then determines the class values. After that, the achieved probability values of the predicted classes from the output layer are compared with the desired target. Cross-entropy is frequently used to calculate the difference between the expected and predicted multinomial probability distributions, and this difference is then used to update the model (see Equations (9)–(11)). Figure 15 shows the practical investigation of this activation function.

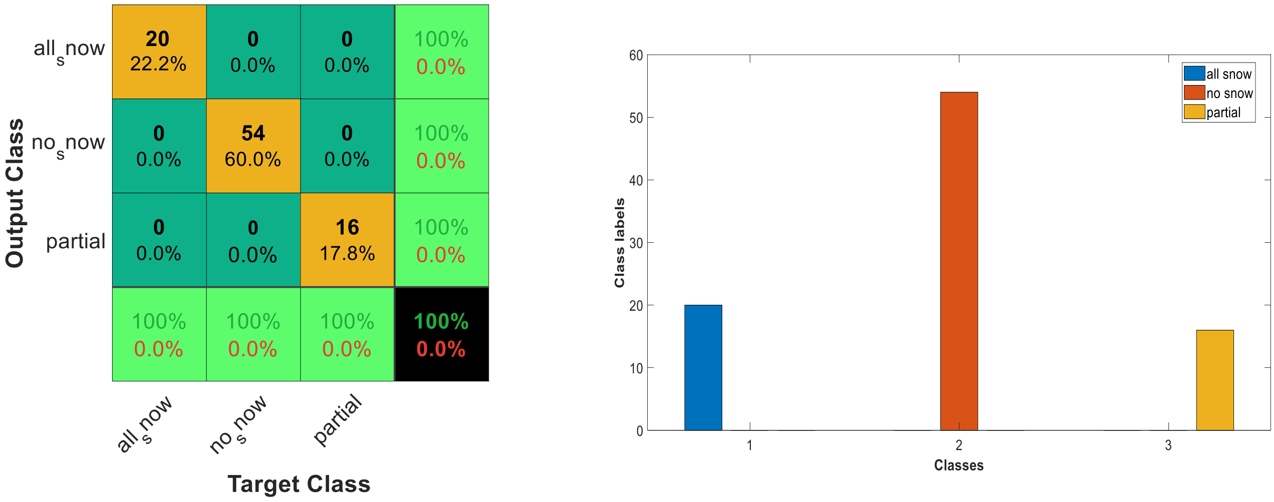
Table 5. Comparison results for second case based on the last iterations of the epochs.

Models	**E**	**I**	**M^{acc}**	**V^{acc}**	**m^{loss}**	**v^{loss}**	**v^{loss} Mean**	**LR**
VGG-16	26	702	100.00%	98.89%	**2.8537×10^{-5}**	**0.0420**	**0.07938**	0.0003
	27	729	100.00%	96.67%	**0.0006**	**0.2663**		0.0003
	28	756	100.00%	100.00%	**1.0395×10^{-5}**	**1.5562×10^{-5}**		0.0003
	29	783	100.00%	100.00%	**0.0002**	**0.0017**		0.0003
	30	810	100.00%	100.00%	**3.7628×10^{-5}**	**0.0028**		0.0003
VGG-19	26	702	100.00%	100.00%	**8.4398×10^{-6}**	**2.1099×10^{-6}**	**0.1194**	0.0003
	27	729	100.00%	100.00%	**3.5763×10^{-8}**	**2.3045×10^{-6}**		0.0003
	28	756	100.00%	100.00%	**1.7643×10^{-6}**	**2.3482×10^{-6}**		0.0003
	29	783	100.00%	100.00%	**6.1989×10^{-7}**	**2.1430×10^{-6}**		0.0003
	30	810	100.00%	100.00%	**0.0006**	**2.0662×10^{-6}**		0.0003
RESNET-18	26	702	100.00%	100.00%	**0.0017**	**0.0360**	**0.1123**	0.0003
	27	729	100.00%	94.44%	**0.0001**	**0.0912**		0.0003
	28	756	100.00%	100.00%	**0.0321**	**0.0219**		0.0003
	29	783	100.00%	97.78%	**6.4626×10^{-5}**	**0.0367**		0.0003
	30	810	100.00%	96.67%	**0.0003**	**0.0509**		0.0003
RESNET-50	26	702	100.00%	96.67%	**0.0006**	**0.1680**	**0.2628**	0.0003
	27	729	100.00%	92.22%	**0.0057**	**0.2466**		0.0003
	28	756	100.00%	92.22%	**0.0086**	**0.1849**		0.0003
	29	783	100.00%	96.67%	**0.0001**	**0.1257**		0.0003
	30	810	100.00%	94.44%	**4.6536×10^{-5}**	**0.1453**		0.0003
RESNET-101	26	702	100.00%	92.22%	**7.6751×10^{-5}**	**0.2197**	**0.2633**	0.0003
	27	729	100.00%	93.33%	**0.0005**	**0.1845**		0.0003
	28	756	100.00%	95.56%	**0.0008**	**0.1203**		0.0003
	29	783	100.00%	93.33%	**0.0019**	**0.1500**		0.0003
	30	810	100.00%	93.33%	**0.0006**	**0.1504**		0.0003
Proposed BIDA-CNN	26	702	100.00%	100.00%	**0.0019**	**0.0002**	**0.03797**	0.0003
	27	729	100.00%	100.00%	0.0031	0.0002		0.0003
	28	756	100.00%	100.00%	**0.0006**	0.0002		0.0003
	29	783	100.00%	100.00%	0.0086	0.0002		0.0003
	30	810	100.00%	100.00%	0.0017	0.0003		0.0003

Note: The bold is the minimum losses values, which shows the researchers which model is the best and which model gets the least loss.

Table 6. Comparison of the testing results based on evaluation metrics for second case.

Models	**Classes**	**S_e**	**P_r**	**F1**	**ACC**
All Models	all snow	1	1	1	100%
	no snow	1	1	1	
	partial	1	1	1	



(**a**) (**b**)

Figure 14. (**a**) Confusion matrix results of VGG-16, VGG-19, RESNET-18, RESNET-50, RESNET-101, and our proposed BIDA-CNN; (**b**) predicted distribution over classes of the compared models VGG-16, VGG-19, RESNET-18, RESNET-50, RESNET-101, and our proposed BIDA-CNN.

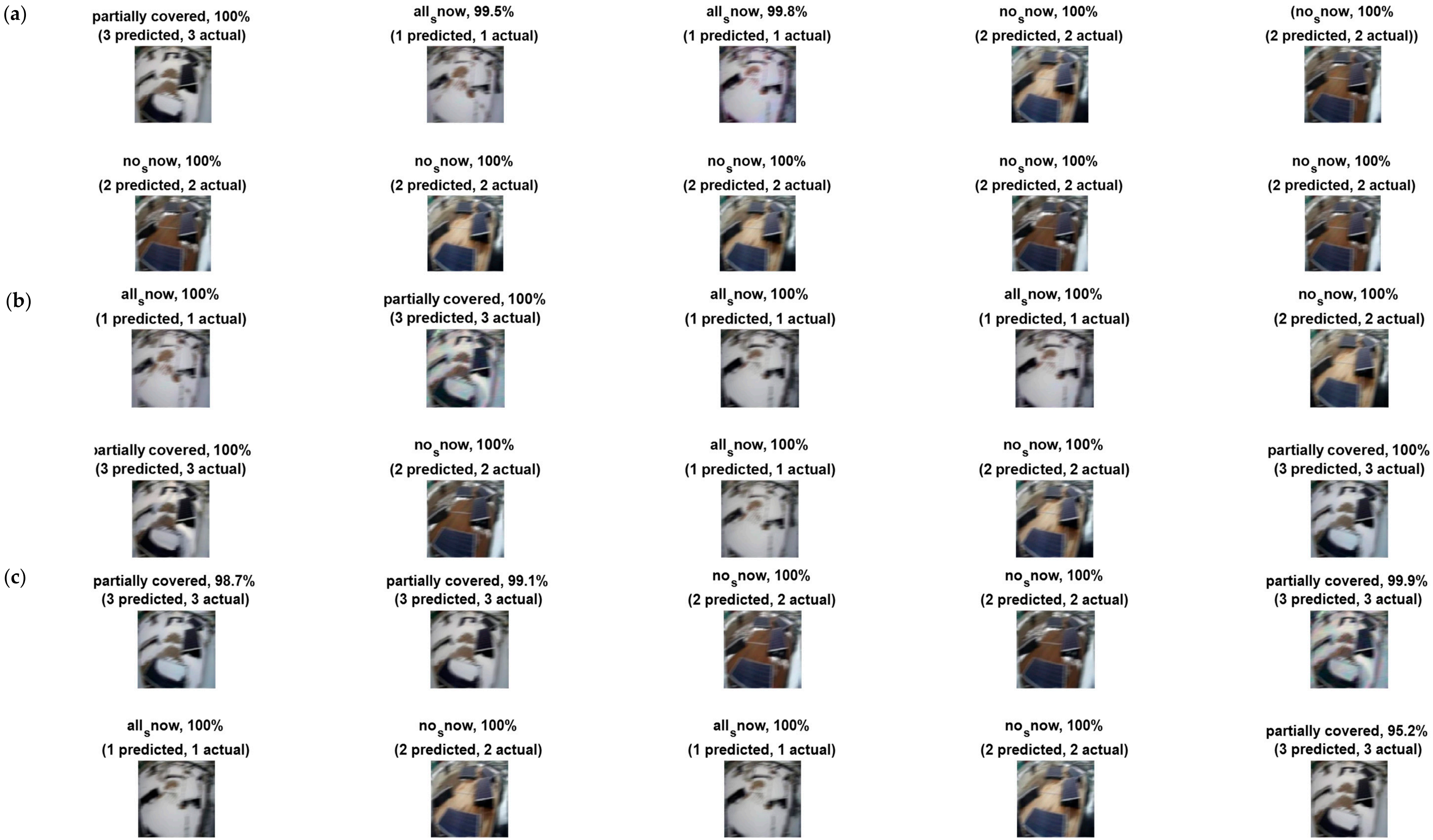


Figure 15. *Cont.*

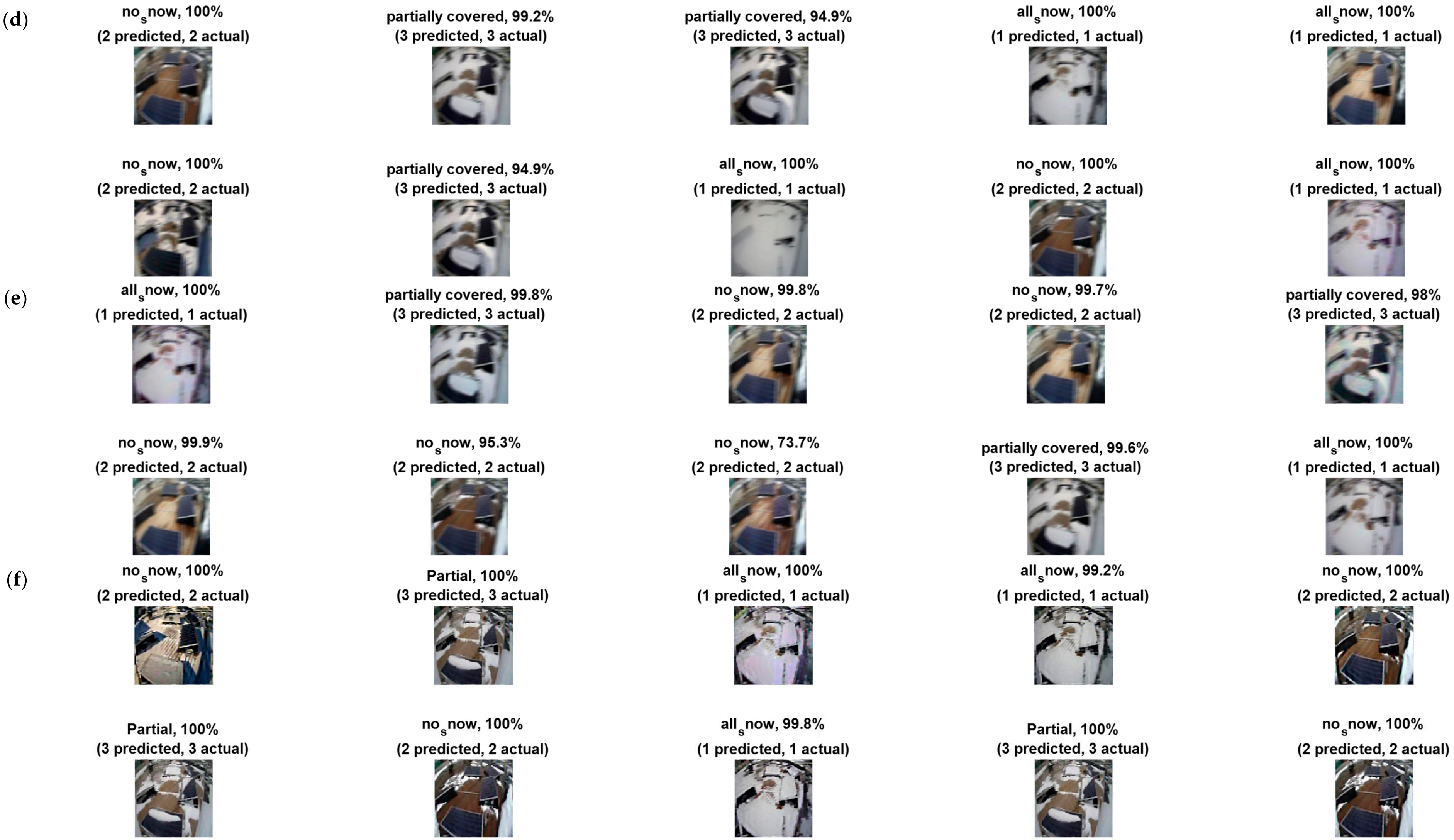
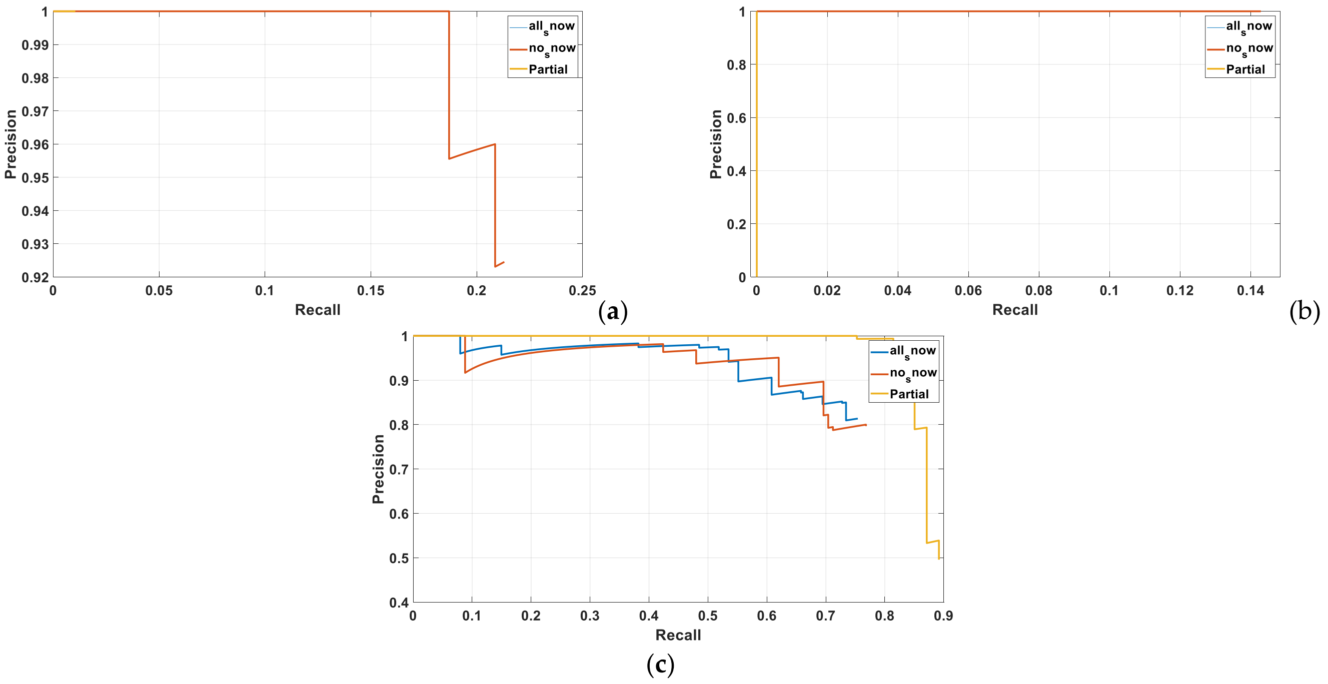


Figure 15. Second case image probabilities with a softmax function for (**a**) VGG-16, (**b**) VGG-19, (**c**) RESNET-18, (**d**) RESNET-50, and (**e**) RESNET-101, as well as the (**f**) proposed BIDA-CNN.

3.2. Detection Performance Results

For the second proposed approach (detection case), the image datasets used in this experiment consist of 442 images for training and 150 for testing. In the first step, we create ground truth for all datasets by using a rectangular box for annotation; after that, we begin the training with a negative overlap range of 0.3, 30 epochs, 19,530 iterations, and a 0.0001 learning rate. The RCNN detector used in this experiment depends on three different backbones: VGG-16, VGG-19, and our proposed BIDA-CNN. At time t, distortion may occur in the image pixels captured by the imaging system, which is crucial for the detection model. The detector cannot work as well with images that have distorted pixels that give an unclean image; therefore, we propose a deblurring model to restore the features of the images. Figure 16 and Table 7 show that the detector based on the VGG-16 and VGG-19 backbones have a breakdown at time t. the BIDA-CNN model achieves high performance at time t. Accordingly, the use of models that depend on preprocessing images is very important and necessary. It increases the efficiency of the models and gives high performance in real time. Figure 17 shows the visual results of all models used in the detection process. The results show the accuracy of the proposed model in recovering the original images, as well as detecting the panels.



(a) (b) (c)

Figure 16. RCNN testing results with backbone (**a**) VGG-16, (**b**) VGG-19, and (**c**) proposed BIDA-CNN.

Table 7. Results evaluation of three RCNN backbones.

Models	**Classes**	**E**	**I**	**AP**
	-	Training		Testing
VGG-16	all snow no snow partial	30	19,530	0.00 0.21 0.01
VGG-19	all snow no snow partial	30	19,530	0.00 0.14 0.00
Proposed BIDA-CNN	all snow no snow partial	30	19,530	**0.71** **0.72** **0.87**

Note: The bold shows the performance of the proposed model.

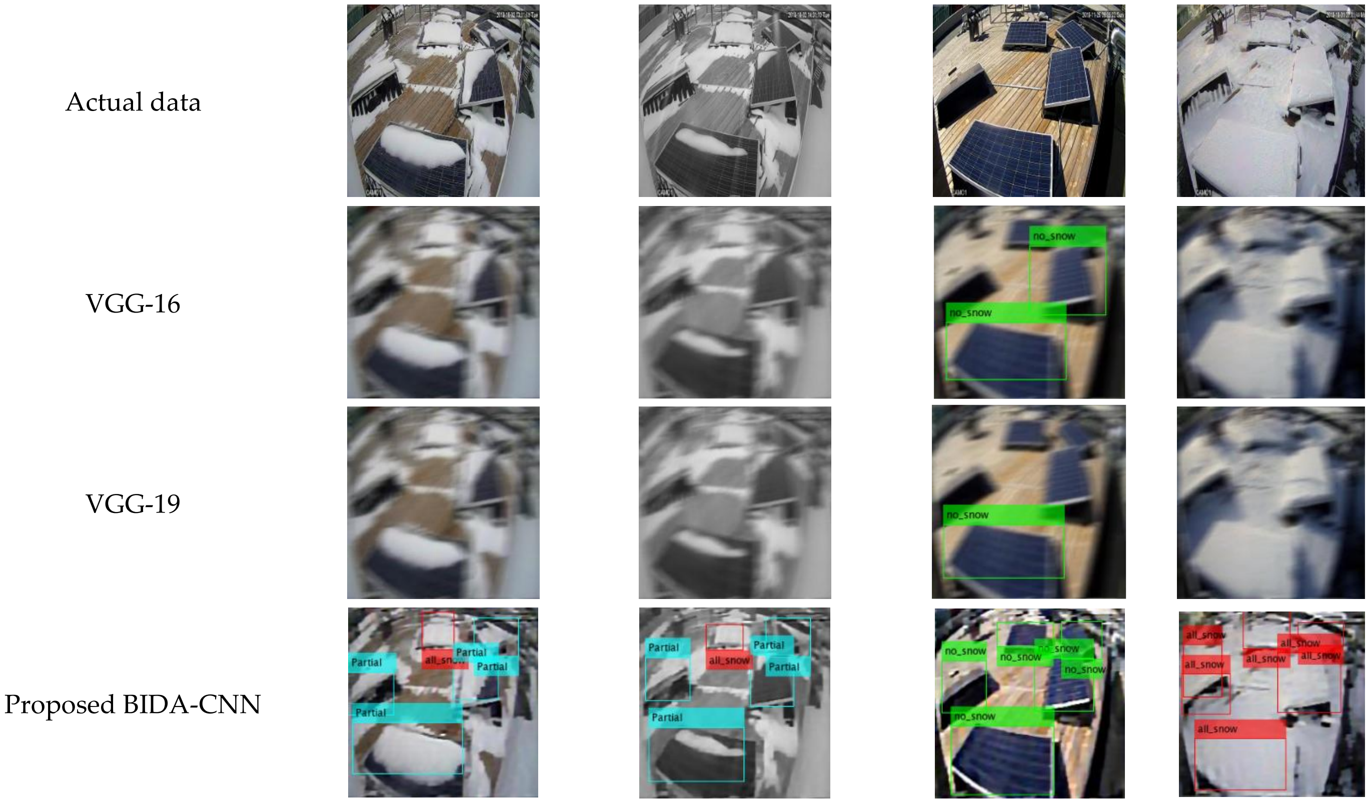


Figure 17. Visual illustration of test results.

4. Conclusions

It is crucial to investigate the factors that affect solar panels and reduce their efficiency in photovoltaic energy production. One main factor that prevents solar panels from working properly is snowfall. Snow must be cleared from solar panels to maximize solar energy production and reduce losses due to snow. Before starting the snow removal process, snow-covered panels should be effectively identified to improve snow removal speed and removal efficiency. The study presented a comprehensive analysis of classification and detection to identify whether solar panels are covered by snow or not. The proposal incorporates five deep learning architectures constructed from the ground up and the proposed BIDA-CNN model, which preprocesses the images before the classification and detection processes. The investigated architectures extract meaningful features based on highly connected deep learning layers. The study performance shows how the augmented classes perform and can give appreciated results. Within the scope of this experiment, six different models, namely VGG-16, VGG-19, RESNET-18, RESNET-50, RESNET-101, and BIDA-CNN, were trained, tested, and compared in terms of their overall performance. For the classification approach, the experiments were conducted in two different cases. The first case showed the comparative performance of all models tested on the original dataset without any preprocessing. The second case simulated extreme climate conditions by generating motion noise and showed how to deal with this issue. The dataset is replicated using the upsampling technique to handle the unbalancing issue. The implemented dataset is tried in two different cases; the first case is with 395 images, and for the second case, the dataset after upsampling is 437 images. The datasets are divided into three classes for both cases—all_snow, no_snow, and partial.

The testing accuracy results for the classification analysis approach show that all six models give the same results with 100% accuracy; for the first case, in the loss metric, the average of the lowest validation loss for all iterations was 0.01865 v^{loss} mean, achieved by the BIDA-CNN model. For the second case, the best model in this experiment was the BIDA-CNN model, which achieved 0.03797 v^{loss} mean. In addition, all models used in this experiment recorded 100% overall precision, sensitivity, F1-score, and testing accuracy. Achieving high accuracy does not mean that the probability of the model was 100%; the probability ranges from 50% to 99%. We always work on proposing models that increase the probability of being correct through a mathematical model input in the decision layer that improves the process of extracting features with important information. Finally, for classification, the learning behavior of BIDA-CNN is more robust than the others by convergence with fewer iterations. The detection analysis consists of one case based on three backbones, namely VGG-16, VGG-19, and BIDA-CNN. The RCNN detector based on BIDA-CNN achieved the best result when recognizing the three categories: all_snow, no_snow, and partial. From this performance analysis, we can conclude that the accuracy results in all cases show that the proposed BIDA-CNN outperformed its peers.

From a future perspective, we plan to investigate other factors and apply any kind of preprocessing and other advanced deep learning techniques. This paper presented a proposal for a new research area, where it is possible through researcher to expand studies and integrate advanced technologies to work on reducing the factors that limit the production of electrical energy; moreover, it is also possible to benefit from the proposals of the paper in the future, especially to detect the defect or damage to which the solar panels are exposed.

Author Contributions: Conceptualization, A.A.A.-D., F.P.G.M. and M.S.; data curation, A.A.A.-D., M.T.G. and A.A.H.; formal analysis, F.P.G.M., N.L.F. and M.S.; investigation, A.A.A.-D., M.T.G. and A.A.H.; methodology, M.T.G., A.A.H. and N.L.F.; software, A.A.A.-D., M.T.G. and M.S.; supervision, F.P.G.M.; validation, F.P.G.M., N.L.F. and M.S.; visualization, A.A.H. and N.L.F.; writing—original draft preparation, A.A.A.-D., M.T.G. and A.A.H.; writing—review and editing, F.P.G.M., N.L.F. and M.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Yamashita, R.; Nishio, M.; Do, R.K.G.; Togashi, K. Convolutional Neural Networks: An Overview and Application in Radiology. *Insights Imaging* **2018**, *9*, 611–629. [CrossRef] [PubMed]
2. Alzubaidi, L.; Zhang, J.; Humaidi, A.J.; Al-Dujaili, A.; Duan, Y.; Al-Shamma, O.; Santamaría, J.; Fadhel, M.A.; Al-Amidie, M.; Farhan, L. Review of Deep Learning: Concepts, CNN Architectures, Challenges, Applications, Future Directions. *J. Big Data* **2021**, *8*, 53. [CrossRef] [PubMed]
3. Simonyan, K.; Zisserman, A. Very Deep Convolutional Networks for Large-Scale Image Recognition. In Proceedings of the 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, 7–9 May 2015.
4. Sangeetha, V.; Prasad, K.J.R. Syntheses of Novel Derivatives of 2-Acetylfuro[2,3-a]Carbazoles, Benzo[1,2-b]-1,4-Thiazepino[2,3-a]Carbazoles and 1-Acetyloxycarbazole-2-Carbaldehydes. *ChemInform* **2006**, *37*. [CrossRef]
5. Ahsan, M.M.; Ahad, M.T.; Soma, F.A.; Paul, S.; Chowdhury, A.; Luna, S.A.; Yazdan, M.M.S.; Rahman, A.; Siddique, Z.; Huebner, P. Detecting SARS-CoV-2 From Chest X-Ray Using Artificial Intelligence. *IEEE Access* **2021**, *9*, 35501–35513. [CrossRef] [PubMed]
6. Xu, X.; Jiang, X.; Ma, C.; Du, P.; Li, X.; Lv, S.; Yu, L.; Ni, Q.; Chen, Y.; Su, J.; et al. A Deep Learning System to Screen Novel Coronavirus Disease 2019 Pneumonia. *Engineering* **2020**, *6*, 1122–1129. [CrossRef] [PubMed]
7. Kumari, P.; Toshniwal, D. Real-Time Estimation of COVID-19 Cases Using Machine Learning and Mathematical Models—The Case of India. In Proceedings of the 2020 IEEE 15th International Conference on Industrial and Information Systems (ICIIS), Rupnagar, India, 26–28 November 2020; pp. 369–374.
8. Ishengoma, F.S.; Rai, I.A.; Said, R.N. Identification of Maize Leaves Infected by Fall Armyworms Using UAV-Based Imagery and Convolutional Neural Networks. *Comput. Electron. Agric.* **2021**, *184*, 106124. [CrossRef]
9. Zhu, H.; Yang, L.; Fei, J.; Zhao, L.; Han, Z. Recognition of Carrot Appearance Quality Based on Deep Feature and Support Vector Machine. *Comput. Electron. Agric.* **2021**, *186*, 106185. [CrossRef]
10. Hashemi, B.; Cretu, A.-M.; Taheri, S. Snow Loss Prediction for Photovoltaic Farms Using Computational Intelligence Techniques. *IEEE J. Photovolt.* **2020**, *10*, 1044–1052. [CrossRef]

11. Solangi, K.H.; Islam, M.R.; Saidur, R.; Rahim, N.A.; Fayaz, H. A Review on Global Solar Energy Policy. *Renew. Sustain. Energy Rev.* **2011**, *15*, 2149–2163. [CrossRef]
12. Marion, B.; Schaefer, R.; Caine, H.; Sanchez, G. Measured and Modeled Photovoltaic System Energy Losses from Snow for Colorado and Wisconsin Locations. *Sol. Energy* **2013**, *97*, 112–121. [CrossRef]
13. Pawluk, R.E.; Chen, Y.; She, Y. Photovoltaic Electricity Generation Loss Due to Snow—A Literature Review on Influence Factors, Estimation, and Mitigation. *Renew. Sustain. Energy Rev.* **2019**, *107*, 171–182. [CrossRef]
14. Andrews, R.W.; Pollard, A.; Pearce, J.M. The Effects of Snowfall on Solar Photovoltaic Performance. *Sol. Energy* **2013**, *92*, 84–97. [CrossRef]
15. Andrews, R.W.; Pearce, J.M. Prediction of Energy Effects on Photovoltaic Systems Due to Snowfall Events. In Proceedings of the 2012 38th IEEE Photovoltaic Specialists Conference, Austin, TX, USA, 3–8 June 2012; pp. 003386–003391.
16. Hosseini, S.; Taheri, S.; Farzaneh, M.; Taheri, H. Modeling of Snow-Covered Photovoltaic Modules. *IEEE Trans. Ind. Electron.* **2018**, *65*, 7975–7983. [CrossRef]
17. Hayibo, K.S.; Petsiuk, A.; Mayville, P.; Brown, L.; Pearce, J.M. Monofacial vs Bifacial Solar Photovoltaic Systems in Snowy Environments. *Renew. Energy* **2022**, *193*, 657–668. [CrossRef]
18. Sharma, V.; Chandel, S.S. Performance and Degradation Analysis for Long Term Reliability of Solar Photovoltaic Systems: A Review. *Renew. Sustain. Energy Rev.* **2013**, *27*, 753–767. [CrossRef]
19. Tsanakas, J.A.; Vannier, G.; Plissonnier, A.; Ha, D.L.; Barruel, F. Fault Diagnosis and Classification of Large-Scale Photovoltaic Plants through Aerial Orthophoto Thermal Mapping. In Proceedings of the 31st European Photovoltaic Solar Energy Conference and Exhibition, Hamburg, Germany, 14–18 September 2015; pp. 1783–1788. [CrossRef]
20. Eder, G.; Voronko, Y.; Hirschl, C.; Ebner, R.; Újvári, G.; Mühleisen, W. Non-Destructive Failure Detection and Visualization of Artificially and Naturally Aged PV Modules. *Energies* **2018**, *11*, 1053. [CrossRef]
21. Grimaccia, F.; Leva, S.; Dolara, A.; Aghaei, M. Survey on PV Modules' Common Faults After an O&M Flight Extensive Campaign Over Different Plants in Italy. *IEEE J. Photovolt.* **2017**, *7*, 810–816. [CrossRef]
22. Liao, K.-C.; Lu, J.-H. Using UAV to Detect Solar Module Fault Conditions of a Solar Power Farm with IR and Visual Image Analysis. *Appl. Sci.* **2021**, *11*, 1835. [CrossRef]
23. Starzyński, J.; Zawadzki, P.; Harańczyk, D. Machine Learning in Solar Plants Inspection Automation. *Energies* **2022**, *15*, 5966. [CrossRef]
24. Supe, H.; Avtar, R.; Singh, D.; Gupta, A.; Yunus, A.P.; Dou, J.A.; Ravankar, A.; Mohan, G.; Chapagain, S.K.; Sharma, V.; et al. Google Earth Engine for the Detection of Soiling on Photovoltaic Solar Panels in Arid Environments. *Remote Sens.* **2020**, *12*, 1466. [CrossRef]
25. Deitsch, S.; Christlein, V.; Berger, S.; Buerhop-Lutz, C.; Maier, A.; Gallwitz, F.; Riess, C. Automatic Classification of Defective Photovoltaic Module Cells in Electroluminescence Images. *Sol. Energy* **2019**, *185*, 455–468. [CrossRef]
26. Su, B.; Chen, H.; Zhu, Y.; Liu, W.; Liu, K. Classification of Manufacturing Defects in Multicrystalline Solar Cells With Novel Feature Descriptor. *IEEE Trans. Instrum. Meas.* **2019**, *68*, 4675–4688. [CrossRef]
27. Chen, H.; Hu, Q.; Zhai, B.; Chen, H.; Liu, K. A Robust Weakly Supervised Learning of Deep Conv-Nets for Surface Defect Inspection. *Neural Comput. Applic* **2020**, *32*, 11229–11244. [CrossRef]
28. Fioresi, J.; Colvin, D.J.; Frota, R.; Gupta, R.; Li, M.; Seigneur, H.P.; Vyas, S.; Oliveira, S.; Shah, M.; Davis, K.O. Automated Defect Detection and Localization in Photovoltaic Cells Using Semantic Segmentation of Electroluminescence Images. *IEEE J. Photovolt.* **2022**, *12*, 53–61. [CrossRef]
29. Karabuk University Provides Geographic Data Sets. Available online: https://www.karabuk.edu.tr/en/ (accessed on 25 September 2022).
30. Solar and Meteorological Data Sets from NASA. Available online: https://power.larc.nasa.gov/ (accessed on 25 September 2022).
31. Gu, J.; Wang, Z.; Kuen, J.; Ma, L.; Shahroudy, A.; Shuai, B.; Liu, T.; Wang, X.; Wang, G.; Cai, J.; et al. Recent Advances in Convolutional Neural Networks. *Pattern Recognit.* **2018**, *77*, 354–377. [CrossRef]
32. Rasheed, J.; Hameed, A.A.; Djeddi, C.; Jamil, A.; Al-Turjman, F. A Machine Learning-Based Framework for Diagnosis of COVID-19 from Chest X-Ray Images. *Interdiscip. Sci. Comput. Life Sci.* **2021**, *13*, 103–117. [CrossRef]
33. Zhao, X.; Wei, H.; Wang, H.; Zhu, T.; Zhang, K. 3D-CNN-Based Feature Extraction of Ground-Based Cloud Images for Direct Normal Irradiance Prediction. *Sol. Energy* **2019**, *181*, 510–518. [CrossRef]
34. Dong, J.; Pan, J.; Su, Z.; Yang, M.-H. Blind Image Deblurring with Outlier Handling. In Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017; pp. 2497–2505.
35. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. ImageNet Classification with Deep Convolutional Neural Networks. *Commun. ACM* **2017**, *60*, 84–90. [CrossRef]
36. Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. In Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014; pp. 580–587.