sensors

Article

A Robust Deep Learning Ensemble-Driven Model for Defect and Non-Defect Recognition and Classification Using a Weighted Averaging Sequence-Based Meta-Learning Ensembler

Okeke Stephen *, Samaneh Madanian and Minh Nguyen

Computer Science & Software Engineering, Auckland University of Technology, Auckland 1010, New Zealand
* Correspondence: stephen.okeke@aut.ac.nz; Tel.: +64-20-4083-5671

Abstract: The need to overcome the challenges of visual inspections conducted by domain experts drives the recent surge in visual inspection research. Typical manual industrial data analysis and inspection for defects conducted by trained personnel are expensive, time-consuming, and characterized by mistakes. Thus, an efficient intelligent-driven model is needed to eliminate or minimize the challenges of defect identification and elimination in processes to the barest minimum. This paper presents a robust method for recognizing and classifying defects in industrial products using a deep-learning architectural ensemble approach integrated with a weighted sequence meta-learning unification framework. In the proposed method, a unique base model is constructed and fused together with other co-learning pretrained models using a sequence-driven meta-learning ensembler that aggregates the best features learned from the various contributing models for better and superior performance. During experimentation in the study, different publicly available industrial product datasets consisting of the defect and non-defect samples were used to train, validate, and test the introduced model, with remarkable results obtained that demonstrate the viability of the proposed method in tackling the challenges of the manual visual inspection approach.

Keywords: deep learning ensemble; defect recognition and classification; visual inspection; industrial products; product quality control; conv-LSTM

check for updates

Citation: Stephen, O.; Madanian, S.; Nguyen, M. A Robust Deep Learning Ensemble-Driven Model for Defect and Non-Defect Recognition and Classification Using a Weighted Averaging Sequence-Based Meta-Learning Ensembler. *Sensors* **2022**, *22*, 9971. https://doi.org/10.3390/s22249971

Academic Editor: Maria Gabriella Xibilia

Received: 1 December 2022
Accepted: 14 December 2022
Published: 17 December 2022

1. Introduction

Sustaining quality standards is a crucial task for every industry, and visual inspections deal with the detection of defects from manufactured products for quality control. Quality inspections can be conducted at any stage of the industrial production circle, such as product components, products within the manufacturing lines, incoming material, or finished products. Inspection examines products to determine those that meet the set standards and those that deviate from the set quality requirements, paving the way for the rejection of faulty products and progression to the next stage of those that conform to the set standards [1]. In situ or in-process inspections are standard practices conducted during industrial parts and other product manufacturing processes [2]. Manual inspections for defects are characterized by challenges such as boredom of inspection operators, failure to meet production targets, bias, inadequate inspection skillset, subjective judgements, etc.

The limitations of the human-oriented industrial visual inspection for faulty product identification could be addressed through independent, intelligent models and computer vision algorithms. In recent years, intelligent machine vision models have become desirable in tackling high costs and other shortcomings of human-driven defect recognition and analysis processes. Deep learning (DL) models, in particular, convolutional neural networks (CNNs), have been increasingly used in the automation of inspection processes [3,4]. CNN-driven models have proven effective in performing visual inspections by recognizing, classifying, and detecting defects and non-defects in objects of interest [5]. Despite the remarkable performance of deep learning techniques in recent times, significant issues and challenges, such as model robustness, performance accuracy, and efficiency, are still abound. Therefore, in this work, we propose a robust deep-learning method driven by the model ensemble concept with a sequence-enabled meta-learning unifier to perform the recognition and classification of an industrial product for defect identification (see Figure 1).

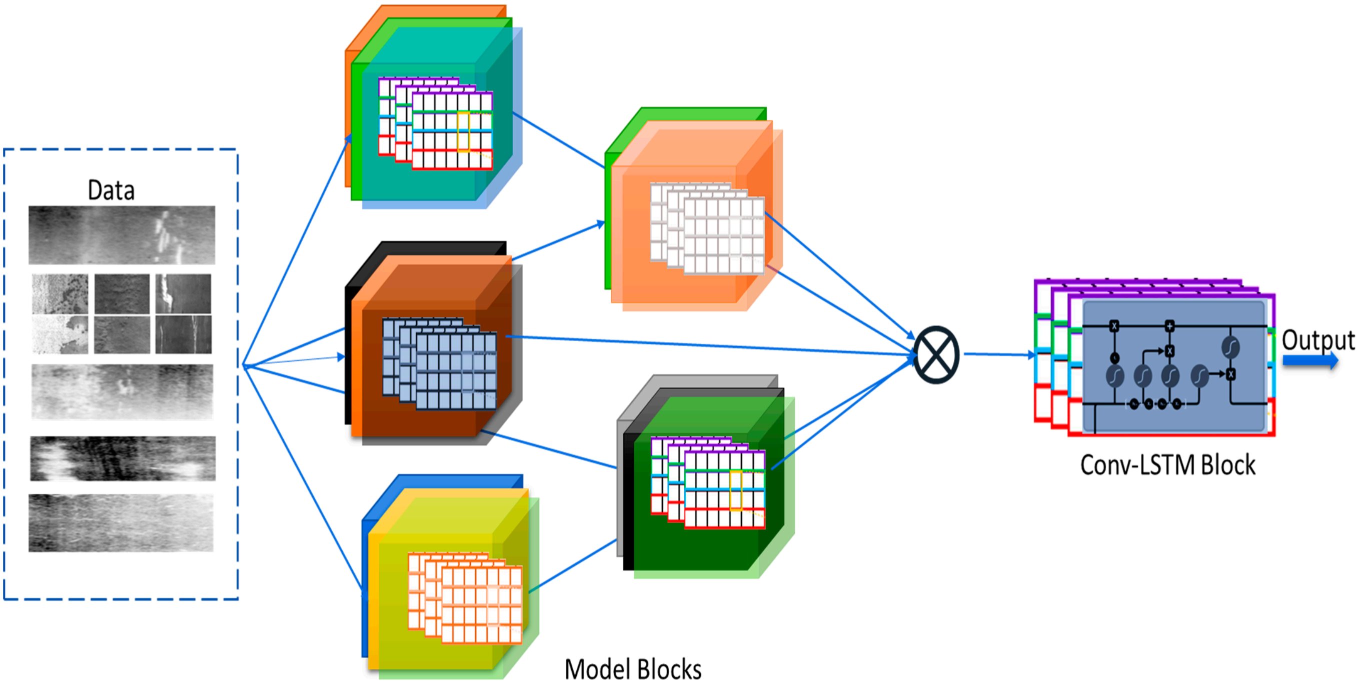


Figure 1. The flow sketch of the proposed Weighted Averaging Sequence-based Meta-learning Ensembler.

This article section presents an overview of the current state and the limitations of human-in-the-loop industrial product inspection systems. In Section 2, relevant and related literature were explored, ranging from the use of complex multilayer CNN architecture, conditional random field (CRFs) algorithm with CNN, fully convolutional network (FCN), meta-learning CNN architectural framework, deep convolutional sparse-coding-based network, etc. Section 3 provides the theoretical background and the method adopted in the study, while the experimental procedure used in the study is elaborated in Section 4. In Section 5, the results obtained from the study are presented and concisely explained. Comparisons with related works are also made in Section 5, while the study's conclusion is presented in Section 6.

2. Related Works

Recently, product defect identification and classification for visual inspections have attracted considerable research interest. He et al. [6] and Borji et al. [7] deployed a deep-learning model based on the LeNet network structure [8]. Their proposed framework detects defects in industrial products using a complex multilayer CNN architecture to extract defect image features and then a full end-to-end training process to learn and classify the defects. In another work on defect spotting and the classification of products, a CNN model and a conditional random field (CRFs) algorithm were combined to train and optimize a built DL network prediction process [9]. Xue and Li (2018) deployed a region-based fully convolutional network (FCN) DL model to build an intelligent classification and detection model for rapid tunnel lining defect detections.

Furthermore, Bartler et al. [10] proposed a DL-based classification pipeline to identify solar cell defects automatically. A meta-learning CNN architectural framework was introduced to perform a multi-target concrete defect classification in concrete bridge image frames [11,12]. In another work, a DL model was deployed to ensure sustainable transportation by developing a model that fused the features of two models for high accuracy in classifying defects on rail tracks [13]. Krummenacher et al. [14] proposed a wheel defect identification system based on machine learning methods on railway wagons for easy damage recognition on rolling stocks and railway infrastructures. A deep convolutional sparse coding-based network was deployed to perform tire defect classification tasks to ensure an efficient quality control process [15]. A weld defects classification framework driven by transfer learning and activation features of deep learning was proposed to detect defects on industrial weld X-ray images for a rapid, nondestructive test process [16].

Konovalenko et al. [17], in their work on defect classification, proposed a deep residual neural network-based model to classify defects and non-defects on steel surfaces. In a similar study, a time-efficient steel surface defect classification built with a completed local binary pattern was introduced by Luo et al. [18]. Wang et al. (2021) presented a graph convolution network-based semi-supervised model to learn the inter-class similarities and intra-class variations in surfaces for fault and non-fault recognition and classifications. With the aid of a hybrid chromosome genetic algorithm, Hu et al. [19] developed a large-scale strip steel surface defect classification framework. Additionally, an automatic PCB defect classification, analysis, and inspection system was introduced by Deng et al. [20], and Zhang et al. [21] proposed a multi-label class classification of PCB defects using a multi-task convolutional neural network framework. For a micro-defect diagnosis on piston throats, Chen et al. [22] proposed a SMOTE in conjunction with a new model selection method utilized on the active learning of the SVM algorithm (E-SVM-AL). Additionally, an image processing-based piston surface defect recognition system combined different strategies, such as edge detection, threshold segmentation, and morphological operations, to recognize defects on piston surfaces [23]. Furthermore, Nikolić et al. [24] introduced a deep learning-based classification methodology to detect the porosity defects in aluminum alloys, and Habibpour et al. [25] proposed an uncertainty-aware deep learning model to detect defects in industrial casting products. Despite these studies on defect recognition and classification, little effort has been made on the robust model for the defect spotting in products; therefore, we propose a weighted sequence-based meta-learning ensemble on a collection of models aggregated together to learn the class and interclass similarities and dissimilarities in objects for defect and non-defect separation.

3. Theoretical Background and Method

This section presents the underground theoretical method for the proposed defects recognition and classification framework. Let $D = \{(d_z, c_z)\, 1 \leq z \leq N\}$ represent the dataset consisting of N number of training samples, with $c_z = \{1, 2, \ldots, C\}$ their corresponding class labels and C the total sum of the classes. Then, the proposed model contains M different numbers of deep learning models fused with convolutional LSTM layers that learn from the meta-features emanating from the various participating models for superior performance. The proposed method can ensemble different numbers of given CNN models. However, in this study, we used $M = 5$ number of models for the metal surface defects classification and $M = 6$ for the other datasets. In the deep learning ensemble process, the resultant features R from the various model is expressed as $R = [r_1 r_2, \ldots, r_n]$. During training, a forward propagation process is conducted in each epoch to generate features from each co-learning model and then fussed together by the integrated sequence-based convolutional LSTM layers.

3.1. The Contributing CNN Models

In this investigation, we crafted a unique base model and adopted four other state-of-the-art convolutional neural network-based models: Inceptionv3 [26], DenseNet [27], Xception [28], and MobileNet [29]. The built base model contains four significant layers and sublayers, as shown in Table 1, with 223,873 total parameters used in the model training process. The respective feature extractors in the CNN architecture have conv2 × 3, 32;

conv2 × 3, 64; and conv2 × 3, 128 layer sizes, as well as a 2 × 2 max pooling layer size between the 1st and the 2nd layers. A Relu activator was equally used in the 1st three layers (see Table 1), and a stride of 2 was used across all the layers excluding the last layer.

Table 1. The conventional base model.

Layer Type	Output Shape	Parameters
conv2d (Conv2D)	(None, 45, 45, 32)	320
max_pooling2d	(None, 22, 22, 32)	0
conv2d_1 (Conv2D)	(None, 11, 11, 64)	18,496
max_pooling2d_1	(None, 5, 5, 64)	0
flatten (Flatten)	(None, 1600)	0
dense (Dense)	(None, 128)	204,928
dense_1 (Dense)	(None, 1)	129

Inception-v3, which belongs to the inception model group, consists of a label smoothing mechanism, factorized 7 × 7 convolutions, and an auxiliary classifier that channels training data label details from the top to the lower ebb of the network [26]. The DenseNet (densely connected convolutional networks) [27], on the other hand, is a variant of the deep CNN model that consists of dense blocks and uses dense connections between layers in the network to propagate information across the network. Furthermore, the Xception model depends on depth-wise separable convolution layers to compute the spatial information from the training and validation data. Finally, on the co-learning models, the MobileNet was initially designed and built for mobile applications [29]; however, many applications have adopted the framework to solve different scientific problems [30–32].

All the adopted models in this study were ImageNet dataset pre-trained, but the decision layers were chopped off during our experiments because the models were pre-trained initially to classify objects of 1000 classes. During the training process in the experiments conducted in this study, the meta-features that emanated from the various models were concatenated via the integrated weighted averaging sequence-based meta-learning ensemble, which then performed the final classification tasks. The convolution components of the models were useful in extracting the features from the defect and non-defect data samples. The features fed to the unification framework are called meta-features and are significantly valuable for distinguishing the defective and non-defectives data samples. The lower layer of the networks extracted the local image data features, while the higher layers extracted more semantic meta-features through convolution operations.

3.2. The Weighted Averaging Sequence-Based Meta-Feature Learning Derivative

In the weighted averaging ensemble strategy, the final model's classification report was acquired by obtaining the outputs of the various contributing models and averaging the results with some weight inducements for better predictive performance. This, in particular, motivated us to adopt this approach because of its robustness and ability to handle imbalanced datasets, such as the ones used in this investigation. The weighted averaging sequence-based meta-feature learning component of the proposed method was inspired by the work of Shi et al. [33], in which the meta-features from all the inputs $I_1, \cdots I_{t,}$ the output of the cells $O_1, \cdots O_t$, the hidden states $H_1, \cdots H_t$, and the gates g_t, l_t, m_t of the ConvLSTM layer were 3D tensors that enable our proposed method to learn the spatial meta-features of the defect and non-defect samples in the final aggregate layers (see Equations (1)–(5)). In other words, the ensemble layer of the proposed method consists of ConvLSTM layers that possess convolution operators which join the features emanating from the various co-learning models. The ConvLSTM layer component of the proposed weighted averaging sequence-based meta-feature learning is expressed as:

$$g_t = \sigma\left(W_{ig} \cdot I_t + W_{hg} \cdot \, H_{t-1} + W_{og} \cdot O_{t-1} + k_g\right) \tag{1}$$

$$l_t = \sigma(W_{il} \cdot I_t + W_{hl} \cdot H_{t-1} + W_{ol} \cdot O_{t-1} + k_l) \quad (2)$$

$$O_t = l_t \cdot O_{t-1} + g_t \cdot tanh(W_{io} \cdot I_t + W_{ho} \cdot H_{t-1} + k_o) \quad (3)$$

$$m_t = \sigma(W_{im} \cdot I_t + W_{hm} \cdot H_{t-1} + W_{om} \cdot O_t + k_m) \quad (4)$$

$$H_t = m_t \cdot tanh(O_t) \quad (5)$$

In training the model, padding is required before the application of convolution operations to guarantee that the same number of matrix computations are conducted as the inputs possessed by the state. In the ConvLSTM layer, all the states in the LSTM are initialized to zero before the arrival of the first input, and a zero-padding was used in the hidden states in this study, so that the boundary points of the training dataset were computed differently for prompt learning of the intra-class differences in the defect and non-defect samples.

4. Experiments

4.1. Dataset Preparation Process

We adopted four datasets consisting of different types of defect and non-defect samples to train and validate the proposed method for robustness and performance. The first dataset employed in this study is the NEU (Northeastern University) surface defect dataset consisting of six distinct classes of typical surface defects [34]. The data were collected and made available for research from hot-rolled steel strips with patches (Pa), pitted surface (PS), rolled-in scale (RS), crazing (Cr), scratches (Sc), and inclusion (In) as defect classes. The dataset initially contained 1,800 grayscale images of 200 × 200 size. During the experiments in this study, the dataset was split into 1152 samples for the train set, 288 samples for validation, and 360 for testing the final trained model. The second dataset [35] used in the experiment consisted of 512 × 512 grey scale-size images of submersible pump impellers with defects and non-defect samples available publicly for research. The total number of images was 7348 samples, which were reduced to 300 × 300 grayscale sizes and split into 4644 training samples, 1989 validation samples, and 715 samples for testing the proposed model.

Furthermore, the third dataset employed in this investigation was the printed circuit board (PCB) industrial dataset [36], which comprised 1500 images of defect and non-defect samples. The data were obtained from linear scan CCD processes in the resolution range of 48 pixels per 1 mm. The dataset was cross-examined and certified for suitability for training and validating the proposed model and was split into 892 training samples, 223 validation samples, and 180 test samples. Finally, the fourth dataset used in the experiments was the piston image dataset from industrial mechanic components with shaped-out, greasy, broken, fallen, rust stains, and oily class samples [37]. The dataset contained 285 samples and was collected during the AC's pistons production process. During the experiments in this study, the dataset was divided into 173 samples for training, 42 validation samples, and 70 test samples.

4.2. Experimental and Evaluation Metrics

We implemented the proposed DL method using a high-end computing resource integrated with two GPU cores, each having a 12GB video card and a RAM size of 32GB. Keras and TensorFlow open-source DL libraries, in conjunction with other supportive python modules, such as NumPy, pandas, matplotlib, sklearn, etc., as well as a Linux operating system, were also used for the implementation of the introduced DL method. There were 30 epochs, each involved in the training and validating of the individual models used in the experiments. Furthermore, a binary cross-entropy loss function was used for the datasets, except for the NEU categorical dataset; thus, a categorical cross-entropy loss function was used. A $1 \times 10 - 3 \times 0.9$ learning rate scheduler with a 1×10^{-4} learning rate Adam optimizer was employed to train and validate the models. To help boost the performance of the models' entire training and learning process, standard augmentation techniques were deployed to artificially increase the training data (i.e., horizontal flipping, random cropping, rotation, and shear range). At the end of the training and validation process of the selected models, the meta-features obtained from the models were aggregated using the weighted averaging sequence-based meta-feature learning ensembler to form the new proposed model.

Different DL model evaluation metrics were employed to thoroughly examine the results obtained from the experiments run in this investigation. One such metric is the Cohen kappa score, which calculates the inter-rater trustworthiness of the proposed model. Additionally, Matthew's correlation coefficient (MCC) estimates the quality of association between the pairs of the defect and non-defect samples. The mean square error measures the mean square of the difference between the actual data samples and the predicted samples. In contrast, the mean square log error obtains the relative error between the actual defective and non-defect data samples and the predicted samples. We further extracted the precision, recall, F1-score, and weighted average scores of the individual models and the proposed model, thus solidifying the results of the proposed method.

5. Results

We first constructed a unique, but efficient, custom CNN model to handle the problems of dataset limitations effectively. The CNN model is simple and unique because it consists of a few layers and parameters (see Table 1), but proactively learns the defect's features and non-defect data samples. Additionally, we adopted the transfer learning (TL) concept on the other pre-trained models selected in this study. We first trained the custom model, Inceptionv3, Xception, DenseNet, and MobileNet, with the data samples allotted for training. Then, we fused the outputs of the models using the introduced weighted averaging sequence-based meta-feature learning ensembler to form an entirely new model for superior performance. A comparative performance of each of the adopted co-learning models with the proposed model was conducted and tabulated accordingly (see Tables 2–5). The classification performance, with respect to the Cohen kappa (Kp), Matthew's correlation coefficient (MCC), accuracy, mean square error (MSE), and mean square log error (MSLE), using the NEU dataset, is shown in Table 2 below.

Table 2. The classification report for each of the models and the final model using the NEU dataset.

	Kp	MCC	Accuracy	MSE	MSLE
Inceptionv3	9.93×10^{-1}	9.93×10^{-1}	9.94×10^{-1}	4.72×10^{-2}	3.58×10^{-3}
Custom	9.99×10^{-1}	9.95×10^{-1}	9.94×10^{-1}	4.21×10^{-2}	3.49×10^{-3}
DenseNet	9.99×10^{-1}	9.94×10^{-1}	9.94×10^{-1}	4.53×10^{-2}	2.78×10^{-3}
MobileNet	9.94×10^{-1}	9.98×10^{-1}	9.93×10^{-1}	3.17×10^{-2}	2.34×10^{-3}
Proposed	9.99×10^{-1}	$1.00 \times 10^{+00}$	$1.00 \times 10^{+00}$	3.47×10^{-4}	3.48×10^{-6}

Table 3. The classification report for each of the models and the final model using the piston dataset.

	Kp	MCC	Accuracy	MSE	MSLE
Custom	8.51×10^{-1}	8.61×10^{-1}	9.43×10^{-1}	5.71×10^{-2}	2.75×10^{-2}
Inceptionv3	7.90×10^{-1}	7.90×10^{-1}	9.14×10^{-1}	8.57×10^{-2}	4.12×10^{-2}
Xception	9.66×10^{-1}	9.66×10^{-1}	9.86×10^{-1}	1.43×10^{-2}	6.87×10^{-3}
Densenet	7.46×10^{-1}	7.79×10^{-1}	9.10×10^{-1}	1.00×10^{-1}	4.61×10^{-2}
MobileNet	7.66×10^{-1}	7.69×10^{-1}	9.00×10^{-1}	1.00×10^{-1}	4.81×10^{-2}
Proposed	9.80×10^{-1}	9.72×10^{-1}	9.49×10^{-1}	2.89×10^{-2}	2.93×10^{-2}

Table 4. The classification report for each of the models and the final model using the casting dataset.

	Kp	MCC	Accuracy	MSE	MSLE
Custom	9.88×10^{-1}	9.88×10^{-1}	9.94×10^{-1}	5.59×10^{-3}	2.69×10^{-3}
Inceptionv3	9.88×10^{-1}	9.88×10^{-1}	9.94×10^{-1}	5.59×10^{-3}	2.69×10^{-3}
Xception	9.91×10^{-1}	9.91×10^{-1}	9.96×10^{-1}	4.20×10^{-3}	2.02×10^{-3}
DenseNet	9.85×10^{-1}	9.94×10^{-1}	9.93×10^{-1}	4.53×10^{-2}	2.78×10^{-3}
MobileNet	9.88×10^{-1}	9.88×10^{-1}	9.94×10^{-1}	5.59×10^{-3}	2.69×10^{-3}
Proposed	9.98×10^{-1}	$1.00 \times 10^{+00}$	$1.00 \times 10^{+00}$	6.70×10^{-6}	7.80×10^{-8}

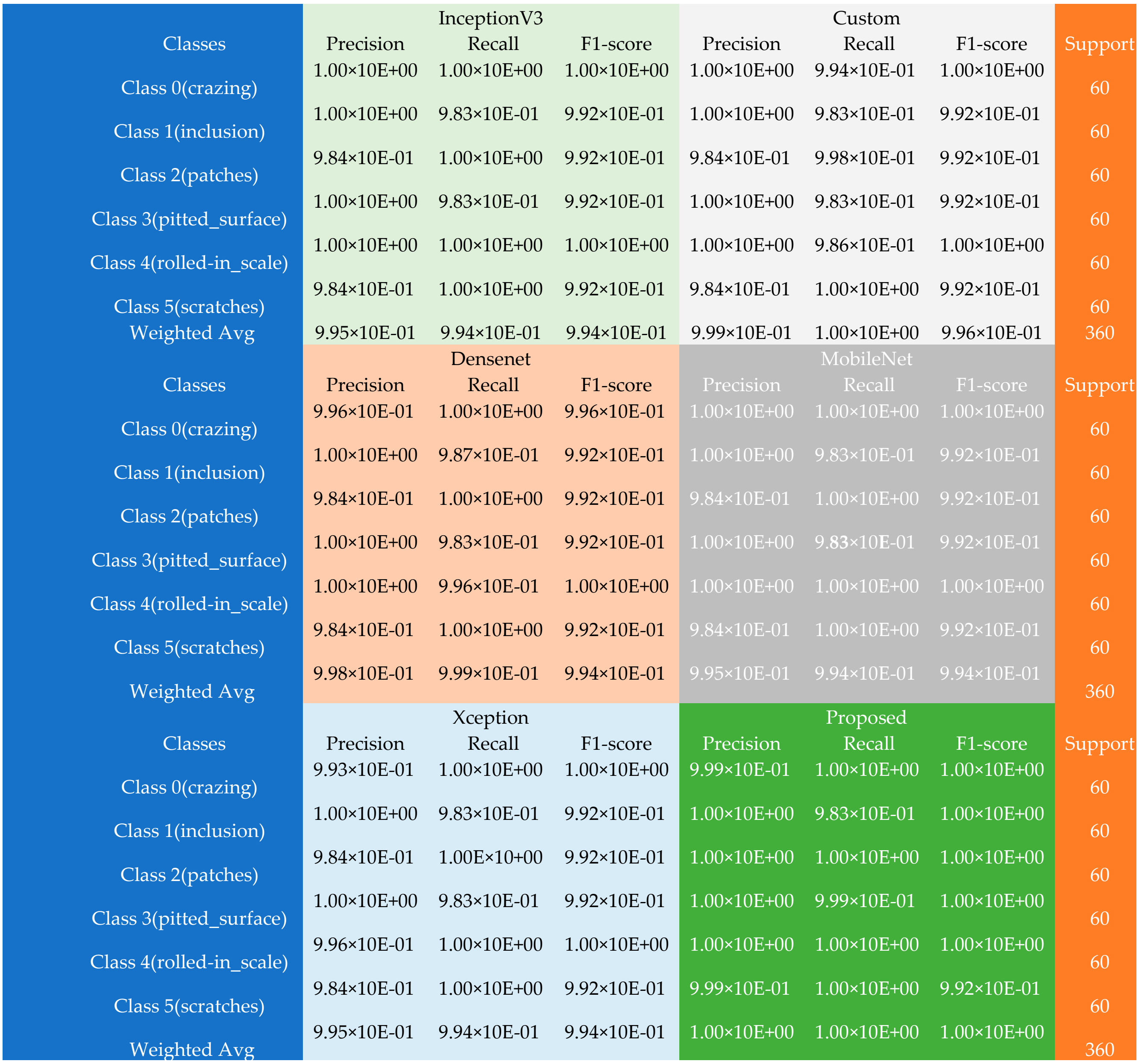
Table 5. The classification report for each of the models and the final model using the PCB dataset.

	Kp	MCC	Accuracy	MSE	MSLE
Custom	8.52×10^{-1}	7.83×10^{-1}	8.39×10^{-1}	2.76×10^{-1}	1.66×10^{-1}
Inceptionv3	9.44×10^{-1}	9.45×10^{-1}	9.72×10^{-1}	2.78×10^{-2}	1.34×10^{-2}
Xception	7.97×10^{-1}	8.11×10^{-1}	8.96×10^{-1}	4.72×10^{-1}	2.40×10^{-1}
DenseNet	7.78×10^{-1}	7.79×10^{-1}	8.89×10^{-1}	1.11×10^{-1}	5.34×10^{-2}
MobileNet	9.00×10^{-1}	9.03×10^{-1}	9.50×10^{-1}	5.00×10^{-2}	2.40×10^{-2}
Proposed	9.78×10^{-1}	9.78×10^{-1}	9.89×10^{-1}	1.11×10^{-2}	5.34×10^{-3}

As tabulated in Table 2, the classification performance significantly improved after the models' ensemble process with approximately 9.99×10^{-1} KP score, $1.00 \times 10^{+00}$ MCC, $1.00 \times 10^{+00}$ accuracy, and low MSE and MSLE of 3.47×10^{-4} and 3.48×10^{-6}, respectively. More vividly, the performance accuracy scores rose from 9.94×10^{-1} (Inceptionv3), 9.94×10^{-1} (both the Custom and DenseNet), and 9.93×10^{-1} (MobileNet) to $1.00 \times 10^{+00}$ for the proposed method. Detailed performance of the proposed method is shown using the confusion matrix table having the experiments' precision, recall, and F1-scores (see Figure 2).

The rows in the confusion matrix table correspond to the various classes of the NEU dataset, i.e., class 0 equals the crazing data samples, class 1 represents the inclusion samples, and class 2 represents the patched samples. The pitted_surface samples are denoted by class 3, while the rolled-in_scale samples are defined by class 4 and class 5, represented by the scratch samples. Additionally, the overall weighted average is found in the rows. According to the confusion matrix table, the InceptionV3 model returned close to 100% scores in precision, recall, and F1-score across the samples, with a weighted average precision of 9.95×10^{-1}, weighted average recall of 9.94×10^{-1}, and a weighted average F1-score of 9.94×10^{-1}. The scores were similar across the other participating models, but the proposed model yielded a superior performance, with a weighted average precision, recall, and F1 score of approximately 100%. Furthermore, the results obtained using the piston dataset are shown in Table 3.

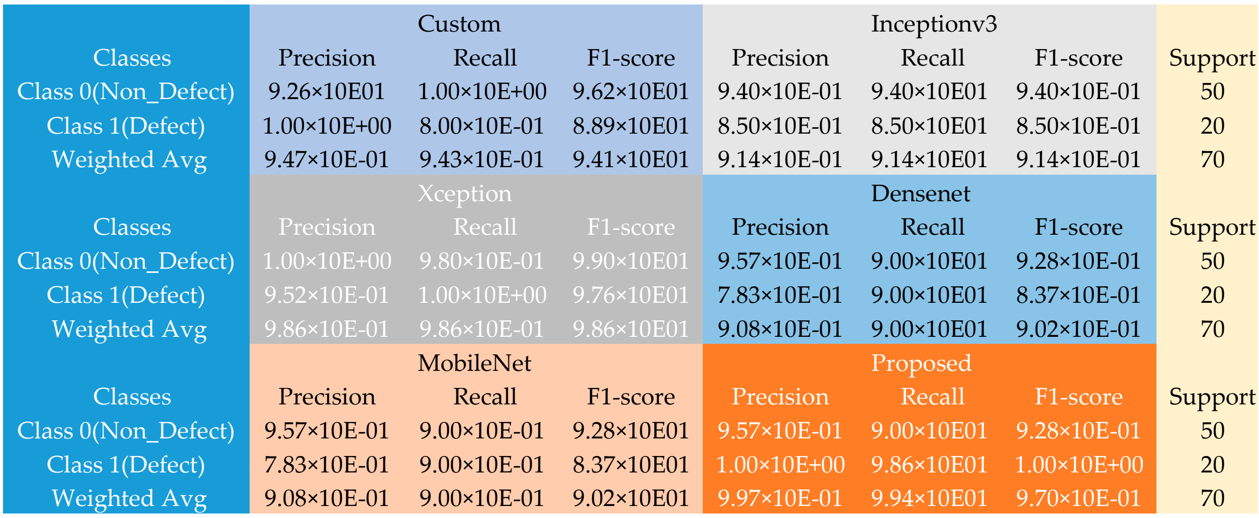
From Table 3, both the DenseNet and MobileNet recorded the least Kp, MCC, and accuracy scores. The Inceptionv3 followed this with 7.90×10^{-1} Kp, 7.90×10^{-1} MCC, and 9.14×10^{-1} accuracy scores, while the crafted custom model returned accuracy scores of 9.43×10^{-1}, 8.51×10^{-1} Kp, and 8.61×10^{-1} MCC. In continuation, the Xception model produced the second-best performance, with 9.66×10^{-1} Kp, 9.66×10^{-1} MCC, and 9.14×10^{-1} accuracy. Meanwhile, the proposed model produced the overall best performance, with 9.80×10^{-1} Kp, 9.72×10^{-1} MCC, and 9.49×10^{-1} accuracy, showing a remarkable performance accuracy gain over the various participating models.



	InceptionV3			Custom			
Classes	Precision	Recall	F1-score	Precision	Recall	F1-score	Support
Class 0(crazing)	1.00×10E+00	1.00×10E+00	1.00×10E+00	1.00×10E+00	9.94×10E-01	1.00×10E+00	60
Class 1(inclusion)	1.00×10E+00	9.83×10E-01	9.92×10E-01	1.00×10E+00	9.83×10E-01	9.92×10E-01	60
Class 2(patches)	9.84×10E-01	1.00×10E+00	9.92×10E-01	9.84×10E-01	9.98×10E-01	9.92×10E-01	60
Class 3(pitted_surface)	1.00×10E+00	9.83×10E-01	9.92×10E-01	1.00×10E+00	9.83×10E-01	9.92×10E-01	60
Class 4(rolled-in_scale)	1.00×10E+00	1.00×10E+00	1.00×10E+00	1.00×10E+00	9.86×10E-01	1.00×10E+00	60
Class 5(scratches)	9.84×10E-01	1.00×10E+00	9.92×10E-01	9.84×10E-01	1.00×10E+00	9.92×10E-01	60
Weighted Avg	9.95×10E-01	9.94×10E-01	9.94×10E-01	9.99×10E-01	1.00×10E+00	9.96×10E-01	360
	Densenet			MobileNet			
Classes	Precision	Recall	F1-score	Precision	Recall	F1-score	Support
Class 0(crazing)	9.96×10E-01	1.00×10E+00	9.96×10E-01	1.00×10E+00	1.00×10E+00	1.00×10E+00	60
Class 1(inclusion)	1.00×10E+00	9.87×10E-01	9.92×10E-01	1.00×10E+00	9.83×10E-01	9.92×10E-01	60
Class 2(patches)	9.84×10E-01	1.00×10E+00	9.92×10E-01	9.84×10E-01	1.00×10E+00	9.92×10E-01	60
Class 3(pitted_surface)	1.00×10E+00	9.83×10E-01	9.92×10E-01	1.00×10E+00	9.83×10E-01	9.92×10E-01	60
Class 4(rolled-in_scale)	1.00×10E+00	9.96×10E-01	1.00×10E+00	1.00×10E+00	1.00×10E+00	1.00×10E+00	60
Class 5(scratches)	9.84×10E-01	1.00×10E+00	9.92×10E-01	9.84×10E-01	1.00×10E+00	9.92×10E-01	60
Weighted Avg	9.98×10E-01	9.99×10E-01	9.94×10E-01	9.95×10E-01	9.94×10E-01	9.94×10E-01	360
	Xception			Proposed			
Classes	Precision	Recall	F1-score	Precision	Recall	F1-score	Support
Class 0(crazing)	9.93×10E-01	1.00×10E+00	1.00×10E+00	9.99×10E-01	1.00×10E+00	1.00×10E+00	60
Class 1(inclusion)	1.00×10E+00	9.83×10E-01	9.92×10E-01	1.00×10E+00	9.83×10E-01	1.00×10E+00	60
Class 2(patches)	9.84×10E-01	1.00E×10+00	9.92×10E-01	1.00×10E+00	1.00×10E+00	1.00×10E+00	60
Class 3(pitted_surface)	1.00×10E+00	9.83×10E-01	9.92×10E-01	1.00×10E+00	9.99×10E-01	1.00×10E+00	60
Class 4(rolled-in_scale)	9.96×10E-01	1.00×10E+00	1.00×10E+00	1.00×10E+00	1.00×10E+00	1.00×10E+00	60
Class 5(scratches)	9.84×10E-01	1.00×10E+00	9.92×10E-01	9.99×10E-01	1.00×10E+00	9.92×10E-01	60
Weighted Avg	9.95×10E-01	9.94×10E-01	9.94×10E-01	1.00×10E+00	1.00×10E+00	1.00×10E+00	360

Figure 2. The confusion matrix table using the NEU dataset.

Figure 3 shows more detailed results extracted from the various experiments conducted with the piston dataset. Given the 50 non-defect and 20 defect samples, the introduced method yielded a tremendous performance against the different pre-trained models and the typical model constructed during the experiments. The custom model produced weighted average precision, recall, and F1 scores of approximately 95% each across the 70 defect and non-defect test samples, and the Inceptionv3 model output weighted average precision, recall, and F1 scores of about 91% each; the Xception model generated weighted average precision, recall, and F1 scores of approximately 98.6% each. In contrast, the DenseNet and MobileNet produced weighted average precision, recall, and F1 scores of roughly 91% each across the 70 defect and non-defect test samples. In comparison, the proposed method outclassed these models by producing an improved weighted average precision of about 99.7%, 99.7% recall, and F1 scores of approximately 97% across the 70 defect and non-defect test samples.



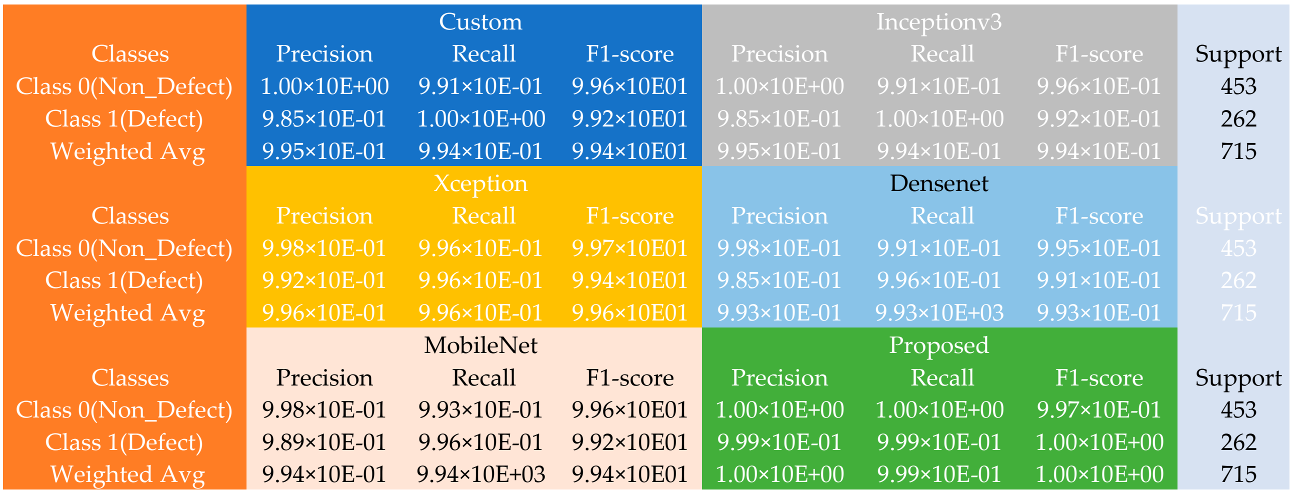
	Custom			Inceptionv3			
Classes	Precision	Recall	F1-score	Precision	Recall	F1-score	Support
Class 0(Non_Defect)	9.26×10E01	1.00×10E+00	9.62×10E01	9.40×10E-01	9.40×10E01	9.40×10E-01	50
Class 1(Defect)	1.00×10E+00	8.00×10E-01	8.89×10E01	8.50×10E-01	8.50×10E01	8.50×10E-01	20
Weighted Avg	9.47×10E-01	9.43×10E-01	9.41×10E01	9.14×10E-01	9.14×10E01	9.14×10E-01	70
	Xception			Densenet			
Classes	Precision	Recall	F1-score	Precision	Recall	F1-score	Support
Class 0(Non_Defect)	1.00×10E+00	9.80×10E-01	9.90×10E01	9.57×10E-01	9.00×10E01	9.28×10E-01	50
Class 1(Defect)	9.52×10E-01	1.00×10E+00	9.76×10E01	7.83×10E-01	9.00×10E01	8.37×10E-01	20
Weighted Avg	9.86×10E-01	9.86×10E-01	9.86×10E01	9.08×10E-01	9.00×10E01	9.02×10E-01	70
	MobileNet			Proposed			
Classes	Precision	Recall	F1-score	Precision	Recall	F1-score	Support
Class 0(Non_Defect)	9.57×10E-01	9.00×10E-01	9.28×10E01	9.57×10E-01	9.00×10E01	9.28×10E-01	50
Class 1(Defect)	7.83×10E-01	9.00×10E-01	8.37×10E01	1.00×10E+00	9.86×10E01	1.00×10E+00	20
Weighted Avg	9.08×10E-01	9.00×10E-01	9.02×10E01	9.97×10E-01	9.94×10E01	9.70×10E-01	70

Figure 3. The confusion matrix table using the piston dataset.

To continue demonstrating the robustness of the proposed method in learning and classifying the various kinds of defect and non-defect data samples in manufacturing products, another casting dataset was employed to train and test the introduced model. The results obtained from the experiments involving this dataset are shown in Table 4. According to the table, the presented method produced enhanced results against the other models, with Kp scores of 9.98×10^{-1}, $1.00 \times 10^{+00}$ MCC, and $1.00 \times 10^{+00}$ accuracy. It also returned the least MSE of 6.70×10^{-6} and MSLE of 7.80×10^{-8}, respectively.

Furthermore, the performance of the various models using the casting dataset is further showcased using the matrix table in Figure 4. Given the 453 non-defect and 262 defect samples, the proposed model returned an incredible performance improvement against the various pre-trained models and the custom model constructed during the experiments. The custom model produced weighted average precision, recall, and F1 scores of approximately 9.95×10^{-1} each across the 714 faulty and non-faulty test samples, with the Inceptionv3 model returning weighted average precision, recall, and F1 scores closely similar to the typical model. The Xception model, on the other hand, produced weighted average precision, recall, and F1 scores of about 9.96×10^{-1} each. At the same time, both the DenseNet returned weighted average precision, recall, and F1 scores of approximately 9.93×10^{-1} each across the 714 defect and non-defect test samples. However, the MobileNet returned weighted average precision, recall, and F1 scores of approximately 9.95×10^{-1} each across the 714 faulty and non-faulty test samples, and finally, the introduced approach displayed improved performance by returning better weighted average precision and F1 scores of about $1.00 \times 10^{+00}$ and recall of approximately 9.99×10^{-1} across the 714 defect and non-defect test samples.

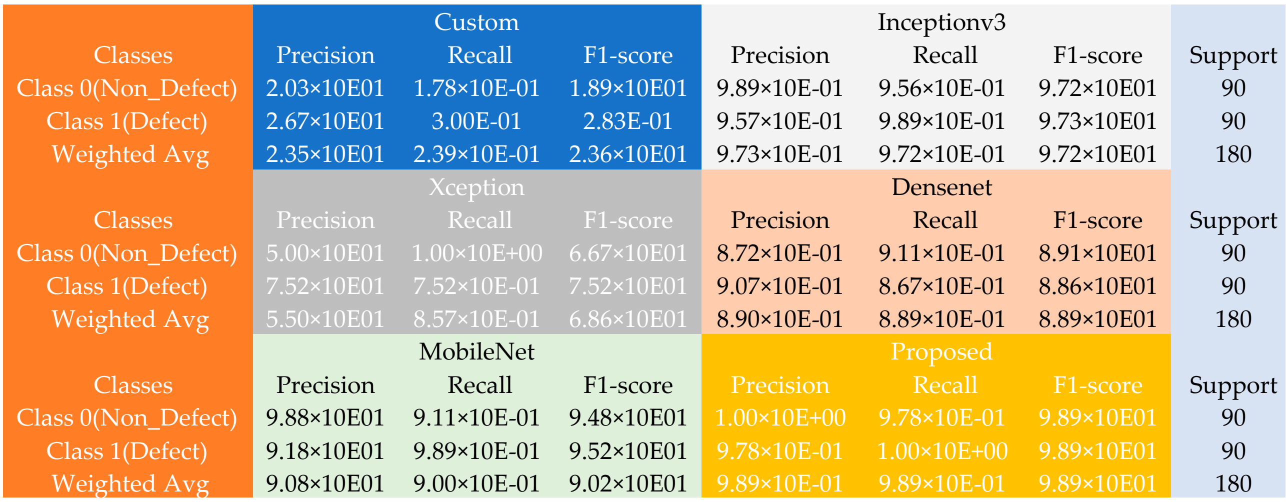
The PCB dataset was the final dataset used to train, test, and validate the proposed approach. As shown in Table 5, the proposed approach returned the enhanced results against the other models, with a Kp score of 9.78×10^{-1}, 9.78^{-1} MCC, and 9.89×10^{-1} accuracy. The introduced method also outputted the lowest possible MSE score of 1.11×10^{-2} and MSLE of 5.34×10^{-3}.



		Custom			Inceptionv3		
Classes	Precision	Recall	F1-score	Precision	Recall	F1-score	Support
Class 0(Non_Defect)	1.00×10E+00	9.91×10E-01	9.96×10E01	1.00×10E+00	9.91×10E-01	9.96×10E-01	453
Class 1(Defect)	9.85×10E-01	1.00×10E+00	9.92×10E01	9.85×10E-01	1.00×10E+00	9.92×10E-01	262
Weighted Avg	9.95×10E-01	9.94×10E-01	9.94×10E01	9.95×10E-01	9.94×10E-01	9.94×10E-01	715
		Xception			Densenet		
Classes	Precision	Recall	F1-score	Precision	Recall	F1-score	Support
Class 0(Non_Defect)	9.98×10E-01	9.96×10E-01	9.97×10E01	9.98×10E-01	9.91×10E-01	9.95×10E-01	453
Class 1(Defect)	9.92×10E-01	9.96×10E-01	9.94×10E01	9.85×10E-01	9.96×10E-01	9.91×10E-01	262
Weighted Avg	9.96×10E-01	9.96×10E-01	9.96×10E01	9.93×10E-01	9.93×10E+03	9.93×10E-01	715
		MobileNet			Proposed		
Classes	Precision	Recall	F1-score	Precision	Recall	F1-score	Support
Class 0(Non_Defect)	9.98×10E-01	9.93×10E-01	9.96×10E01	1.00×10E+00	1.00×10E+00	9.97×10E-01	453
Class 1(Defect)	9.89×10E-01	9.96×10E-01	9.92×10E01	9.99×10E-01	9.99×10E-01	1.00×10E+00	262
Weighted Avg	9.94×10E-01	9.94×10E+03	9.94×10E01	1.00×10E+00	9.99×10E-01	1.00×10E+00	715

Figure 4. The confusion matrix table using the casting dataset.

Finally, the performance of the different models using the PCB dataset was further demonstrated using the matrix table in Figure 5. With the 90 non-defect and 90 defect samples, the introduced model produced a remarkable performance improvement against the other pre-trained models and the unique conventional model constructed during the investigations. The custom model generated the lowest weighted average precision, recall, and F1 scores of approximately 2.35×10^{-1}, 2.39×10^{-1}, and 2.36×10^{-1}, respectively, across the 180 faulty and non-faulty PCB test samples. The Inceptionv3 model returned weighted average precision, recall, and F1 scores of about 9.73×10^{-1}, 9.72×10^{-1}, and 9.72×10^{-1}, respectively, while the Xception model produced weighted average precision, recall, and F1 scores of approximately 5.50×10^{-1}, 8.57×10^{-1}, and 6.86×10^{-1} respectively. Additionally, the DenseNet returned weighted average precision, recall, and F1 scores of approximately 8.90×10^{-1} each across the 180 defect and non-defect PCB test samples. In contrast, the MobileNet returned weighted average precision of 9.08×10^{-1}, 9.00×10^{-1} recall, and 9.02×10^{-1} F1 scores. Finally, the proposed approach demonstrated an improved performance by producing superior weighted average precision, recall, and F1 scores of about 9.89×10^{-1} each across the 180 defect and non-defect PCB test samples.



		Custom			Inceptionv3		
Classes	Precision	Recall	F1-score	Precision	Recall	F1-score	Support
Class 0(Non_Defect)	2.03×10E01	1.78×10E-01	1.89×10E01	9.89×10E-01	9.56×10E-01	9.72×10E01	90
Class 1(Defect)	2.67×10E01	3.00E-01	2.83E-01	9.57×10E-01	9.89×10E-01	9.73×10E01	90
Weighted Avg	2.35×10E01	2.39×10E-01	2.36×10E01	9.73×10E-01	9.72×10E-01	9.72×10E01	180
		Xception			Densenet		
Classes	Precision	Recall	F1-score	Precision	Recall	F1-score	Support
Class 0(Non_Defect)	5.00×10E01	1.00×10E+00	6.67×10E01	8.72×10E-01	9.11×10E-01	8.91×10E01	90
Class 1(Defect)	7.52×10E01	7.52×10E-01	7.52×10E01	9.07×10E-01	8.67×10E-01	8.86×10E01	90
Weighted Avg	5.50×10E01	8.57×10E-01	6.86×10E01	8.90×10E-01	8.89×10E-01	8.89×10E01	180
		MobileNet			Proposed		
Classes	Precision	Recall	F1-score	Precision	Recall	F1-score	Support
Class 0(Non_Defect)	9.88×10E01	9.11×10E-01	9.48×10E01	1.00×10E+00	9.78×10E-01	9.89×10E01	90
Class 1(Defect)	9.18×10E01	9.89×10E-01	9.52×10E01	9.78×10E-01	1.00×10E+00	9.89×10E01	90
Weighted Avg	9.08×10E01	9.00×10E-01	9.02×10E01	9.89×10E-01	9.89×10E-01	9.89×10E01	180

Figure 5. The confusion matrix table using the PCB dataset.

We also compared our results with other similar studies in the literature. Our model outperformed ShuffleDefectNet [38], which used the ShuffleNet to detect metallic surface defects on the Northeastern University (NEU) dataset. While their method achieved a mean average accuracy of 99.75%, our proposed method returned an approximately mean weighted average of 100%. The introduced method in this work also outclassed the approach that combined a modified AlexNet architecture and support vector machine algorithm to classify the steel strip defect NEU dataset that yielded 99.7% accuracy [39]. For the classification of faulty and non-faulty samples of the PCB dataset, our method also outperformed the 98.79% accuracy score recorded by Adibhatla et al. [40], 97.5% by Khalilian et al. [41], 98% score by Kim et al. [42], and 98.1% by Bhattacharya and Cloutier [43] (see Figure 6).

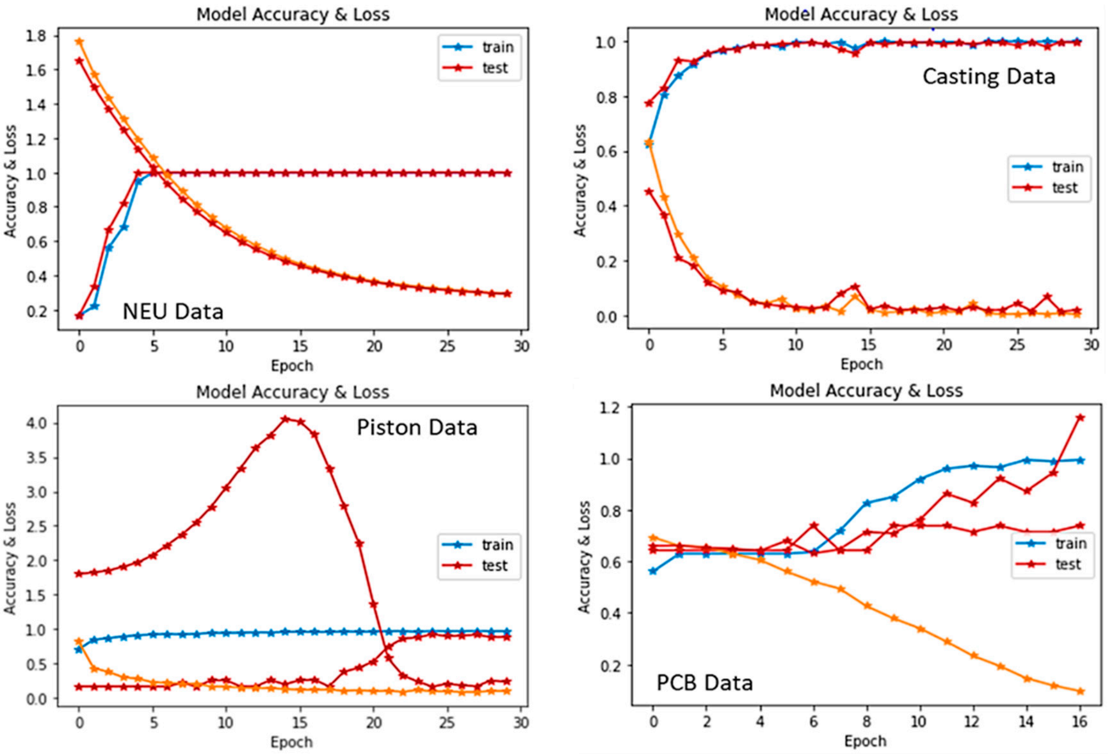


Figure 6. The output summary of the model training and testing process with the datasets.

Additionally, our introduced method performed better than the 98.06% accuracy score obtained by [44,45] and the 99% accuracy score recorded by Tang et al. [46] in the defect and non-defect piston data classification. Our result also surpassed the 95.5% accuracy score obtained by Lin et al. [47], with a 99.9% accuracy, 9.98×10^{-1} Kp score, and $1.00 \times 10^{+00}$ MCC score, respectively, for the identification and separation of the defect and non-defect submersible pump impeller casting data samples. Hence, it can be inferred that our proposed method can achieve high accuracy in industrial implementation with high robustness in different defect and non-defect types.

6. Conclusions

This paper introduced an ensembled deep learning model for accurately and rapidly identifying and classifying defects and non-defects from manufactured industrial products. During the experiments in the study, different DL models were trained to individually learn the vital features necessary for distinguishing the faulty industrial products from non-faulty ones and then unifying them using the convolution LSTM sequence-based ensemler to obtain high accuracy and near-optimal models for inferencing and implementation. Different matrices were employed to test and validate the proposed model, with remarkable results obtained to support the usefulness of the new method.

The proposed model yielded a superior performance because of the fine-tuning process of the different parameters in the different adopted models used in the experiment with the specific datasets and the fusion of the features from the various participating models to overcome the drawbacks of the individual models. The results from the multiple experiments showed that the adopted DL networks yielded good generalization on the NEU dataset, and they are highly adaptable for transfer into different domains. The same characteristics exhibited on the NEU dataset were also repeated on the piston dataset by the various adopted models. However, the Xception and the proposed model returned better performance than the custom, Inceptionv3, DenseNet, and MobileNet on the casting dataset. On the other hand, the Inceptionv3, MobileNet, and the introduced model performed better than the custom, DenseNet, and Xception DL networks on the PCB data samples.

The ensembled architectures offered the proposed model great assistance in aggregating the fine features from the different classes of the datasets used in the experiments and, in turn, returned better results. The proposed model extracted the defect and non-defect features from the data samples pertinent to learning to distinguish faulty and non-faulty manufactured products. Our introduced method offers a general ensembling architecture for learning more deep feature representations from different and diverse datasets for robustness and better performance.

Author Contributions: Conceptualization, O.S.; methodology, O.S.; software, O.S.; formal analysis, O.S. and M.N.; data curation, O.S.; writing—original draft preparation, O.S.; writing—review and editing, O.S., S.M., and M.N.; visualization, O.S.; supervision, S.M. and M.N. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding and the APC was funded by the Auckland University of Technology.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: (1) NEU (Northeastern University) surface defect dataset [34]. (2) PCB defect dataset [36]. (3) https://www.kaggle.com/datasets/ravirajsinh45/real-life-industrial-dataset-of-casting-product, accessed on 12 August 2022. (4) https://www.kaggle.com/datasets/satishpaladi11/mechanic-component-images-normal-defected, accessed on 12 August 2022.

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

References

1. Anand, S.; Priya, L. *A Guide for Machine Vision in Quality Control*; CRC Press: Boca Raton, FL, USA, 2019.
2. Goetsch, D.L.; Davis, S.B. *Quality Management for Organizational Excellence*; Pearson: Upper Saddle River, NJ, USA, 2014.
3. Naranjo-Torres, J.; Mora, M.; Hernández-García, R.; Barrientos, R.J.; Fredes, C.; Valenzuela, A. A review of convolutional neural network applied to fruit image processing. *Appl. Sci.* **2020**, *10*, 3443. [CrossRef]
4. Czimmermann, T.; Ciuti, G.; Milazzo, M.; Chiurazzi, M.; Roccella, S.; Oddo, C.M.; Dario, P. Visual-based defect detection and classification approaches for industrial applications—A survey. *Sensors* **2020**, *20*, 1459. [CrossRef] [PubMed]
5. Alom, M.Z.; Taha, T.M.; Yakopcic, C.; Westberg, S.; Sidike, P.; Nasrin, M.S.; Van Esesn, B.C.; Awwal, A.A.S.; Asari, V.K. The history began from alexnet: A comprehensive survey on deep learning approaches. *arXiv* **2018**, arXiv:1803.01164.
6. He, Y.; Song, K.; Meng, Q.; Yan, Y. An end-to-end steel surface defect detection approach via fusing multiple hierarchical features. *IEEE Trans. Instrum. Meas.* **2019**, *69*, 1493–1504. [CrossRef]
7. Borji, A.; Cheng, M.-M.; Jiang, H.; Li, J. Salient object detection: A benchmark. *IEEE Trans. Image Process.* **2015**, *24*, 5706–5722. [CrossRef]
8. LeCun, Y. LeNet-5, Convolutional Neural Networks. 2015. Available online: http://yann.lecun.com/exdb/lenet (accessed on 12 August 2022).
9. Tao, X.; Wang, Z.; Zhang, Z.; Zhang, D.; Xu, D.; Gong, X.; Zhang, L. Wire defect recognition of spring-wire socket using multitask convolutional neural networks. *IEEE Trans. Compon. Packag. Manuf. Technol.* **2018**, *8*, 689–698. [CrossRef]
10. Bartler, A.; Mauch, L.; Yang, B.; Reuter, M.; Stoicescu, L. Automated detection of solar cell defects with deep learning. In Proceedings of the 2018 26th European Signal Processing Conference (EUSIPCO), Roma, Italy, 3–7 September 2018; pp. 2035–2039.
11. Mundt, M.; Majumder, S.; Murali, S.; Panetsos, P.; Ramesh, V. Meta-learning convolutional neural architectures for multi-target concrete defect classification with the concrete defect bridge image dataset. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–17 June 2019; pp. 11196–11205.
12. Stephen, O.; Maduh, U.J.; Sain, M. A Machine Learning Method for Detection of Surface Defects on Ceramic Tiles Using Convolutional Neural Networks. *Electronics* **2021**, *11*, 55. [CrossRef]

13. Aydin, I.; Akin, E.; Karakose, M. Defect classification based on deep features for railway tracks in sustainable transportation. *Appl. Soft Comput.* **2021**, *111*, 107706. [CrossRef]
14. Krummenacher, G.; Ong, C.S.; Koller, S.; Kobayashi, S.; Buhmann, J.M. Wheel defect detection with machine learning. *IEEE Trans. Intell. Transp. Syst.* **2017**, *19*, 1176–1187. [CrossRef]
15. Zheng, Z.; Shen, J.; Shao, Y.; Zhang, J.; Tian, C.; Yu, B.; Zhang, Y. Tire defect classification using a deep convolutional sparse-coding network. *Meas. Sci. Technol.* **2021**, *32*, 055401. [CrossRef]
16. Ajmi, C.; Zapata, J.; Elferchichi, S.; Zaafouri, A.; Laabidi, K. Deep learning technology for weld defects classification based on transfer learning and activation features. *Adv. Mater. Sci. Eng.* **2020**, *2020*, 1574350. [CrossRef]
17. Konovalenko, I.; Maruschak, P.; Brezinová, J.; Viňáš, J.; Brezina, J. Steel surface defect classification using deep residual neural network. *Metals* **2020**, *10*, 846. [CrossRef]
18. Luo, Q.; Sun, Y.; Li, P.; Simpson, O.; Tian, L.; He, Y. Generalized completed local binary patterns for time-efficient steel surface defect classification. *IEEE Trans. Instrum. Meas.* **2018**, *68*, 667–679. [CrossRef]
19. Hu, H.; Liu, Y.; Liu, M.; Nie, L. Surface defect classification in large-scale strip steel image collection via hybrid chromosome genetic algorithm. *Neurocomputing* **2016**, *181*, 86–95. [CrossRef]
20. Deng, Y.-S.; Luo, A.-C.; Dai, M.-J. Building an automatic defect verification system using deep neural network for pcb defect classification. In Proceedings of the 2018 4th International Conference on Frontiers of Signal Processing (ICFSP), Poitiers, France, 24–27 September 2018; pp. 145–149.
21. Zhang, L.; Jin, Y.; Yang, X.; Li, X.; Duan, X.; Sun, Y.; Liu, H. Convolutional neural network-based multi-label classification of PCB defects. *J. Eng.* **2018**, *2018*, 1612–1616. [CrossRef]
22. Chen, Z.; Zhao, F.; Zhou, J.; Huang, P.; Song, W. A novel approach applied to fault diagnosis for micro-defects on piston throat. *Measurement* **2021**, *173*, 108508. [CrossRef]
23. Zheng, B.; Wang, C.; Qing, S. Piston Surface Defect Recognition Method Based on Image Processing. In Proceedings of the 2022 IEEE 10th Joint International Information Technology and Artificial Intelligence Conference (ITAIC), Chongqing, China, 17–19 June 2022; pp. 928–932.
24. Nikolić, F.; Štajduhar, I.; Čanađija, M. Casting Defects Detection in Aluminum Alloys Using Deep Learning: A Classification Approach. *Int. J. Met.* **2022**. [CrossRef]
25. Habibpour, M.; Gharoun, H.; Tajally, A.; Shamsi, A.; Asgharnezhad, H.; Khosravi, A.; Nahavandi, S. An Uncertainty-Aware Deep Learning Framework for Defect Detection in Casting Products. *arXiv* **2021**, arXiv:2107.11643. [CrossRef]
26. Szegedy, C.; Vanhoucke, V.; Ioffe, S.; Shlens, J.; Wojna, Z. Rethinking the inception architecture for computer vision. In Proceedings of the IEEE Conference on Computer Vision And Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 2818–2826.
27. Huang, G.; Liu, Z.; Van Der Maaten, L.; Weinberger, K.Q. Densely connected convolutional networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 4700–4708.
28. Chollet, F. Xception: Deep learning with depthwise separable convolutions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 1251–1258.
29. Howard, A.G.; Zhu, M.; Chen, B.; Kalenichenko, D.; Wang, W.; Weyand, T.; Andreetto, M.; Adam, H. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv* **2017**, arXiv:1704.04861.
30. Li, Y.; Huang, H.; Xie, Q.; Yao, L.; Chen, Q. Research on a surface defect detection algorithm based on MobileNet-SSD. *Appl. Sci.* **2018**, *8*, 1678. [CrossRef]
31. Rabano, S.L.; Cabatuan, M.K.; Sybingco, E.; Dadios, E.P.; Calilung, E.J. Common garbage classification using mobilenet. In Proceedings of the 2018 IEEE 10th International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment and Management (HNICEM), Baguio City, Philippines, 29 November–2 December 2018; pp. 1–4.
32. Michele, A.; Colin, V.; Santika, D.D. Mobilenet convolutional neural networks and support vector machines for palmprint recognition. *Procedia Comput. Sci.* **2019**, *157*, 110–117. [CrossRef]
33. Shi, X.; Chen, Z.; Wang, H.; Yeung, D.-Y.; Wong, W.-K.; Woo, W.-C. Convolutional LSTM network: A machine learning approach for precipitation nowcasting. *Adv. Neural Inf. Process. Syst.* **2015**, arXiv:1506.04214.
34. Bao, Y.; Song, K.; Liu, J.; Wang, Y.; Yan, Y.; Yu, H.; Li, X. Triplet-graph reasoning network for few-shot metal generic surface defect segmentation. *IEEE Trans. Instrum. Meas.* **2021**, *70*, 1–11. [CrossRef]
35. Dabhi, R. Casting Product Image Data for Quality Inspection. Available online: https://www.kaggle.com/datasets/ravirajsinh45/real-life-industrial-dataset-of-casting-product (accessed on 12 August 2022).
36. Tang, S.; He, F.; Huang, X.; Yang, J. Online PCB defect detector on a new PCB defect dataset. *arXiv* **2019**, arXiv:1902.06197.
37. Paladi, S. Mechanic Component Images (Normal/Defected) Dataset. Available online: https://www.kaggle.com/datasets/satishpaladi11/mechanic-component-images-normal-defected (accessed on 12 August 2022).
38. Anvar, A.; Cho, Y.I. Automatic metallic surface defect detection using shuffledefectnet. *J. Korea Soc. Comput. Inf.* **2020**, *25*, 19–26.
39. Boudiaf, A.; Benlahmidi, S.; Harrar, K.; Zaghdoudi, R. Classification of surface defects on steel strip images using convolution neural network and support vector machine. *J. Fail. Anal. Prev.* **2022**, *22*, 531–541. [CrossRef]
40. Adibhatla, V.A.; Chih, H.-C.; Hsu, C.-C.; Cheng, J.; Abbod, M.F.; Shieh, J.-S. Defect detection in printed circuit boards using you-only-look-once convolutional neural networks. *Electronics* **2020**, *9*, 1547. [CrossRef]

41. Khalilian, S.; Hallaj, Y.; Balouchestani, A.; Karshenas, H.; Mohammadi, A. Pcb defect detection using denoising convolutional autoencoders. In Proceedings of the 2020 International Conference on Machine Vision and Image Processing (MVIP), Qom, Iran, 18–20 February 2020; pp. 1–5.
42. Kim, J.; Ko, J.; Choi, H.; Kim, H. Printed circuit board defect detection using deep learning via a skip-connected convolutional autoencoder. *Sensors* **2021**, *21*, 4968. [CrossRef]
43. Bhattacharya, A.; Cloutier, S.G. End-to-end deep learning framework for printed circuit board manufacturing defect classification. *Sci. Rep.* **2022**, *12*, 12559. [CrossRef]
44. Zhu, Y.; Li, G.; Wang, R.; Tang, S.; Su, H.; Cao, K. Intelligent fault diagnosis of hydraulic piston pump based on wavelet analysis and improved alexnet. *Sensors* **2021**, *21*, 549. [CrossRef]
45. Tang, S.; Zhu, Y.; Yuan, S.; Li, G. Intelligent diagnosis towards hydraulic axial piston pump using a novel integrated CNN model. *Sensors* **2020**, *20*, 7152. [CrossRef] [PubMed]
46. Tang, S.; Zhu, Y.; Yuan, S. An adaptive deep learning model towards fault diagnosis of hydraulic piston pump using pressure signal. *Eng. Fail. Anal.* **2022**, *138*, 106300. [CrossRef]
47. Lin, J.; Yao, Y.; Ma, L.; Wang, Y. Detection of a casting defect tracked by deep convolution neural network. *Int. J. Adv. Manuf. Technol.* **2018**, *97*, 573–581. [CrossRef]