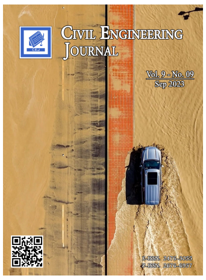
Available online at www.CivileJournal.org

Civil Engineering Journal

(E-ISSN: 2476-3055; ISSN: 2676-6957)

Vol. 9, No. 09, September, 2023



Convolutional Neural Network for Predicting Failure Type in Concrete Cylinders During Compression Testing

Jose Manuel Palomino Ojeda [1], Billy Alexis Cayatopa-Calderón [2], Lenin Quiñones Huatangari [1], Jose Luís Piedra Tineo [2], Manuel Emilio Milla Pino [3*], Wilmer Rojas Pintado [2]

[1] Research Institute of Data Science, National University of Jaen, Jaen 06800, Cajamarca, Peru.

[2] Seismological and Construction Research Institute, National University of Jaen, Jaen 06800, Cajamarca, Peru.

[3] Faculty of Engineering, National University of Jaen, Jaen 06800, Cajamarca, Peru.

Received 03 June 2023; Revised 20 August 2023; Accepted 23 August 2023; Published 01 September 2023

Abstract

Cracks in concrete cause structural damage, and it is important to identify and classify them. The objective of the research was to describe the behavior and predict the type of failure in concrete cylinders using convolutional neural networks. The methodology consisted of creating a database of 2650 images of failure types in concrete cylinders tested in compression at the Laboratory of Testing and Strength of Materials of the National University of Jaen, Cajamarca, Peru. To identify cracks on the concrete surface, the database was divided into training (60%), validation (20%), and testing (20%), and a transfer learning approach was developed using the MobileNet, DenseNet121, ResNet50, and VGG16 algorithms from the Keras library, programmed in Python. To validate the performance of each model, the following indicators were used: recall, precision, and F1 score. The results show that the models studied correctly classified the type of failure in concrete with accuracies of 96, 91, 86, and 90%, with the MobileNet algorithm being the best predictor with 96%. The novelty of the study was the development of deep learning algorithms with different architectures that can be used in structural health assessment as an automated and reliable method compared to traditional ones. In addition, these trained algorithms can be used as source code in drones for structural monitoring.

Keywords: Concrete; Computer Vision; Deep Learning; Crack Detection; Image Processing.

1. Introduction

Concrete is a widely used construction material in the world due to its mechanical properties such as fire resistance, durability, and formability [1]. It is used in the construction of bridges, buildings, and roads due to its low cost and mechanical properties. However, it is susceptible to moisture, corrosion of reinforcement, high temperature, overloading, and differential settlement of soil, which causes cracks in its structure to appear in the stress zones [2, 3], which show deterioration and affect the safety, workability, and durability of concrete structures [4]. Research has deepened the study of its mechanical properties [5], being the compressive strength and the type of failure the most evaluated properties, through laboratory tests where the preparation of concrete cylinders of 150×300 mm (ASTM C-172) is performed to be evaluated in a hydraulic press (ASTM C-39).

Cracks are an important representation in structural evaluation because they provide information on the situational condition of concrete structures [6] and allow the quality of the materials used in the production of concrete to be

[*] Corresponding author: manuel.milla@unj.edu.pe

http://dx.doi.org/10.28991/CEJ-2023-09-09-01

assessed through crack analysis. They are classified by traditional methods, which tend to be subjective, error-prone, ineffective, unreliable [7-9], costly, time-consuming, and limited because they do not allow the total evaluation of the structure, as they cannot access dangerous areas [6, 10], since they are based on visual inspection and professional evaluation. The identification, classification, and analysis allow for the evaluation of the safety and durability of concrete, indicating when timely maintenance should be performed to avoid structural deterioration, which will guarantee the service life and reliability of concrete structures [11]. Its early detection and evaluation prevent further damage and ensure structural safety, thus avoiding possible accidents. Therefore, the speed and accuracy of classification algorithms constitute key elements for the detection of damage in concrete. Several methods use deep learning models that automate the process [5, 12], using inspection techniques based on computer vision for the classification of crack types [7].

Artificial Intelligence (AI) has permeated virtually all engineering applications due to recent advances in computer technology. Machine learning and deep learning are two major subfields of AI that incorporate and extend learning capabilities into data classification during the preprocessing phase [8]. Machine learning has been widely applied in civil engineering, particularly in structural engineering, for tasks such as predicting the shear strength of grouted reinforced concrete, calculating damage in structures, and identifying the seismic failure mode in reinforced concrete shear walls [9]. Deep learning identifies hierarchical patterns and features using a convolutional neural network (CNN) as an efficient and automated method to detect and classify cracks in concrete structures [13]. Deep learning-based models play a significant and successful role in the accurate characterization, classification, and detection of exposed cracks in complex environments [14].

Crack classification algorithms have been developed to improve the inspection and maintenance of civil infrastructure. Koch et al. [15] reviewed computer vision-based methods and classified them into three types: preprocessing (features), models (patterns), and 3D reconstruction, with learning and non-learning-based approaches. Most of the non-learning-based approaches use image processing techniques such as noise filtering, edge detection, binary thresholding, and morphological operations [16, 17], techniques that have limitations when dealing with situations with complicated backgrounds, noise, and visual disturbances such as shadows. Adhikari et al. [18] proposed a method that combines histogram enhancement, median filtering, and thresholding for crack detection, but this approach also faces challenges when the images lack high contrast. Other approaches, such as the histogram threshold-based method proposed by Dinh et al. [19], have also shown limitations in their ability to deal with high-contrast surface images. There is an ongoing need for more robust and accurate algorithms with transfer learning approaches that can address the challenges of crack detection in complex environments. The present research has addressed these gaps present in the literature by proposing different architectures of deep learning algorithms that automatically identify relevant features and patterns from the training data, which have been enhanced by data augmentation techniques that solve the limitations of contrast, noise, and complex backgrounds, allowing the algorithms to identify cracks under these scenarios.

The objective of the research was to use convolutional neural networks to classify crack types in concrete cores using deep learning algorithms: MobileNet, DenseNet121, ResNet50, and VGG16, configured with different architectures, for a more equitable and measurable approach in the automation of the crack identification and classification process as an alternative to the traditional method, which will improve accuracy and reduce evaluation costs.

The contribution of the research lies in the configuration of algorithms with different parameters, trained with a proprietary database of different types of failures, extracted from the testing and strength of materials laboratory, for the recognition of patterns and distinctive characteristics, in the classification of the type of concrete failure in six categories: Type I, II, III, IV, V and VI. In addition, the proposed classification has not been addressed in the previous research analyzed, since most of them focus on the classification into non-crack and crack. In this sense, this approach, based on the ASTM C-39 standard, allows us to accurately represent and extract more information on concrete failures, being fundamental as a parameter in the structural evaluation of infrastructures.

2. Literature Review

Several researchers have explored automated crack detection and classification methods using image-based machine learning, such as convolutional neural networks (CNNs). Lee et al. [20], used an autonomous deep-learning crack detection algorithm to support timely and proactive management of concrete structures with an accuracy of 83.40%. Ali et al. [21], proposed an algorithm using a two-branch CNN architecture to effectively discriminate relevant cracks and other components such as noise and edge-like image components in concrete surfaces with an accuracy of 99.70%. Ye et al. [22], used the STCNet I model, a fast detection network architecture using deep learning-based dilated convolution, to detect apparent cracks in concrete runways with an accuracy of 99.33%. Li et al. [23], use the ResNet model to automatically identify and locate cracks in concrete dam structures with an accuracy of 99.39%. Several authors use different CNN models, varying convolutional layers, parameters, and activation functions (see Table 1), to classify cracks in concrete.

Table 1. Convolutional neural network models for crack classification

Model	Convolutional layers	Parameters (million)	Activation function	Precision	Reference
CCA+CRA	10	N.A.	N.A.	83.40%	Lee et al. [20]
ResNet-50	4	8.40	ReLU	99.70%	Ali et al. [21]
STCNet I	18	1.59	LeakyReLU/Sigmoid	99.33%	Ye et al. [22]
ResNet	17	24.03	ReLU/Softmax	99.39%	Li et al. [23]
InceptionV3	48	23.9	N.A.	90.90%	Nguyen et al. [8]
SegCrack	4	28.44	GELU (Gaussian Error Linear Unit)	96.66%	Wang & Su [24]
RUC-Net	8	N.A.	RELU	88.33%	Yu et al. [25]

CNNs are powerful tools for learning and extracting features from image data, making it easy to identify subtle information that might otherwise be difficult to detect [26]. Implemented in place of computer vision algorithms to automatically and accurately detect and classify concrete damage images, they have shown great potential in image-based object detection tasks [27]. CNNs are composed of deep layers of operations, where layered filters automatically and hierarchically learn features from the raw data, known as deep learning (DL). DL systems benefit from fast computation, so implementations on graphics processing units (GPUs), cloud-based computing services, or fast processors have been standardized [28]. In the field of civil engineering, due to advances in digital image processing and machine learning algorithms, DL is increasingly used to build classification and failure detection models for different structural elements [26], allowing highly efficient monitoring by image classification algorithms by training on large databases of concrete failures [29].

Some of the main algorithms used and recommended by different researchers for the training and validation of the models were:

MobileNet: Developed by Google for small models with low power and low latency requirements. They are designed for tasks such as classification, detection, embedding, and segmentation. Uses a depth-separable convolution instead of a standard convolution to extract high-level features from the input data. This technique minimizes parameter size and computational latency, making it suitable for use on embedded platforms [30].

DenseNet121: DenseNet is a neural network architecture that uses dense connectivity and consists of a dense block, a transition layer, and a growth rate. The output of each dense block is concatenated into a single input tensor. DenseNet has four types of architectures with different numbers of dense blocks, and DenseNet121 is a special type with a scheme of (6, 12, 24, and 16) dense blocks. To reduce overfitting for tasks with smaller training sets, DenseNet121 uses global average clustering and softmax classifiers instead of fully connected layers [31].

ResNet50: This is a pre-trained neural network trained on ImageNet, a dataset of nearly 1.2 million images. The features and weights learned during this training can be used for other tasks through a process called transfer learning. Fine-tuning is a technique that allows the network to adapt to new tasks with different numbers of classes and categories, using fewer epochs than training from scratch [32].

VGG16: It is a very classical convolutional neural network model and its network structure is very simple. It significantly reduces the model parameters because the model layers are reduced at the same time. Therefore, the efficiency of crack detection is improved by extracting useful features with fewer parameters [33].

3. Materials and Methods

3.1. Methodology

The study was based on a supervised deep learning approach for predicting the type of failure in concrete. Figure 1 shows the methodological flowchart that guided the conduct of the study. A database was created with images of failure types collected from the laboratory; the data were divided into training, validation, and test sets; then CNNs were programmed in Google Colab with different architectures that were trained and validated with the dataset. Finally, the type of failures in the test set was predicted by evaluating the model through performance and accuracy with metrics such as precision, sensitivity, and specificity. A high-performance GPU was used to accelerate the training of CNN models, and the software included machine learning libraries, TensorFlow and Keras, to build and train CNN models.

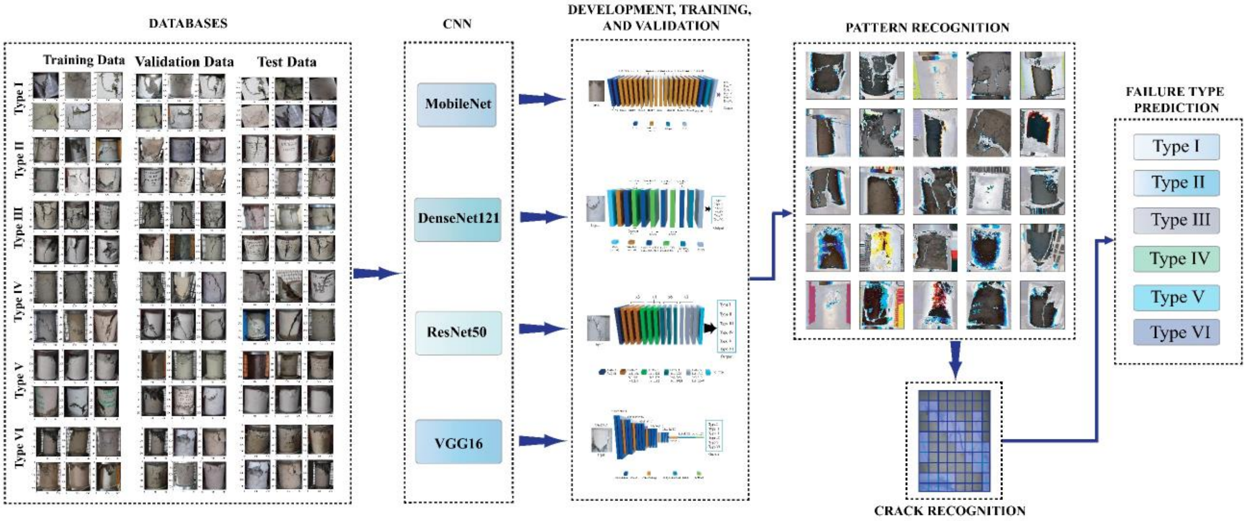


Figure 1. Methodology flowchart

3.2. Concrete Cylinders

The concrete cylinders were manufactured according to ASTM C-172, Figure 2-a shows a diameter of 150 mm and a height of 300 mm, and then tested according to ASTM C-39 using a hydraulic press that applies a gradually increasing load until the concrete cylinder breaks, thus obtaining the compressive strength. Figure 2-b shows the fracture, where a crack pattern can be observed that provides valuable information about the quality of the materials used in the mix. The cracks are classified into six types: Type I, II, III, IV, V, and VI. This test is used in the construction industry to evaluate the quality of the concrete used in a structure, as it provides information on the compressive strength of the material.

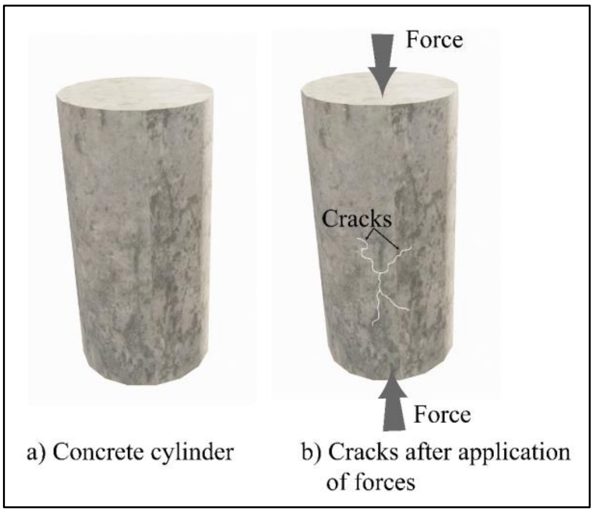


Figure 2. a-Concrete cylinder before and b- after application of the load

3.3. CNN

It employs convolutional kernels to perform multiple transformations on each layer. Unlike conventional neural networks, the layers of a CNN are not fully connected, but are sparsely connected, which is called local connection. This local connection takes advantage of the spatial correlation between the layers of the neural network, which is suitable for extracting relevant features in image data, speech, natural language text, etc. [34]. Figure 3 shows the structure of the CNN used to classify the types of failures in concrete, composed of the input layer, the image of the breakage of the concrete cylinder, and the hidden layers composed of the Convolutional layer, Pooling layer, and the output layer composed by the type of failure in the concrete cores. Each image is processed by the CNN using filters and transformations in its dimensions until the class (type of concrete failure) is determined.

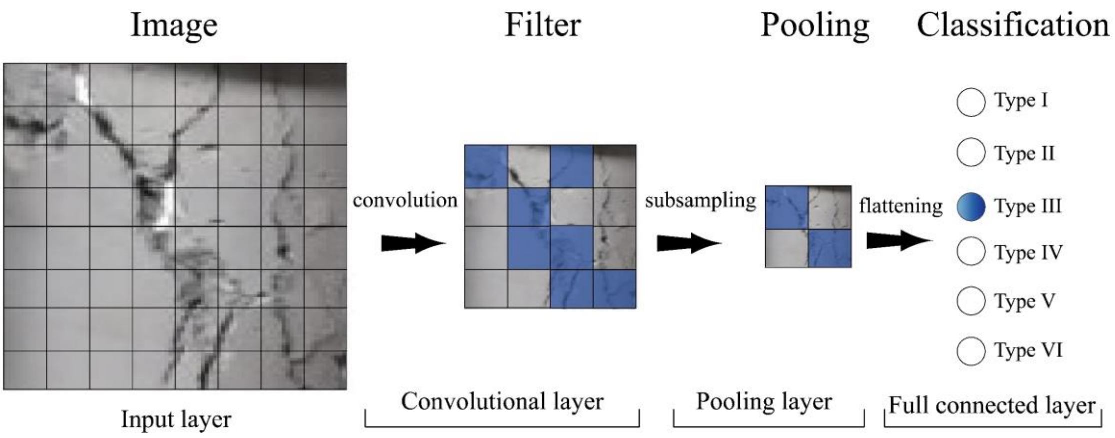


Figure 3. Convolutional neural network structure

3.4. Databases

The database was obtained from the Laboratory of Testing and Strength of Materials of the National University of Jaén, between the years 2019-2023, Figure 4 shows the images of cracks of concrete cylinders, of different types: Type I, Type II, Type III, Type VI, Type V and Type VI, with a total of 2650 images with a resolution of 1280×720 pixels and three color space channels (RGB). They were cropped into 224×224 pixel images with different surface finishes and illumination conditions, as this reduces the computational load on the model, accelerating convergence and crack classification, and allowing predictions greater than 90% [35]. In addition, data augmentation techniques were applied to improve the quality and quantity of training data to build an efficient model [36]. The techniques used included: flipping, cropping, rotation, and color space transformation, among others. The images were stored in a single file according to the type of defect in the Google Drive platform.

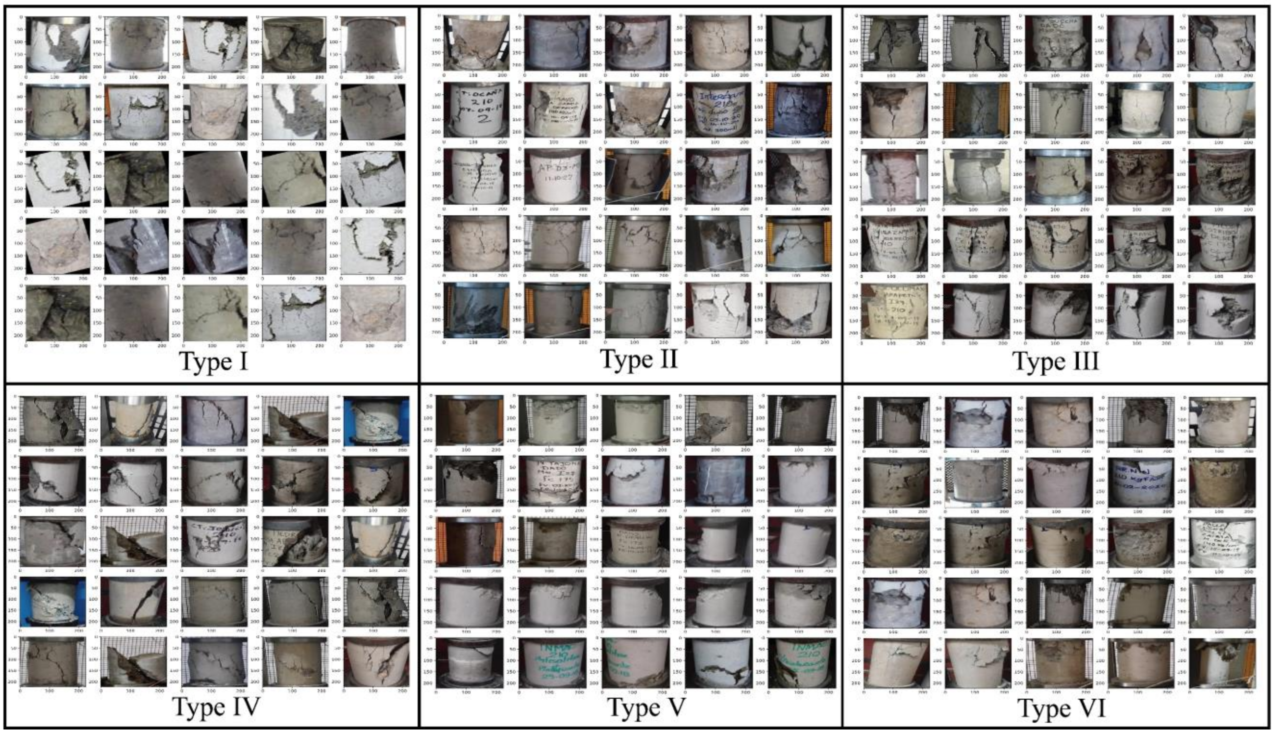


Figure 4. Failure type database

3.5. Development, Training, and Validation of CNN

The data matrix was constructed using the Python Keras library, a high-level convolutional neural network API that can run on top of the TensorFlow or Theano libraries. Using the Colab cloud computing interface with a high-performance NVIDIA Tesla V100 GPU and 26.9 GB of RAM on the Google Colab platform, these configurations allow for the massive parallel computing needed to train large models. The database was imported from Google Drive and the input parameters were configured for each convolutional neural network, then the dataset was created for training (60%), validation (20%), and testing (20%), and the models MobileNet, DenseNet 121, ResNet 50 and VGG16 were imported with loss = categorical_crossentropy, optimize = adam, metrics = accuracy and epochs = 50, the models were generated with the parameters shown in Table 2.

Table 2. Parameters of the different models

	MobileNet	DenseNet 121	ResNet 50	VGG16
Total params	8.25 M	17.07 M	36.45 M	139.59 M

Each CNN was configured as shown in Figures 5 to 8, representing different architectures and activation functions. Figure 5, shows the architecture of the MobileNet model [37], which consists of (a) an input layer that receives the image of the concrete fault type utilizing a 224×224 pixel matrix and color channel, (b) the information is processed in the depth-separable convolutional layers that perform a 1×1 convolution, (c) the bottleneck residual layers reduce the dimensionality of the input data before applying the convolutional layers, which reduces the number of channels and the amount of computation required, (d) the information is passed to the ave-pool layer by clustering points, where the average of the pixel clusters is calculated, (e) the fully connected (FC) layer with six neurons classifies the type of flaw in the concrete based on the features extracted from the previous layers.

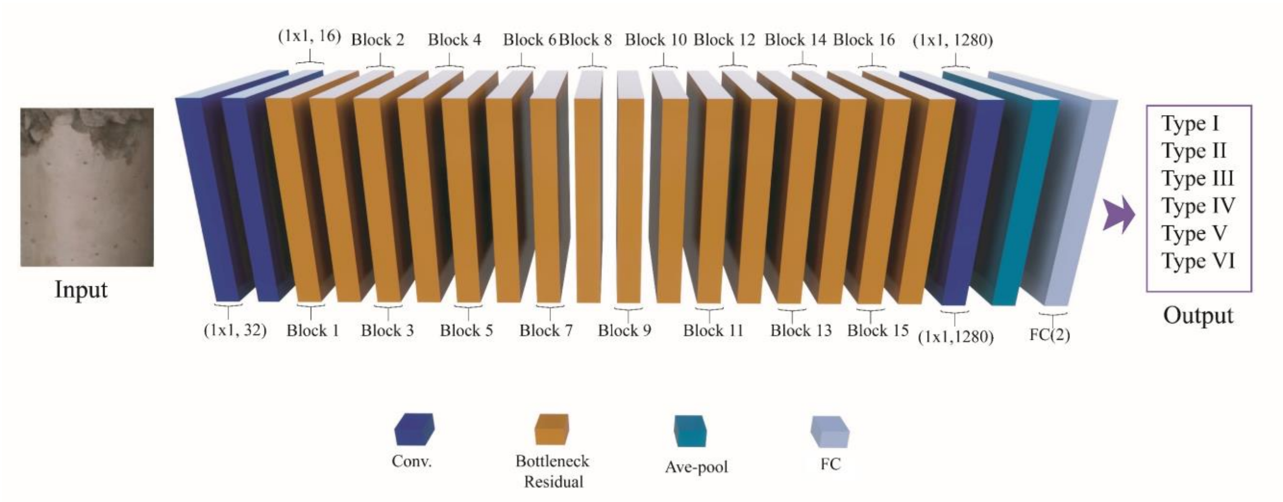


Figure 5. MobileNet model

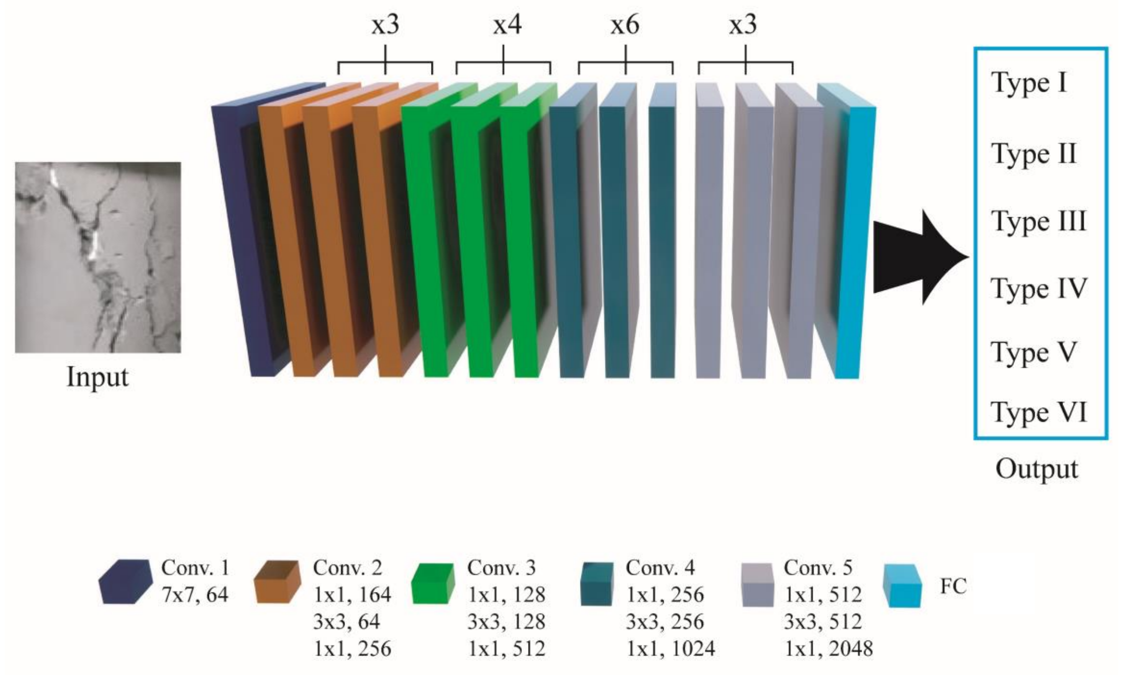


Figure 6. ResNet50 model

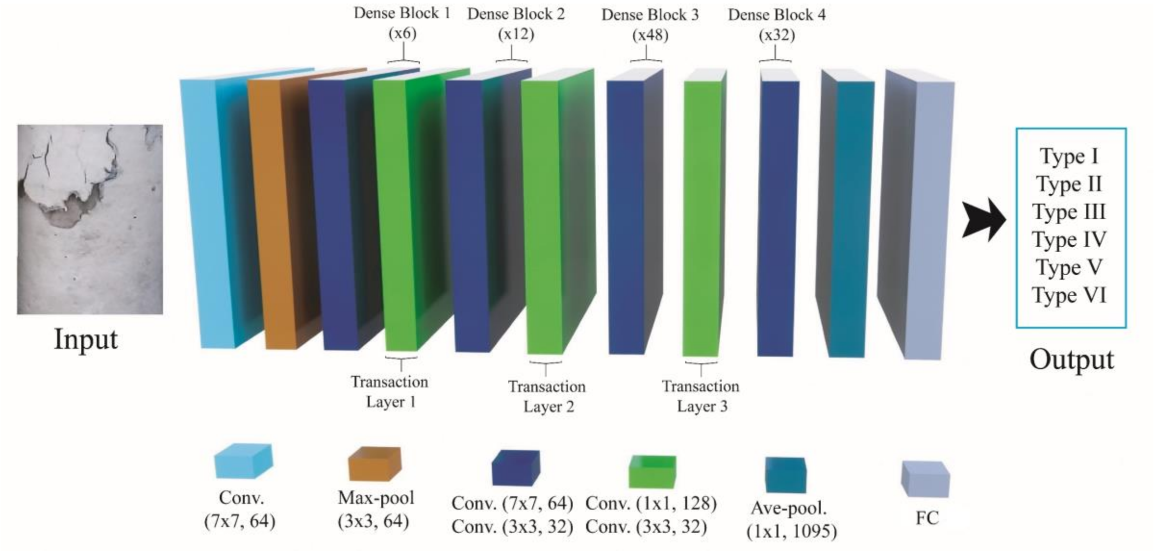


Figure 7. DenseNet121 model

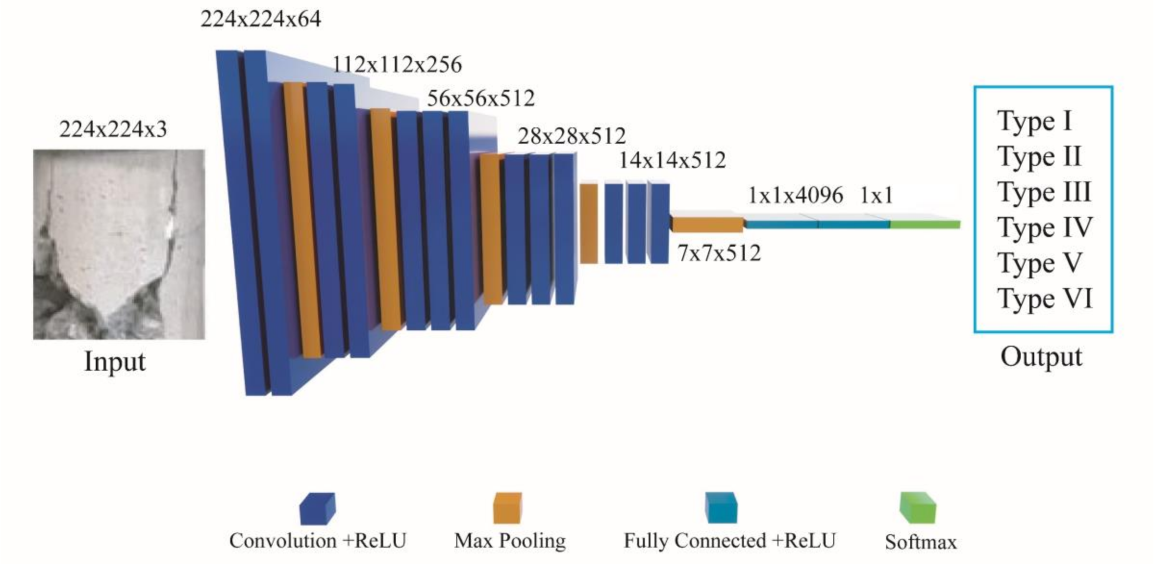


Figure 8. VGG16 model

Figure 6 shows the architecture of the ResNet50 model [38], which consists of (a) a convolutional layer with 64 filters of size 7×7 that processes the image of the failure type in concrete; (b) the information is passed to four residual blocks with similar structure but with different filters in each convolutional layer (conv. 2,3,4 and 5) of 1×1, 3×3 and 1×1 to improve the representation of the features and (c) the Fully Connected (FC) output layer classifies the failure type with the Softmax activation function, converting the outputs into probabilities of belonging to each class. The architecture is characterized by the use of residual blocks that allow maintaining the flow of information through the network, which contributes to better performance in image classification tasks.

Figure 7 illustrates the structure of the DenseNet121 model [39], which consists of (a) a convolutional layer (cov) that extracts the image information of the concrete defect type; (b) the information is passed to the max-pooling layer (max-pool) for the extraction and reduction of the initial features, (c) these are connected to the dense block layers 1, 2, 3 and 4, composed of several convolutional layers in series, where the output of each layer is concatenated with the input of all the following layers in the block, allowing a direct connection between all of them, which facilitates the access to the features extracted in previous stages, improving the efficiency and the information flow in the network, between these dense blocks are used (d) a transition convolutional layer. Then, the previous layer is connected to (e) the pooling layer (ave-pool), to reduce the number of features before advancing to the next block, thus controlling the size of the network and the computational complexity. After the dense blocks, connected layers are added to classify the defect type.

Figure 8 shows the structure of the VGG16 model [40], which consists of the following elements: (a) an input layer that receives the defect image in 224×224 pixels and RGB colors; followed by (b) 13 convolutional layers with 64 filters, each of size 3×3, and a ReLU activation function to extract relevant features. The information is then passed to the (c)

MaxPooling layer with a 2×2 window and a stride of 2, which aims to reduce the spatial dimensions. Next, (d) additional convolutional layers with 128, 256, and 512 filters are applied, followed by ReLU activation functions. (e) A penultimate MaxPooling layer with a 2×2 window and a stride of two further reduces the dimensions. Then three fully connected layers are added (f) A layer with 4096 neurons and a ReLU activation function. Finally, the previous layer is connected to the last layer with six neurons and a softmax activation function to classify the defect type.

3.6. CNN Performance Evaluation

To evaluate the effectiveness of the previously trained models in crack-type classification, several common performance indices were used for CNNs: (1) model overfitting, (2) validation accuracy, and (3), $F_{1-score}$.

$$\text{Precision} = \frac{TP}{TP + FP} \tag{1}$$

$$Re\,c\,all \ = \frac{TP}{TP + FN} \tag{2}$$

$$F_{1-score} \ = \frac{\text{2*Recall*Precision}}{\text{Recall + Precision}} \tag{3}$$

Where TP (True positive), TN (True negative), FN (False negative), and FP (False positive) represent different scenarios based on the actual and predicted classes of the images [15].

TP: represents the case where both the actual class and the predicted class are crack images, indicating correct crack detection in the images.

TN: This is the scenario in which both the actual class and the predicted class are background images, allowing correct identification of areas without cracks in the images.

FN: This is the situation where the actual class is a broken image, but the prediction incorrectly identifies it as a background image, resulting in missed crack detection.

FP: Represents the actual class of a background image, but the prediction incorrectly identifies it as a crack image, which means a false alarm or misidentification of non-cracked areas as cracks.

In the statistical analysis of binary classification, the F1 score serves as a measure of accuracy within a model, it combines two other performance indices, namely "precision" and "recall" (see Equation 3). The F1 score ranges from 0 (indicating an inability to classify correctly) to 1 (representing perfect classification). By combining accuracy and recall, the F1 score provides a comprehensive assessment of model accuracy [8]. Using these indicators, we evaluated the performance of the crack classification model quantitatively

4. Results

After generating from Python the models MobileNet, RestNet50, DenseNet121, and VGG16, configured with different architectures and input variables, using 2650 images, the failure types of 530 experimentally obtained fracture images were estimated and compared to validate the models. Figure 9 shows the training generation, obtaining the best results in terms of accuracy with the MobileNet model with 96% and an error of 4% in 50 epochs, capturing the different characteristics of the cracks in the concrete images. The other models presented a precession greater than 85%, being reliable to be used as a basis for the classification of other concrete pathologies.

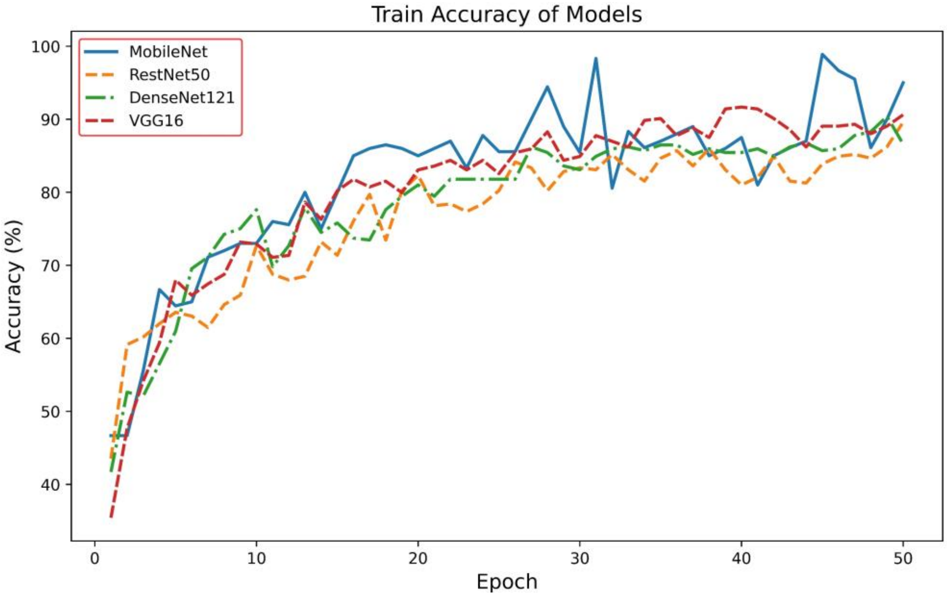


Figure 9. Accuracy of models during training

Figure 10 shows the validation of the models in predicting the type of failure, applied to the test dataset, obtaining 96% accuracy with the MobileNet model, which identifies patterns and characterizes each type with an error of 4%, the other models presented an error of less than 15%. To choose the number of epochs, the work was based on Qu et al. [33], taking into account the factors: the complexity of the data set, model architecture, and convergence of the training process, calculating in 50 the optimal number of epochs determined through experimentation and validation, to improve the performance of the models, reducing the computational time.

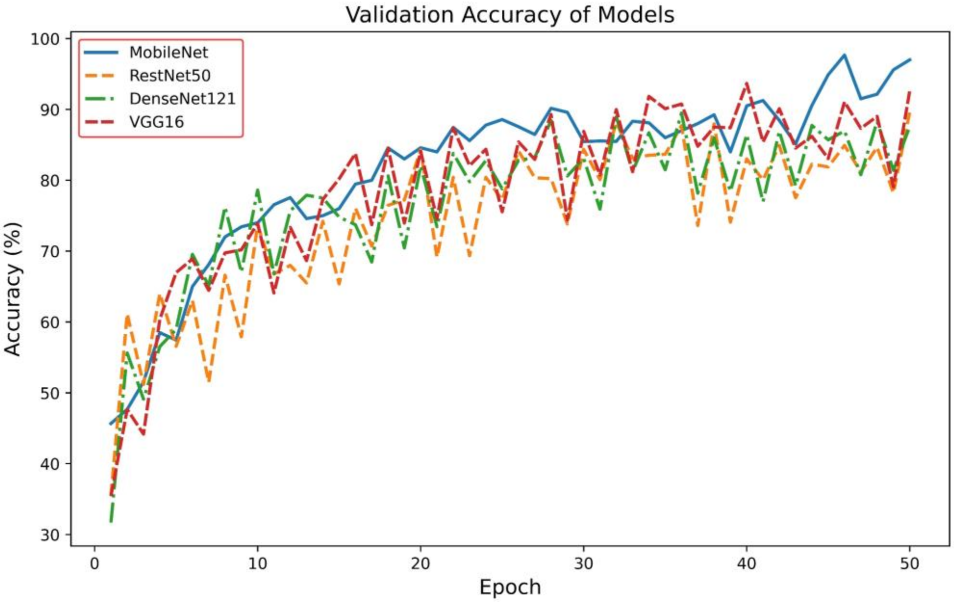


Figure 10. Model accuracy during validation

The MobileNet, RestNet50, DenseNet121, and VGG16 models and their statistics are shown in Table 3 to 6; being evaluated using the experimentally obtained database in terms of accuracy (see Equations 1 to 3)

Table 3. MobileNet model statistics

Model	Type	P (%)	R (%)	F1 (%)
MobileNet	I	89	89	89
	II	83	86	85
	III	98	97	98
	IV	91	96	93
	V	97	99	98
	VI	100	88	94

Table 4. RestNet50 model statistics

Model	Type	P (%)	R (%)	F1 (%)
RestNet50	I	81	81	81
	II	76	83	86
	III	98	91	95
	IV	86	86	86
	V	94	96	95
	VI	86	90	88

Table 5. DenseNet121 model statistics

Model	Type	P (%)	R (%)	F1 (%)
DenseNet121	I	70	70	70
	II	66	100	80
	III	94	78	85
	IV	65	70	67
	V	92	100	96
	VI	100	67	80

Table 6. VGG16 model statistics

Model	Type	P (%)	R (%)	F1 (%)
VGG16	I	75	78	76
	II	70	83	77
	III	91	92	92
	IV	85	80	82
	V	97	97	97
	VI	89	80	84

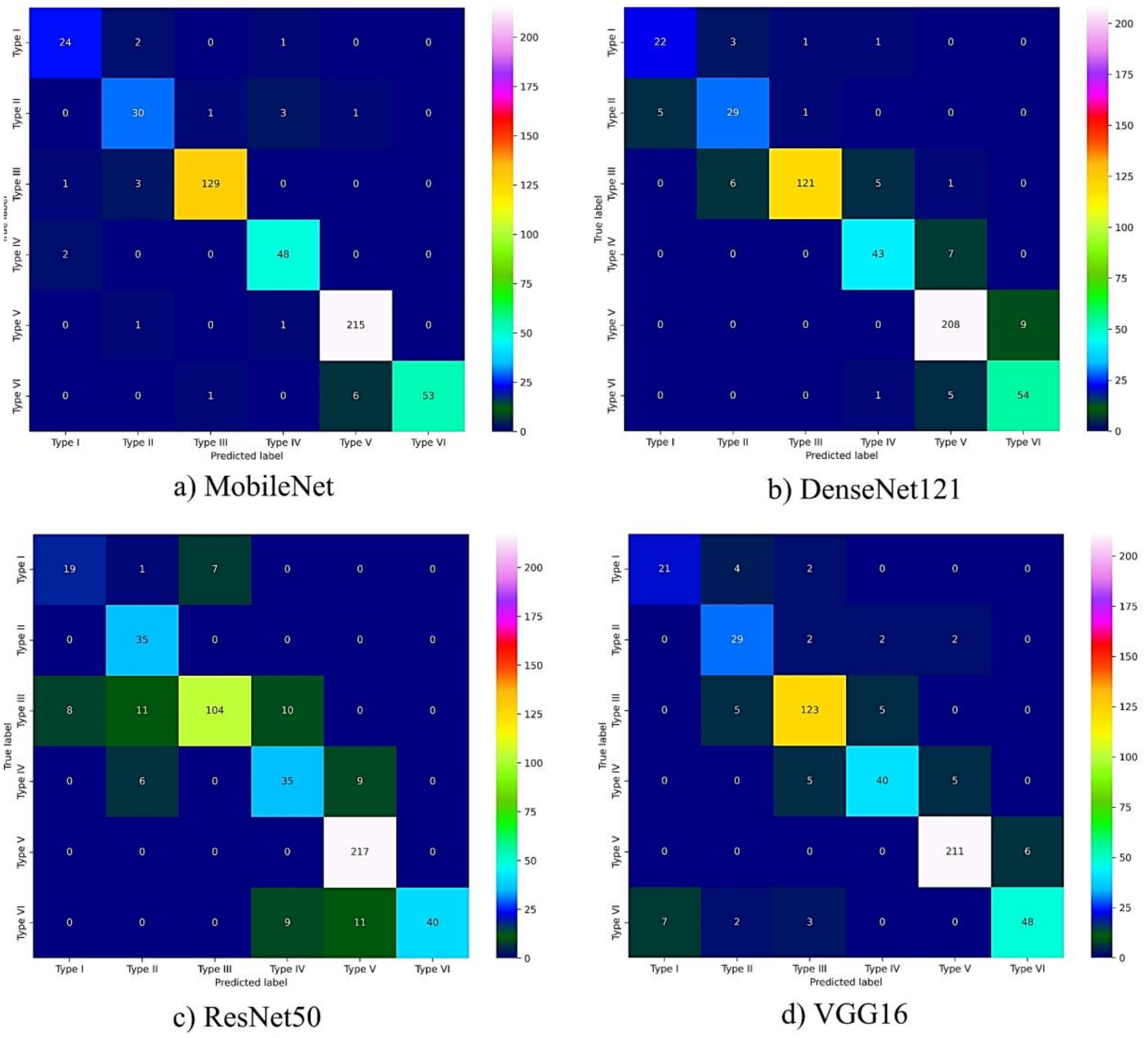
Table 3 shows the statistics of the MobileNet model in predicting the types of concrete failures (types I to VI), expressed in terms of precision (P), recall (R), and F1 score. Type I achieves an accuracy, recall, and F1 score of 89%, maintaining a balance in its classifications. Type II has an accuracy of 83%, a recall of 86%, and an F1-score of 85%, with a higher true positive rate but lower accuracy. Type III has an accuracy of 98%, a recall of 97%, and an F1 score of 98%, showing high classification performance. Type IV shows an accuracy of 91%, a recall of 96%, and an F1 score of 93%, making it a solid and balanced model. Type V achieves an accuracy of 97%, a recall of 99%, and an F1 score of 98%, excelling in the accurate classification of positive samples. Finally, type VI achieves a perfect accuracy of 100%, but its recall is 88% and its F1 score is 94%, demonstrating high accuracy but with losses of positive samples.

Table 4 shows the statistics of the ResNet50 model evaluated in different types of faults using the metrics of precision (P), recall (R), and F1 measurement. It shows good performance, especially with Type III, with an accuracy of 98% and a recall of 91%, indicating that the model achieves an accurate and efficient detection of this class. Type V also shows performance with an accuracy of 94% and a recall of 96%, resulting in an F1 measurement of 95%. On the other hand, Type II shows a balanced performance with an accuracy of 76% and a recall of 83%, which is an acceptable detection in this class. The results of the other classes, such as Type I, Type IV, and Type VI, show values close to 81, 86, and 88% in the three metrics.

Table 5 shows the statistics of the DenseNet121 model in predicting concrete failure types using the metrics of precision (P), recall (R), and F1-measure. Type I had accuracy, recall, and an F1 measure of 70%, indicating a balanced detection, but with room for improvement in this class. In the case of Type II, it shows a recall of 100%, being able to recover all true instances of this class, but the accuracy was 66%, indicating that not all predictions are correct, resulting in an F1 measure of 80%. Type III shows an accuracy of 94% and a recall of 78%, reflecting accurate detection but with a lower ability to recover true instances, resulting in an F1 measurement of 85%. Type IV shows an accuracy of 65% and a recall of 70%, which is acceptable in this class, with an F1 measurement of 67%. On the other hand, Type V has a recall of 92% and 100%, respectively, resulting in an F1 measurement of 96%, highlighting a great performance in this class. Finally, type VI shows an accuracy of 100%, with a recall of 67% due to undetected instances, resulting in an F1 measurement of 80%.

Table 6 shows the statistics of the VGG16 model in terms of accuracy (P), recall (R), and F1 measurement, evaluated for different types of defects. In Type I, the model shows an accuracy of 75%, a recall of 78%, and an F1 measurement of 76%, indicating a balanced detection with room for improvement in this class. Type II shows 70% accuracy, 83% recall, and 77% F1 measurement, demonstrating robust recoverability but with relatively lower accuracy. For Type III, the model shows 91% accuracy, 92% recall, and 92% F1 measurement, reflecting accurate and complete detection in this class. Type IV shows 85% accuracy, 80% recall, and 82% F1 measurement. In Type V, the model achieves 97% accuracy, recall, and F1 measurement, indicating accurate and complete detection in this class. Finally, type VI shows an accuracy of 89%, a recall of 80%, and an F1 measurement of 84%.

The confusion matrix compares the actual vs. predicted labels with the correctly predicted classes plotted on the diagonal and calculates parameters such as precision, recall, and overall classification accuracy. Figure 11 shows the summary of the validation of the models with the actual vs. predicted class of the concrete core failure types: Type I, II, III, IV, V, and VI, validated with a test set labeled with their respective failure types. By analyzing the confusion matrices of the four models, we observed different levels of identification accuracy, with the MobileNet model demonstrating exceptional performance, reaching 96% accuracy in classifying the failure types.



a) MobileNet

b) DenseNet121

c) ResNet50

d) VGG16

Figure 11. Confusion matrix of the different models

The models with the highest accuracy in estimating the failure type in concrete cores were MobileNet and RestNet50 with 96% and 91%, respectively. Table 7 shows the accuracy of the models used to classify the failure type. The "Accuracy" column shows the overall accuracy of each model, which is the proportion of correct predictions in the test data set, noting that the MobileNet model achieved the highest accuracy with a value of 96%. The "Macro Avg" and "Weighted Ave" columns represent the macro and weighted average accuracy metrics, respectively, of all classes on a uniform basis, taking into account the class imbalance and the assignment of weights according to frequency. The MobileNet model leads both metrics with values of 0.93 and 0.96, but the other models show significant results, with RestNet50, DenseNet121, and VGG16 achieving accuracies of 91, 86, and 90%, respectively. The results provide information for the scientific community to evaluate and select the appropriate model for future concrete failure classification applications.

Table 7. Model accuracy

Model	Accuracy (%)	Macro Avg.	Weighted Ave.
MobileNet	96	0.93	0.96
RestNet50	91	0.87	0.91
DenseNet121	86	0.80	0.86
VGG16	90	0.85	0.90

5. Discussion

The research was carried out in the province of Jaen, Peru, because the buildings have cracks and fissures in the structural elements, so the type of concrete failure must be identified to determine the quality of the materials. Table 8 shows the size of the databases, the dimensions of the images, the distribution of the data set, and the classes present in each study. First, it should be noted that most of the previous studies focus on crack detection in images using a variety of data sizes and splits. Li et al. [23], Falaschetti et al. [28], Nguyen et al. [8], Wang et al. [24], Song et al. [4], Wu et al. [41], and Islam et al. [13] all have "crack" and "non-crack" as the detection target. On the other hand, some studies focus on a more diverse dataset, such as Chen et al. [26], which includes the classes of "concrete cracking", "concrete spalling", "rebar exposure", and "background", allowing for more detailed identification of different types of concrete damage. Compared to these references, the study presents a proprietary database with a total of 2650 images, each with a dimension of 224×224 pixels. The database contains six different classes of concrete cylinder failures identified as "Type I", "Type II", "Type III", "Type IV", "Type V", and "Type VI". This is a significant contribution to the field of concrete failure detection, as it provides a diverse and specific database for identifying different types of concrete damage. Furthermore, it should be noted that some studies, such as Wang et al. [24] and Li et al. [23], use a 7:2:1 data split for training, validation, and testing, respectively, while others, such as Falaschetti et al. [28], use a 7:1.5:1.5 split. The study follows a representative dataset split of 60% for training, 20% for validation, and 20% for testing, which ensures a robust evaluation of model performance. In terms of dimensions, the database uses 224×224 pixels images, while other studies use sizes such as 227×227, 256×256, 480×480, and 608×608 pixels. Although there are variations in the dimensions, it is important to note that all the studies present an adequate resolution for the detection of flaws and cracks in concrete. As the database was composed of 2650 images, it is comparable to or even larger than other references, such as Nguyen et al. [8] with 1097 images and Chen [26] with 600 images.

Table 8. Comparison of database and labels used for crack-type classification

Reference	Database	Dimension	Division of the data set	Classes
Li et al. [23]	50,476	227 × 227	7:2:1	Crack and Background
Falaschetti et al. [28]	40,000	227 × 227	7:1.5:1.5	Crack and non-crack
Nguyen et al. [8]	1,097	256 × 256	-	Crack and non-crack
Wang et al. [24]	2,735	608 × 608	7:2:1	Crack and non-crack
Song et al. [4]	962	512 × 512	-	Crack and non-crack
Wu et al. [41]	500	-	6:1.5:2.5	Crack and non-crack
Islam et al. [13]	40,000	227 × 227	-	Crack and non-crack
Qiao et al. [14]	1800	480 × 480	8:2	Crack and non-crack
Chen et al. [26]	600	227 × 227	-	Concrete cracking, concrete spalling, exposure of reinforcing bars, and Background
Present Study	2650	224 × 224	-	Type I, Type II, Type III, Type VI, Type V, and Type VI

Table 9 shows the different hardware and software used by the authors for crack classification. Falaschetti et al. [28] used the CNN V2 model without specifying the hardware, using TensorFlow v.2.5.0 and Keras v.2.5.0 as software to train and evaluate the model. Wang et al. [24] used the SegCrack model on a GeForce GTX 1080Ti GPU with 11 GB of memory. However, they do not mention the software used to train the model. Song et al. [4] used the ResNet50 model on hardware consisting of an Intel(R) Core(TM) i5-8500CPU processor with a GeForce RTX 2060 GPU and 16 GB of RAM and a Windows 10 64-bit operating system with Python 3.7 and TensorFlow as the deep learning library. Chen et al. [26] used an NVIDIA GeForce GTX 960M GPU and MATLAB 2019 as software for their study. For the study, the MobileNet model was used with NVIDIA Tesla V100 hardware and TensorFlow v.2.13.0, and Keras v.2.2.4 software for model training and evaluation. By using the Tesla V100 GPU, significantly reduced training and prediction times were achieved compared to other models and hardware configurations, which accelerated the detection process and allowed for greater efficiency in generating early warnings of defects. It is important to note that each study used different models and hardware and software configurations according to the needs and resources available. The selection of the MobileNet model and the Tesla V100 GPU was based on the efficiency and performance they provided for the detection task.

Table 9. Comparison of hardware and software used for crack-type classification

Reference	Database	Hardware	Software
Falaschetti et al. [28]	CNN V2	-	- TensorFlow v. 2.5.0 - Keras v. 2.5.0
Wang et al. [24]	SegCrack	GeForce GTX 1080Ti 11 GB GPU.	-
Song et al. [4]	ResNet50	Intel(R) Core(TM) i5-8500CPU@3.00 GHz, GeForce RTX 2060, 16 GB RAM.	Windows 10-64-bit, Python 3.7, TensorFlow.
Chen et al. [26]	-	NVIDIA GeForce GTX 960M.	MATLAB 2019.
Present Study	MobileNet	NVIDIA Tesla V100.	-TensorFlow v. 2.13.0 -Keras v. 2.2.4.

Table 10 shows the accuracy of other models concerning the one obtained in this research. Falaschetti et al. [28] used a CNN V2 model and obtained an accuracy of 99.33%. Although the accuracy was 96%, it is within the range of reliability, considering that the approach was based on the MobileNet model, which was designed to provide a balance between accuracy and computational efficiency. Nevertheless, the model showed competitive performance and was suitable for various practical applications. The InceptionV3 model used by Nguyen et al. [8] achieved an accuracy of 90.90% and an F1 value of 0.90. Wang et al. [24] used the SegCrack model and achieved an accuracy of 96.66% and an F1 value of 0.96. When compared to the proposed approach, the results obtained by the aforementioned authors are similar, as they achieved an accuracy of 96% and an F1 value of 97%. These results indicate that the model proposed as SegCrack has similar performance in correctly classifying the type of failure. As for the YOLOv4, AlexNet, and EMA-DenseNet models used by Wu et al. [41], Islam et al. [13], and Qiao et al. [14], respectively, MobileNet presents an accuracy between 81.97% and 97.34%, which is higher and similar to the results obtained in the research compared to these.

Table 10. Comparison of crack-type classification accuracy

Reference	Model	Accuracy (%)	F1
Falaschetti et al. [28]	CNN V2	99.33	-
Nguyen et al. [8]	InceptionV3	90.90	0.90
Wang et al. [24]	SegCrack	96,66	0.96
Wu et al. [41]	YOLOv4	97.34	0.97
Islam et al. [13]	AlexNet	99.92	99.86
Qiao et al. [14]	EMA-DenseNet	81.97	-
Chen et al. [26]	AlexNet	86.00	-
Present Study	MobileNet	96.00	97.00

6. Conclusion

Given the lack of open access data, this research has created a database with 2650 images of six types of failures: Type I, II, III, IV, V, and VI in concrete cylinders obtained from ASTM C39 tests during the years 2019–2023. Four convolutional neural network models with learning transfer were proposed: MobileNet, RestNet50, DenseNet121, and VGG16, to be classifiers and pattern recognizers in crack classification, using the Python Keras library. For this purpose, the dataset was divided into training (60%), validation (20%), and testing (20%), as recommended in the literature. Each model was trained for 50 epochs. After validation, the predicted classes were evaluated against the real ones (concrete failure types) using the confusion matrix. The MobileNet model proved to be the best estimator of concrete failure type, reaching an accuracy of 96% and an error of 4%. This allows automation in the identification of specific core failure types. Comparing MobileNet with other models in crack classification, it is more accurate, with a difference of more than 5% compared to InceptionV3 and YOLOv4. The results validated in the research with accuracies of 96, 91, and 90% have great potential for engineers and researchers to make informed decisions and solve problems related to the classification of failure types in concrete cylinders. This will enable monitoring and automation in the generation of early warning signals in concrete structures. The research provides a solid foundation for future advances in concrete failure detection and classification and opens the possibility of applying these techniques to other areas of civil and structural engineering.

7. Declarations

7.1. Author Contributions

Conceptualization, J.M.P.O.; methodology, L.Q.H., and J.M.P.O.; software, J.M.P.O.; investigation, J.M.P.O.; writing—original draft preparation, J.M.P.O., and L.Q.H.; writing—review and editing, M.E.M.P., and B.A.C.C.; supervision, J.L.P.T., and W.R.P. All authors have read and agreed to the published version of the manuscript

7.2. Data Availability Statement

The data presented in this study are available in the article.

7.3. Funding

The authors received no financial support for the research, authorship, and/or publication of this article.

7.4. Conflicts of Interest

The authors declare no conflict of interest.

8. References

[1] Anjali, R., & Venkatesan, G. (2022). Optimization of mechanical properties and composition of M-sand and pet particle added concrete using hybrid deep neural network-horse herd optimization algorithm. Construction and Building Materials, 347, 128334. doi:10.1016/j.conbuildmat.2022.128334.

[2] Ewida, E. S. S., Mabrouk, R. T., El-Shafey, N., & Torkey, A. M. (2022). Flexural behavior of one-way slabs reinforced with welded wire mesh under vertical loads. Civil Engineering Journal, 8(4), 654-671. doi:10.28991/CEJ-2022-08-04-03.

[3] Laxman, K. C., Tabassum, N., Ai, L., Cole, C., & Ziehl, P. (2023). Automated crack detection and crack depth prediction for reinforced concrete structures using deep learning. Construction and Building Materials, 370, 130709. doi:10.1016/j.conbuildmat.2023.130709.

[4] Song, L., Sun, H., Liu, J., Yu, Z., & Cui, C. (2022). Automatic segmentation and quantification of global cracks in concrete structures based on deep learning. Measurement, 199, 111550. doi:10.1016/j.measurement.2022.111550.

[5] Ren, J., Tian, Z., & Bu, J. (2018). Simulating Tensile and Compressive Failure Process of Concrete with a User-defined Bonded-Particle Model. International Journal of Concrete Structures and Materials, 12(1), 56. doi:10.1186/s40069-018-0292-1.

[6] Li, Y., Li, H., & Wang, H. (2018). Pixel-wise crack detection using deep local pattern predictor for robot application. Sensors (Switzerland), 18(9), 3042. doi:10.3390/s18093042.

[7] Hadinata, P. N., Simanta, D., Eddy, L., & Nagai, K. (2021). Crack Detection on Concrete Surfaces Using Deep Encoder-Decoder Convolutional Neural Network: A Comparison Study Between U-Net and DeepLabV3+. Journal of the Civil Engineering Forum, 7(3), 323. doi:10.22146/jcef.65288.

[8] Nguyen, A., Gharehbaghi, V., Le, N. T., Sterling, L., Chaudhry, U. I., & Crawford, S. (2023). ASR crack identification in bridges using deep learning and texture analysis. Structures, 50, 494–507. doi:10.1016/j.istruc.2023.02.042.

[9] Golding, V. P., Gharineiat, Z., Munawar, H. S., & Ullah, F. (2022). Crack Detection in Concrete Structures Using Deep Learning. Sustainability (Switzerland), 14(13), 8117. doi:10.3390/su14138117.

[10] Mendonça, Y. V., Naranjo, P. G. V., & Pinto, D. C. (2022). The Role of Technology in the Learning Process. Emerging Science Journal, 6(Special Issue), 280-295. doi:10.28991/ESJ-2022-SIED-020.

[11] Jayaraju, P., Somasundaram, K., Suprakash, A. S., & Muthusamy, S. (2022). A Deep Learning- Image Based Approach for Detecting Cracks in Buildings. Traitement Du Signal, 39(4), 1429–1434. doi:10.18280/ts.390437.

[12] Kolappan Geetha, G., Yang, H.-J., & Sim, S.-H. (2023). Fast Detection of Missing Thin Propagating Cracks during Deep-Learning-Based Concrete Crack/Non-Crack Classification. Sensors, 23(3), 1419. doi:10.3390/s23031419.

[13] Islam, M. M., Hossain, M. B., Akhtar, M. N., Moni, M. A., & Hasan, K. F. (2022). CNN Based on Transfer Learning Models Using Data Augmentation and Transformation for Detection of Concrete Crack. Algorithms, 15(8). doi:10.3390/a15080287.

[14] Qiao, W., Ma, B., Liu, Q., Wu, X., & Li, G. (2021). Computer Vision-Based Bridge Damage Detection Using Deep Convolutional Networks with Expectation Maximum Attention Module. Sensors, 21(3), 824. doi:10.3390/s21030824.

[15] Koch, C., Georgieva, K., Kasireddy, V., Akinci, B., & Fieguth, P. (2015). A review on computer vision based defect detection and condition assessment of concrete and asphalt civil infrastructure. Advanced Engineering Informatics, 29(2), 196–210. doi:10.1016/j.aei.2015.01.008.

[16] Abdel-Qader, I., Abudayyeh, O., & Kelly, M. E. (2003). Analysis of Edge-Detection Techniques for Crack Identification in Bridges. Journal of Computing in Civil Engineering, 17(4), 255–263. doi:10.1061/(asce)0887-3801(2003)17:4(255).

[17] Haque, I., Alim, M., Alam, M., Nawshin, S., Noori, S. R. H., & Habib, M. T. (2022). Analysis of recognition performance of plant leaf diseases based on machine vision techniques. Journal of Human, Earth, and Future, 3(1), 129-137. doi:10.28991/HEF-2022-03-01-09.

[18] Adhikari, R. S., Moselhi, O., & Bagchi, A. (2014). Image-based retrieval of concrete crack properties for bridge inspection. Automation in Construction, 39, 180–194. doi:10.1016/j.autcon.2013.06.011.

[19] Dinh, T. H., Ha, Q. P., & La, H. M. (2016). Computer vision-based method for concrete crack detection. 2016 14th International Conference on Control, Automation, Robotics and Vision (ICARCV). doi:10.1109/icarcv.2016.7838682.

[20] Lee, J., Kim, H.-S., Kim, N., Ryu, E.-M., & Kang, J.-W. (2019). Learning to Detect Cracks on Damaged Concrete Surfaces Using Two-Branched Convolutional Neural Network. Sensors, 19(21), 4796. doi:10.3390/s19214796.

[21] Ali, L., Alnajjar, F., Jassmi, H. A., Gocho, M., Khan, W., & Serhani, M. A. (2021). Performance Evaluation of Deep CNN-Based Crack Detection and Localization Techniques for Concrete Structures. Sensors, 21(5), 1688. doi:10.3390/s21051688.

[22] Ye, W., Deng, S., Ren, J., Xu, X., Zhang, K., & Du, W. (2022). Deep learning-based fast detection of apparent concrete crack in slab tracks with dilated convolution. Construction and Building Materials, 329, 127157. doi:10.1016/j.conbuildmat.2022.127157.

[23] Li, Y., Bao, T., Xu, B., Shu, X., Zhou, Y., Du, Y., Wang, R., & Zhang, K. (2022). A deep residual neural network framework with transfer learning for concrete dams patch-level crack classification and weakly-supervised localization. Measurement, 188, 110641. doi:10.1016/j.measurement.2021.110641.

[24] Wang, W., & Su, C. (2022). Automatic concrete crack segmentation model based on transformer. Automation in Construction, 139, 104275. doi:10.1016/j.autcon.2022.104275.

[25] Yu, G., Dong, J., Wang, Y., & Zhou, X. (2023). RUC-Net: A Residual-Unet-Based Convolutional Neural Network for Pixel-Level Pavement Crack Segmentation. Sensors, 23(1), 53. doi:10.3390/s23010053.

[26] Chen, L., Chen, W., Wang, L., Zhai, C., Hu, X., Sun, L., Tian, Y., Huang, X., & Jiang, L. (2023). Convolutional neural networks (CNNs)-based multi-category damage detection and recognition of high-speed rail (HSR) reinforced concrete (RC) bridges using test images. Engineering Structures, 276, 115306. doi:10.1016/j.engstruct.2022.115306.

[27] Jiang, Y., Pang, D., & Li, C. (2021). A deep learning approach for fast detection and classification of concrete damage. Automation in Construction, 128, 103785. doi:10.1016/j.autcon.2021.103785.

[28] Falaschetti, L., Beccerica, M., Biagetti, G., Crippa, P., Alessandrini, M., & Turchetti, C. (2022). A Lightweight CNN-Based Vision System for Concrete Crack Detection on a Low-Power Embedded Microcontroller Platform. Procedia Computer Science, 207, 3948–3956. doi:10.1016/j.procs.2022.09.457.

[29] Yeşilmen, S., & Tatar, B. (2022). Efficiency of convolutional neural networks (CNN) based image classification for monitoring construction related activities: A case study on aggregate mining for concrete production. Case Studies in Construction Materials, 17, 1372. doi:10.1016/j.cscm.2022.e01372.

[30] Kulkarni, U., Meena, S. M., Gurlahosur, S. V., & Bhogar, G. (2021). Quantization Friendly MobileNet (QF-MobileNet) Architecture for Vision Based Applications on Embedded Platforms. Neural Networks, 136, 28–39. doi:10.1016/j.neunet.2020.12.022.

[31] Mao, W.-L., Chen, W.-C., Wang, C.-T., & Lin, Y.-H. (2021). Recycling waste classification using optimized convolutional neural network. Resources, Conservation and Recycling, 164, 105132. doi:10.1016/j.resconrec.2020.105132.

[32] Shabbir, A., Ali, N., Ahmed, J., Zafar, B., Rasheed, A., Sajid, M., Ahmed, A., & Dar, S. H. (2021). Satellite and Scene Image Classification Based on Transfer Learning and Fine Tuning of ResNet50. Mathematical Problems in Engineering, 2021, 5843816. doi:10.1155/2021/5843816.

[33] Qu, Z., Mei, J., Liu, L., & Zhou, D. Y. (2020). Crack detection of concrete pavement with cross-entropy loss function and improved VGG16 network model. IEEE Access, 8, 54564–54573. doi:10.1109/ACCESS.2020.2981561.

[34] Yu, Y., Samali, B., Rashidi, M., Mohammadi, M., Nguyen, T. N., & Zhang, G. (2022). Vision-based concrete crack detection using a hybrid framework considering noise effect. Journal of Building Engineering, 61, 105246. doi:10.1016/j.jobe.2022.105246.

[35] Wijaya, I. G. P. S., Dwitama, A. P. J., Widiartha, I. B. K., & Putra, S. A. (2020). Classification of Building Cracks Image Using the Convolutional Neural Network Method. 2020 International Conference on Advancement in Data Science, E-Learning and Information Systems (ICADEIS). doi:10.1109/icadeis49811.2020.9276962.

[36] Shorten, C., & Khoshgoftaar, T. M. (2019). A survey on Image Data Augmentation for Deep Learning. Journal of Big Data, 6(1), 60. doi:10.1186/s40537-019-0197-0.

[37] Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., ... & Adam, H. (2017). Mobilenets: Efficient convolutional neural networks for mobile vision applications. arXiv preprint. doi:10.485550/arXiv.1704.04861.

[38] Cao, Q. H., Nguyen, T. T. H., Nguyen, V. T. K., & Nguyen, X. P. (2023). A Novel Explainable Artificial Intelligence Model in Image Classification problem. arXiv preprint. doi:10.48550/arXiv.2307.04137.

[39] Huang, G., Liu, Z., Van Der Maaten, L., & Weinberger, K. Q. (2017). Densely Connected Convolutional Networks. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). doi:10.1109/cvpr.2017.243.

[40] Qassim, H., Verma, A., & Feinzimer, D. (2018). Compressed residual-VGG16 CNN model for big data places image recognition. 2018 IEEE 8th Annual Computing and Communication Workshop and Conference (CCWC). doi:10.1109/ccwc.2018.8301729.

[41] Wu, P., Liu, A., Fu, J., Ye, X., & Zhao, Y. (2022). Autonomous surface crack identification of concrete structures based on an improved one-stage object detection algorithm. Engineering Structures, 272, 114962. doi:10.1016/j.engstruct.2022.114962.