

Article

Building Surface Defect Detection Using Machine Learning and 3D Scanning Techniques in the Construction Domain

Alexandru Marin Mariniuc, Dorian Cojocaru *  and Marian Marcel Abagiu

Mechatronics and Robotics Department, University of Craiova, 200585 Craiova, Romania; alexandru.mariniuc@edu.ucv.ro (A.M.M.); abagiu.marian.n5u@student.ucv.ro (M.M.A.)

* Correspondence: dorian.cojocaru@edu.ucv.ro

Abstract: The rapid growth of the real estate market has led to the appearance of more and more residential areas and large apartment buildings that need to be managed and maintained by a single real estate developer or company. This scientific article details the development of a novel method for inspecting buildings in a semi-automated manner, thereby reducing the time needed to assess the requirements for the maintenance of a building. This paper focuses on the development of an application which has the purpose of detecting imperfections in a range of building sections using a combination of machine learning techniques and 3D scanning methodologies. This research focuses on the design and development of a machine learning-based application that utilizes the Python programming language and the PyTorch library; it builds on the team's previous study, in which they investigated the possibility of applying their expertise in creating construction-related applications for real-life situations. Using the Zed camera system, real-life pictures of various building components were used, along with stock images when needed, to train an artificial intelligence model that could identify surface damage or defects such as cracks and differentiate between naturally occurring elements such as shadows or stains. One of the goals is to develop an application that can identify defects in real time while using readily available tools in order to ensure a practical and affordable solution. The findings of this study have the potential to greatly enhance the availability of defect detection procedures in the construction sector, which will result in better building maintenance and structural integrity.



Citation: Mariniuc, A.M.; Cojocaru, D.; Abagiu, M.M. Building Surface Defect Detection Using Machine Learning and 3D Scanning Techniques in the Construction Domain. *Buildings* **2024**, *14*, 669. <https://doi.org/10.3390/buildings14030669>

Academic Editors: Yantao Yu and Shuai Han

Received: 4 January 2024

Revised: 10 February 2024

Accepted: 29 February 2024

Published: 2 March 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

Keywords: 3D scanning; machine learning; artificial intelligence

1. Introduction

The paper presents the progress of the research team regarding the use of machine learning in combination with 3D scanning techniques for the detection of defects on the surface of various elements of a building, such as walls, pillars, stairs, ceilings, foundations, and others.

In a previous paper, the research team analyzed the possibility of using some of the experience that it gained from the development and practical implementation of certain applications as a starting point in the development of a machine learning-based 3D scanning application in the construction domain. It is worth mentioning that LiDAR and ZED camera systems were used in acquiring real images of both interior and exterior parts of certain buildings [1].

In the current paper, the team presents the development of a machine learning-based application that uses the Python programming language and the PyTorch library v2.2.0. The team used the Zed camera system for the acquisition of real images of different building elements; the images were then used in combination with standard building images to train the artificial intelligence model to recognize different types of damage or defects on the surface of the aforementioned building elements.

The aim of the research team is for the proposed application to scan and classify the defects in real time whilst using components available on the market, such as the Zed camera and a standard laptop or computer, thus keeping the cost of the solution low and affordable. The cost of a Zed camera is around USD 500, compared to the scanners usually used in the construction domain that range anywhere from tens of thousands to hundreds of thousands of dollars. Due to its cost-effective nature and its small size factor, the solution provides a small initial investment and lower lifecycle costs, which makes it practical not only for big construction companies but also for smaller contractors.

2. Related Work

Machine learning techniques have been adopted by some researchers in the construction domain, and it has been proven that they perform well in tasks such as design automation, automatic control, and optimization in intelligent buildings [2]. This success led to computer vision techniques also being studied by researchers; thus, they have become more capable of automatically detecting surface defects and eliminating the drawbacks of the classic defect detection models [3].

In their paper, Konstantinos Bacharidis et al. presented a methodology for the accurate 3D realistic façade reconstruction of cultural heritage buildings. Their goal was to use deep neural network architectures used for image segmentation and depth prediction to detect structural elements and generate simulations of the surfaces of the buildings scanned. One of their challenges involved differentiating between certain elements of the design that were the key to differentiating modern buildings from historical buildings. Several technologies were used for scanning, such as terrestrial laser scanning and close-range photogrammetry, to produce point cloud maps. To compensate for the weaknesses of laser scanning, such as glass windows, walls, and doors, additional sensors were used when needed, such as geodetic stations and optical camera sensors; thus, the laser distortion effects were reduced. The point clouds are combined with laser scanning data and are used to simulate photo-realistic reconstructions of the surfaces. The proposed framework enhances the automation and the applicability of building surface scanning by utilizing deep learning techniques [4].

Compared to the previously mentioned work, the solution presented in this paper uses an optical camera (i.e., a Zed camera system) instead of a LiDAR system, which is more expensive and, as discovered in the previous works of the research team, more sensitive to infrared light when working outside. Their paper represents a basis for the differentiation between cracks and shadows or stains in our solution, as presented below.

Remote sensing technologies play a crucial role in surveying and scanning the complex geometries and architectural features of historical buildings. In their work, Gustavo Rocha et al. explain the use of 3D laser scanning and photogrammetry in the development of a methodology to obtain a consistent model that can help with the restoration or conservation of historical buildings. The 3D scanning makes it possible to integrate the real-life building in a building information modeling system that allows the restorers to take advantage of certain benefits, such as design alternatives, cost estimates, material quantifications, data management, as-built documentation, constructive state analysis, execution plans, and many others. It is extremely important to ensure that the measurements and scans are as precise as possible since heritage buildings need to be preserved with as much detail as possible. Therefore, BIM (building information modelling) is not just software, but an integrated collaborative methodology centered on a building model that represents the real-life status of the building and contains accurate information about it. The integration of BIM into the 3D scanned point cloud data has proven to be a crucial but difficult step in the development of an integrated system. If used correctly from the first step, a BIM model becomes essential to the success of such an application. In their work, the authors also mention that the operator's experience is crucial at all stages because the use of laser scanning and photogrammetry equipment requires knowledge of architecture and

building techniques. The method presented is mostly conducted manually, with only a few automatic processes used for the 3D reconstruction of the topographic surface [5].

Gustavo Rocha's work serves not only as a basis for the future development of the solution presented in our paper into an autonomous system for the non-invasive scanning of building surfaces; it also helps the research team to have a better understanding of the capturing of façade details, which has proven to be critical for the correct prediction of defects during the training phase of the artificial intelligence model.

Analyzing the potential of 3D scanning for construction management purposes was covered by Matej Mihić et al. in their work. The goal of their research was to find out whether the construction industry was open to change and whether they could justify the higher cost compared to that of traditional monitoring techniques. One of the technologies studied is LiDAR, which is the most accurate vision-based sensing technology for producing high-resolution point clouds. Photogrammetry is another popular vision-based sensing technology that can generate 3D point cloud models from 2D images. This technique involves image processing and makes the classification of objects possible. The possibility of collecting data with the help of unmanned aerial vehicles or terrestrial automated vehicles is also presented in this paper. The UAV, for example, can provide more comprehensive 3D clouds that also include data from the roofs of the buildings, rather than only from the walls. The most common UAVs used are multirotor drones, due to their robustness, maneuverability, low purchase and maintenance costs, hovering ability, vertical take-off, and landing. In combination with depth cameras, they can create 3D models of buildings in innovative ways [6]. This paper represents a basis for the planning of the future work on the application that is presented in our research paper.

The research team involved in the present article worked with LiDAR sensors in a previous research project, using the sensory system to scan various hallways, pathways, access doors, and obstacle-filled spaces in a building. They used the depth data from the LiDAR point cloud, which were associated with the RGB data for each point to simulate certain objects from the data captured [7].

It is worth mentioning that on the construction sites, the conditions are not always ideal. The experience gained during this previous work offered the research team good knowledge about and insight into the best practices for scanning in various environments. It allowed the team to capture the best possible training images, which led to better results in the training loss and the training accuracy. This was later observed in the predictions made by the model, which were more correct.

Convolutional neural networks and transfer learning methods can be used together to help identify the condition of historic building facades. Transfer learning accumulates the knowledge gained by resolving problems in the past and uses it to solve new similar problems [8]. This possibility is studied by Sumaiyah Fitriani Dini et al.; they present the application of deep learning in the field of architecture for building feature classification and recognition. In their paper, the authors focus on identifying the styles and functions of buildings in urban areas; they also focus on the construction era and period and the automatic detection of defects and damage to buildings. The neural networks have also been applied for the automatic detection of concrete cracks, mold, damage, and stains in images and for the detection of real-time building damage based on terrestrial images. The authors acknowledge that the limited amount of datasets significantly influenced the training and testing processes; therefore, it is advisable to use a correct and large enough dataset when training the artificial intelligence model used in the application presented in the following chapters [9].

Their work helped the research team to assess the number of images needed to build a proper dataset for our application. As shown in their paper, a small number of images can lead to low prediction accuracy.

A method for crack detection is detailed by Stamos Katsigiannis et al. in their paper. The approach they use is based on a pre-trained deep convolutional neural network optimized for feature extraction, particularly crack detection. Non-destructive techniques

and photogrammetry have been used for the inspection of brick buildings, but manual inspection is not time-effective, and it can lead to errors. The presented detection methods reduce the time needed for the work to be completed [10]. Transfer learning was used to address this gap by classifying brickwork images as cracked or normal. They used images from historical buildings in combination with images acquired from online sources. The main contribution of this research includes the public release of an image dataset for crack detection on brick surfaces and a comparative study of widely used pre-trained convolutional neural network models adapted for crack detection using transfer learning. The results showed that the method used achieved 100% accuracy using libraries such as the MobileNetV2, InceptionNetV2, and Xception-based models. The MobileNetV2 model was the most efficient one due to its small size, which makes it ideal for handheld devices and autonomous vehicles [11]. This paper has two important points of interest for our work, namely the use of both images acquired from the facades of real historical buildings and stock images from online libraries; the second point is the use of an artificial intelligence model that does not use a lot of resources, making the application light enough to be used on drones or other types of automatically guided vehicles.

3. Data Capturing

Based on the hands-on experience gained through previous research projects, the team acquired the practical ability to correctly scan and acquire point cloud datasets with various sensors [1]. In the following section, we present the reasoning behind choosing one type of sensory system over the other available systems.

3.1. Using Various Sensors to Capture Building Façade Data

Whilst LiDAR has proven to be the most accurate sensor tested, with a scan resolution of 0.1° – 0.4° horizontally and 2° vertically and one of the best refresh rates at 20 times/s, it is safe to say that performance has a high cost [12]. The high costs of the LiDAR technology are justified by its high performance across all its capabilities, but such high performance is not needed in applications that are focused on a niche section such as crack detection, which does not use the entire range of features of a LiDAR system [13]. The goal of the present paper is to develop an affordable and cost-efficient application; therefore, the high prices of LiDAR equipment make it not fit for the proposed application.

Another sensory system used was the ZED camera system, which consists of a dual depth-sensing camera with resolutions ranging from 1344×376 pixels at 100 frames per second up to 4416×1242 pixels, reaching a refresh rate of up to 15 frames per second at this high resolution. This range of resolutions and refresh rates makes the system adaptable to various scanning methods [14]. Jose Eleazar Peralta Lopez et al. have also demonstrated that the ZED camera is rugged enough to withstand real-life conditions by being mounted on the front of an SUV while collecting images on the streets of a real-life city [15]. For example, if the research team decides to scan buildings from a fixed point, then it is better to use a high resolution to capture as much detail as possible. If the decision is to use a drone in the future, then the team must sacrifice resolution for more frames, to match the speed of a drone and keep the flow of data constant. Otherwise, in the case of an automated terrestrial vehicle, there can be a balance between quality and speed, making it the right choice if you need to capture fine details at a faster rate than can be achieved with manual labor. The system's small size and portability also make it suitable for use in combination with a drone or an automatically guided vehicle.

Other sensory systems used by the research team in previous work were Microsoft Kinect, which is like the ZED camera system (Figure 1) and the DJI Guidance system, which was made specifically for drone guidance [16]. The most advanced sensor used was the Velodyne VLP-16 Puck LiDAR, with a 360-degree scan range. It was calculated that it needs at least 6 scans to make a full 360-degree simulation of a building interior [17]. With good calibration, a LiDAR sensor is able to scan and produce 3D maps in a very short timeframe, with an accuracy of 1 mm [18].



Figure 1. ZED camera system.

3.2. Using Artificial Intelligence in Detecting Building Features

In previous research projects, the team acquired knowledge about training a convolutional neural network model. In concordance with image-processing techniques, the team was able to use a trained artificial intelligence model to extract features from real-time images to correctly detect and classify various objects such as road signs. The model was trained on pre-existing data like those on which it was used, and the key features of the objects were tagged manually in the training process.

The convolutional neural network contained 30 layers; after capturing, the image was passed through 5 groups of convolutional layers with 5×5 kernels; then, it was passed through 4 neural layers with 3×3 kernels. To classify the defects, we had to identify the key features in each image; for this, the team used labelling during training [19]. The training was made on 100 manually labelled images and had a duration of 4.30 h. The object detection then worked on discovering the specific area in an image that matched a previously labelled defect [20].

In another application, the team developed a feature extraction model for detecting defects in engine blocks. The accuracy obtained in the training process was 99.4% with the help of CUDA processing on an NVIDIA GPU with the use of PyTorch. The dataset used was limited, with a set of 1000 images with defects and a set of 15,000 images with good parts. The lesson learned from this work is based on the set of data used for training a model; there must be enough training data for the model to stabilize; however, if there are too many data items, the border between a good detection and a false detection becomes too thin and this results in false positives. The application was implemented on the production line of a global car manufacturer.

4. Solution Overview

By its nature, the hardware solution composed of readily available equipment ensures a high compatibility between each individual component. The Zed camera is compatible with any platform that recognizes universal video class devices; therefore, it is compatible with any platform that can support the Linux operating system, be it a laptop, desktop, Raspberry Pi, etc. The PyTorch library is supported on Linux versions newer than 15 July 2012; therefore, there are many versions that could be used and that are compatible with an even larger number of devices, thus ensuring the overall compatibility of the solution with a large number of equipment combinations (Laptop, Linux OS, Zed camera).

For the software solution, the research team considered two artificial intelligence training models, namely ResNet50 and ResNet101. The deep residual network architecture (ResNet) is presented in [21] as a powerful model for image recognition. Some defects are similar and are categorized according to the criteria of the researchers; for example, in the detection of surface defects on steel, there are contusions [20], protrusions [22], abrasions [23], wrinkles [24], rubbing [25], and dents [26]. Similarly, in the construction

domain, there are many defects present on the surface, such as cracks, paint chips, stains, and many more; the standardization of these defects is proposed by Macarulla M. et al. in their paper [27]. Each of the models was first trained and tested on an online image library of various pictures of walls, with or without the aforementioned defects.

The next phase of the training involved the capturing of real-life images of walls from different construction sites. As in the online library, the walls were split into two categories: walls with no apparent defects and walls that presented defects. The models were then tested on real-life images, and the research team documented the results and chose the training model that best suited our needs. As Hongyu Xu et al. presented in detail in their paper, it is very difficult to analyze the large volume of captured images without the help of artificial intelligence; thus, the team was inclined to choose the more capable model [28].

In the following chapter, the tests and the results are presented for each of the models in both situations: the online library and the real-life images captured by the team.

4.1. Results Using the ResNet50 Training Model on an Online Image Library

The research team started with the ResNet50 training model and with an online image library that consists of pictures of walls split into two sections. One section is composed of pictures of walls that have no cracks or other kinds of damage, and the other section is composed of pictures of walls that present defects such as cracks.

In the development of the presented solution, the team used a well-established technique based on a training model which is part of the ResNet (residual network) family. The training model was a convolutional neural network (CNN) architecture composed of many layers (i.e., 50 layers for ResNet50 and 101 layers for ResNet101). A neural network is an architecture composed of an input and an output layer, and between them, there is a large number of hidden layers that are connected and work in parallel [29]. The larger the number of hidden layers, the deeper the neural network becomes [30]. The ResNet architecture allows gradients to flow more directly through its layers; instead of learning the mapping directly from the input to the output of the layer, it learns the residual mapping, which is added to the output of the layer. This makes it easier to train deep networks without losing performance. ResNet is trained using supervised learning with large-scale datasets, and the parameters of the network are updated constantly with algorithms like stochastic gradient descent (SGD). The model can be used for tasks such as image classification, object detection, and image segmentation, which makes it optimal for our approach.

Using the ResNet50 training model, the team ran the training for 10 steps of training for the entire set of pictures, which added up to a total of approximately 140 min. During each step of training, the training accuracy can be seen to improve, as shown in Figure 2.

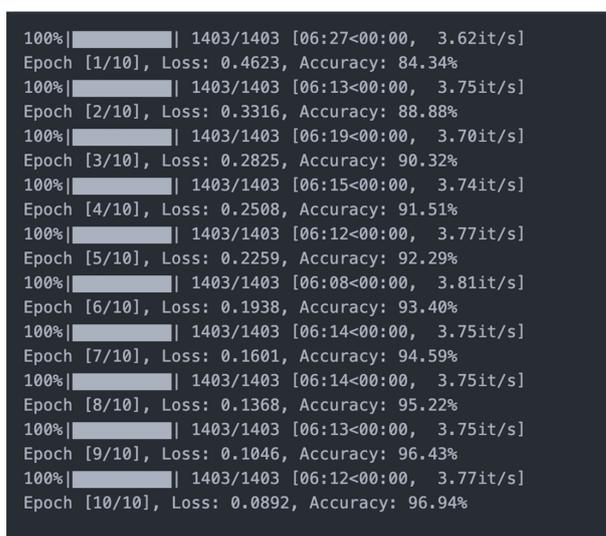


Figure 2. Accuracy of the ResNet50 model improvement at each step of training.

The goal is to train the model until the point where the training loss is minimal and the training accuracy is as high as possible, as shown in Figure 3. The values of the hyperparameter configuration (learning rate, batch size, etc.) are based on the used hardware; in this case, Apple Neural Engine from the M1 Pro was used.

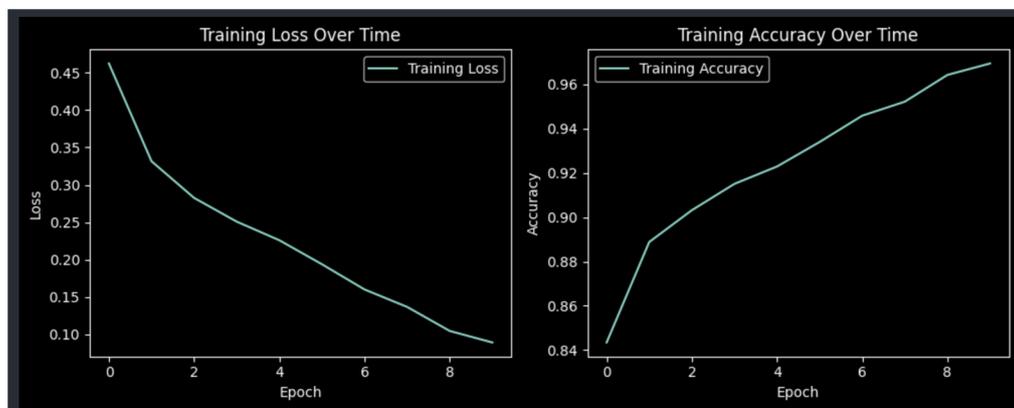


Figure 3. Training loss and training accuracy using ResNet50 on an online image library.

A pre-processing procedure was applied to all images when creating and defining the dataset in order to have a uniform image size and format (resizing of the images, grayscale transformation if needed). Also, the dataset was split, and 25% of the images were used for the testing of the model; the remaining 75% were used in training. The last layer of the ResNet50 model was replaced with the custom classes needed for the wall classification.

The final test accuracy of the automatic detection can be seen in Figure 4. The accuracy is affected by the small dataset and by the low resolution.

Test Accuracy: 92.06%

Figure 4. The final accuracy of the ResNet50 model using an online image library.

4.2. Results Using the ResNet101 Training Model on an Online Image Library

The second step was to combine the available dataset with the new images acquired by the research team from the damaged or undamaged buildings to produce a new set of images. This new dataset was initially used for training a new ResNet50 AI model but with poor results. The training results were promising but when used for inference on real images, the model was not able to classify with an acceptable accuracy.

With this in mind, an upgrade of the training environment was considered, and a ResNet101 neural network was set in place; also, the last layer used for classification was replaced with our custom classes of damaged or undamaged buildings. The model parameters used for both models during the training are presented in Table 1.

Table 1. Model parameters during training.

Model	Learning Rate	Optimizer	Criterion
ResNet50	0.001	Adam	Cross-entropy loss for classification
ResNet101	0.001	Adam	Cross-entropy loss for classification

By comparing the graph of the training loss and training accuracy shown in Figure 3 with the graph shown in Figure 5, it can be observed that the graphs follow the same general behavior, but in the case of the ResNet101 model, the training loss decreases faster

(0.3316 loss for ResNet50 vs. 0.3182 loss for ResNet101 at step 2), and the training accuracy improves in a shorter time right from the beginning (88.88% training accuracy for Resnet50 vs. 89.24% training accuracy for ResNet101 at step 2).

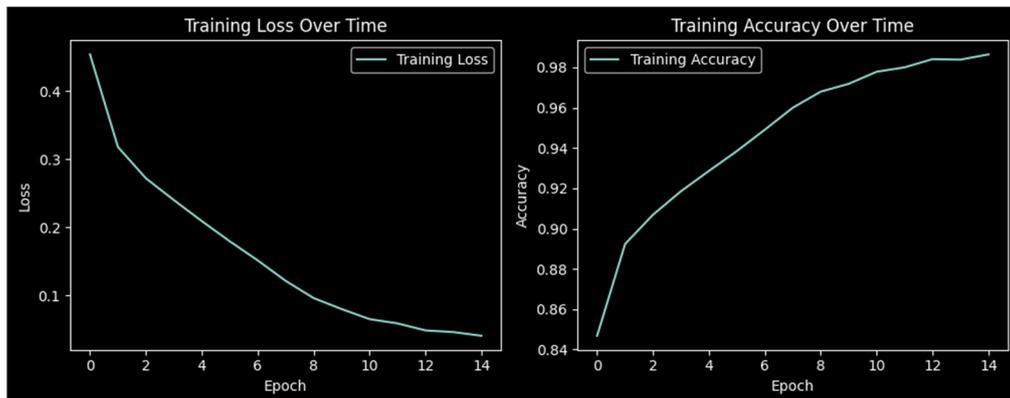


Figure 5. Training loss and training accuracy using ResNet101 on an online image library.

Also, as can be observed in Figure 6, running the training for more steps (15 steps for the ResNet101 model vs. 10 steps for the ResNet50 model), allows the training accuracy to increase up to 98.65%, compared to 96.94%.

```

100%|██████████| 1403/1403 [06:10<00:00, 3.79it/s]
Epoch [1/15], Loss: 0.4541, Accuracy: 84.69%
100%|██████████| 1403/1403 [08:21<00:00, 2.80it/s]
Epoch [2/15], Loss: 0.3182, Accuracy: 89.24%
100%|██████████| 1403/1403 [06:05<00:00, 3.83it/s]
Epoch [3/15], Loss: 0.2723, Accuracy: 90.69%
100%|██████████| 1403/1403 [17:05<00:00, 1.37it/s]
Epoch [4/15], Loss: 0.2403, Accuracy: 91.86%
100%|██████████| 1403/1403 [06:09<00:00, 3.80it/s]
Epoch [5/15], Loss: 0.2094, Accuracy: 92.86%
100%|██████████| 1403/1403 [06:12<00:00, 3.77it/s]
Epoch [6/15], Loss: 0.1798, Accuracy: 93.85%
100%|██████████| 1403/1403 [06:11<00:00, 3.77it/s]
Epoch [7/15], Loss: 0.1517, Accuracy: 94.91%
100%|██████████| 1403/1403 [06:12<00:00, 3.77it/s]
Epoch [8/15], Loss: 0.1217, Accuracy: 96.00%
100%|██████████| 1403/1403 [06:11<00:00, 3.77it/s]
Epoch [9/15], Loss: 0.0964, Accuracy: 96.80%
100%|██████████| 1403/1403 [06:10<00:00, 3.78it/s]
Epoch [10/15], Loss: 0.0803, Accuracy: 97.19%
100%|██████████| 1403/1403 [06:10<00:00, 3.79it/s]
Epoch [11/15], Loss: 0.0655, Accuracy: 97.78%
100%|██████████| 1403/1403 [06:07<00:00, 3.82it/s]
Epoch [12/15], Loss: 0.0593, Accuracy: 98.00%
100%|██████████| 1403/1403 [06:08<00:00, 3.81it/s]
Epoch [13/15], Loss: 0.0490, Accuracy: 98.41%
100%|██████████| 1403/1403 [06:07<00:00, 3.82it/s]
Epoch [14/15], Loss: 0.0465, Accuracy: 98.39%
100%|██████████| 1403/1403 [40:10<00:00, 1.72s/it]
Epoch [15/15], Loss: 0.0411, Accuracy: 98.65%

```

Figure 6. Accuracy of the ResNet101 model improvement at each step of training.

The results of the detection test accuracy of the ResNet101 model reached 91.63%, as shown in Figure 7, compared to 92.06% in the case of the ResNet50 model.

Test Accuracy: 91.63%

Figure 7. The final accuracy of the ResNet101 model using an online image library.

The training of the model was successful; it had a low error and a high training accuracy, presenting a testing accuracy on real images of 95%. This low testing accuracy is determined by the more complex dataset used in training with more complex features in the training images. Even if the testing accuracy determined during training was not that high, the generated model performed well during testing, with real, previously unused images.

4.3. Results Using the ResNet101 Training Model on Real Images Captured by the Research Team

The team chose the ResNet101 model because it had better results when applied to real-life images; the team used it on a new dataset that consisted of real images captured by the research team. The dataset had the same structure as the online image library, meaning that there were two parts, one consisting of images of walls with no defects and the other containing images of walls with various surface defects such as cracks or holes. This practice is also used in Yusnur Muhtar et al.'s paper, where for each image of a handwritten signature, half of the images contain good samples and the other half contain forged samples [31].

Because the dataset was smaller than the online image library, with a total of 800 images, the research team decided to run the training for 40 steps, which added up to over 200 min of training.

As expected, the accuracy of training increased with each step, reaching 99.38%, as can be observed in Figure 8.

```

Epoch [26/40], Loss: 0.0834, Accuracy: 97.03%
100%|██████████| 20/20 [00:33<00:00, 1.68s/it]
Epoch [27/40], Loss: 0.0998, Accuracy: 96.41%
100%|██████████| 20/20 [00:33<00:00, 1.69s/it]
Epoch [28/40], Loss: 0.0713, Accuracy: 97.19%
100%|██████████| 20/20 [00:34<00:00, 1.72s/it]
Epoch [29/40], Loss: 0.0881, Accuracy: 96.25%
100%|██████████| 20/20 [00:33<00:00, 1.69s/it]
Epoch [30/40], Loss: 0.1700, Accuracy: 93.59%
100%|██████████| 20/20 [00:33<00:00, 1.68s/it]
Epoch [31/40], Loss: 0.1086, Accuracy: 95.78%
100%|██████████| 20/20 [00:33<00:00, 1.69s/it]
Epoch [32/40], Loss: 0.0727, Accuracy: 97.50%
100%|██████████| 20/20 [00:33<00:00, 1.69s/it]
Epoch [33/40], Loss: 0.0538, Accuracy: 97.66%
100%|██████████| 20/20 [00:33<00:00, 1.68s/it]
Epoch [34/40], Loss: 0.0392, Accuracy: 98.44%
100%|██████████| 20/20 [00:33<00:00, 1.69s/it]
Epoch [35/40], Loss: 0.0377, Accuracy: 98.75%
100%|██████████| 20/20 [00:33<00:00, 1.69s/it]
Epoch [36/40], Loss: 0.0803, Accuracy: 96.72%
100%|██████████| 20/20 [00:33<00:00, 1.69s/it]
Epoch [37/40], Loss: 0.0406, Accuracy: 98.28%
100%|██████████| 20/20 [00:33<00:00, 1.69s/it]
Epoch [38/40], Loss: 0.0640, Accuracy: 98.91%
100%|██████████| 20/20 [00:33<00:00, 1.69s/it]
Epoch [39/40], Loss: 0.0404, Accuracy: 98.75%
100%|██████████| 20/20 [00:33<00:00, 1.69s/it]
Epoch [40/40], Loss: 0.0208, Accuracy: 99.38%

```

Figure 8. Accuracy of the ResNet101 model on the new dataset.

In this case, the graphs of the training loss and training accuracy do not follow a smooth path like that in the previous cases. As can be seen in Figure 9, both graphs vary slightly during the training period, but both tend towards a general path of improvement over time. The fluctuation may be caused by the quality of the images captured by the team, showing the importance of the quality and reliability of the pictures being captured and the need to take into account natural factors, such as lighting, haze, shadows, etc., as mentioned by Hongpan Lin et al. in their paper [32].

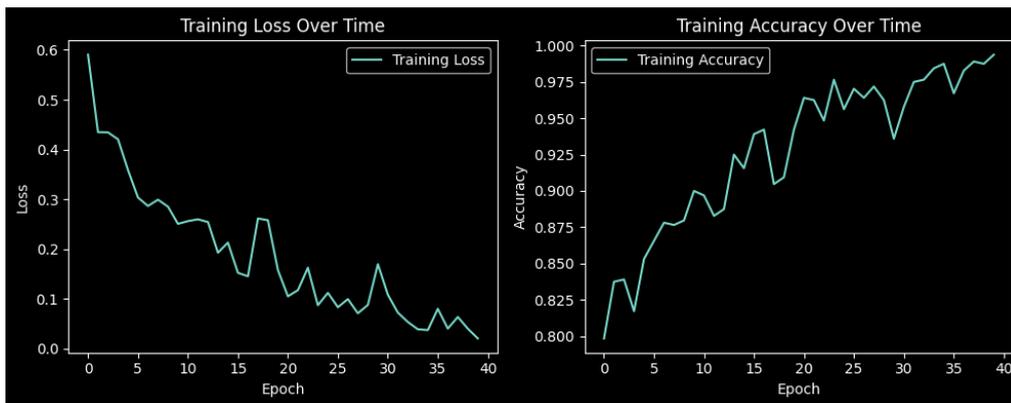


Figure 9. Training loss and training accuracy using ResNet101 model on real images captured by the research team.

The results of the detection test accuracy were not as high as expected, reaching 95%, as shown in Figure 10, but the research team has a strong belief that the results could be improved if the dataset was refined in the future and contained only clear images.

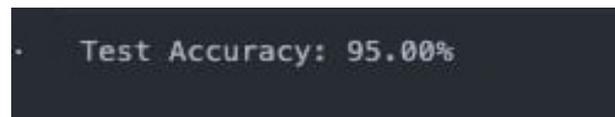


Figure 10. The final accuracy of the ResNet101 model using real images captured by the research team.

On the other hand, this application is meant to be used in a real construction environment where the conditions are not always perfect; thus, not-so-clear images were introduced into the dataset to simulate the real-life conditions found on the working sites.

Some tests were made with the final trained model using real images captured by the research team on real construction sites. The results can be seen in Figures 11–14, where the application correctly identifies an image with or without any cracks/defects and knows how to detect the difference between a crack and a shadow.

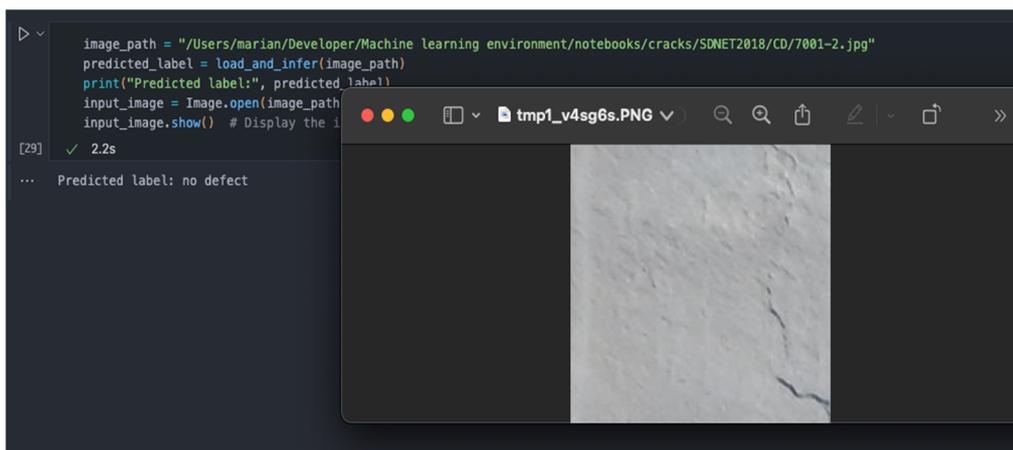


Figure 11. Differentiation between a shadow and a crack.

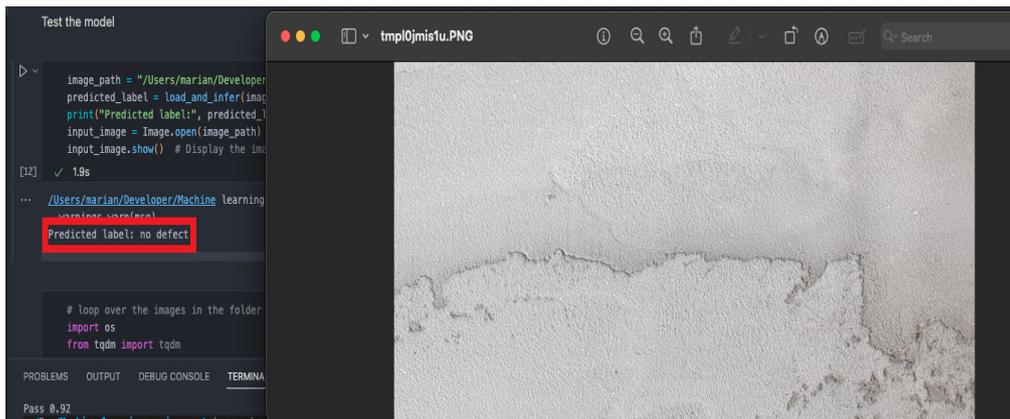


Figure 12. Differentiation between a stain and a crack.

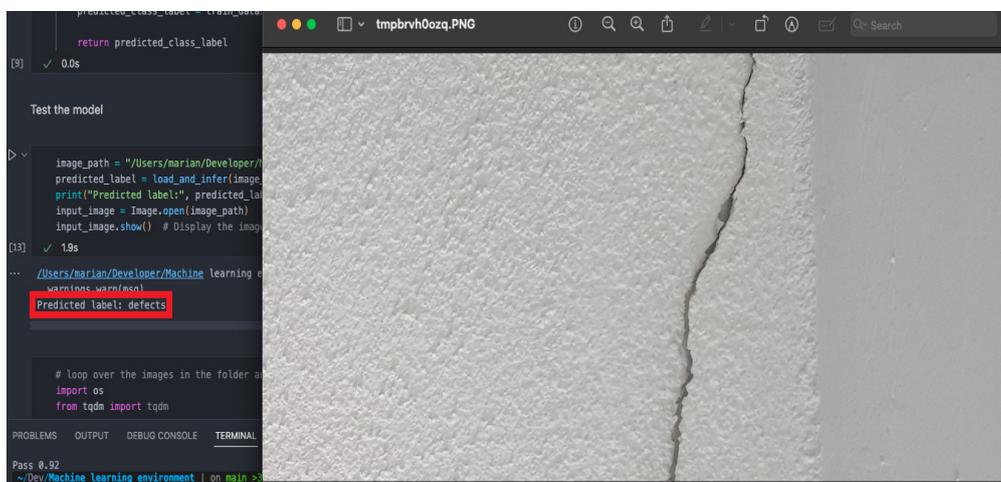


Figure 13. Crack detection using ResNet101 on a real image captured by the research team.

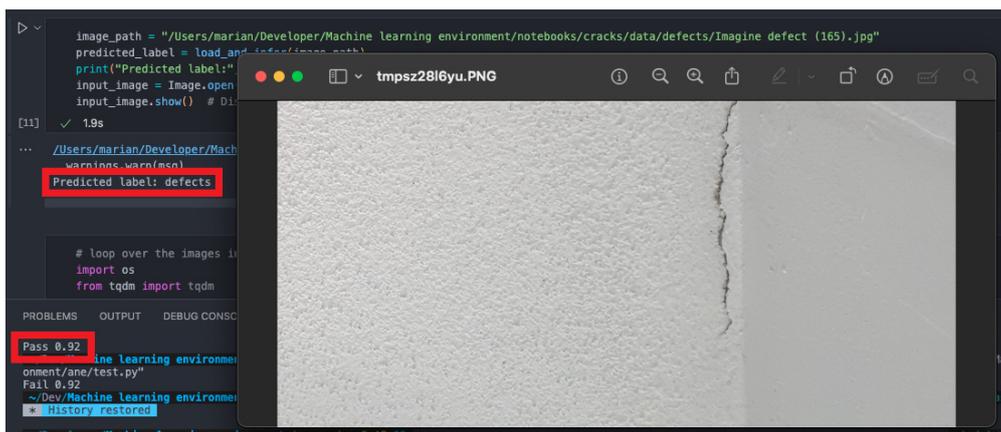


Figure 14. The accuracy of detection reaches 92% in real conditions using ResNet101.

The difference between a simple image-processing algorithm and the machine learning solution used in our paper can be observed in Figures 11 and 12. A simple image-processing algorithm mainly uses edge detection to detect cracks; therefore, such an approach could be deceived by features that are similar to cracks (i.e., shadows or stains), and this could lead to fake positives. Our model uses a set of labelled crack images, with which it learns to differentiate between similar features; therefore, as it can be observed being highlighted in

the red box, it correctly predicts that the stain and the shadow do not represent a structural defect (i.e., a crack).

If the labelling of defects is performed properly in the first steps of training, the model can differentiate between different types of surface damage according to the labelling on which it was trained. This feature makes the model a versatile one that can be adapted to meet the needs of each construction site.

Shown above in Figures 13 and 14 and highlighted in the red boxes are the results of the approach chosen by the research team. The solution (i.e., the ResNet101 training model) can correctly predict structural defects with high accuracy (i.e., 92%) on real images, and as shown earlier, its accuracy does not decrease when confronted with naturally occurring factors on a construction site, such as stains or shadows, which proves that it can be used for future work in this domain.

4.4. Further Testing on a New Set of Real Images Captured by the Research Team

For further testing and the checking of the trained model, the research team captured a new set of approximately 230 more real-life images. Out of these, 56 of them presented walls with defects such as cracks or paint chips and the other 176 presented walls with no defects.

The team adopted a three-step procedure for this test: first, the model was tested only using the 176 images with no defects; then, it was tested with the 56 images that presented surface defects; lastly, the team used the entire set of 230+ images, both with and without defects, to test the model.

For the first step, just as predicted, the model passed all 176 images with no defects, as shown in Figure 15. Based on these positive results, the team is confident that the model is able to identify a picture of a wall with no cracks or defects.



Figure 15. Tests run on the images of walls with no defects.

For the second step, the model failed all 56 images that presented surface defects, as can be observed in Figure 16. Therefore, we can state that in these controlled conditions, the model can correctly identify an image with a wall that presents faults such as cracks, holes, or other similar surface defects.

For the last step, the model correctly identified both types of images (with and without defects) with high accuracy, as shown in Figure 17. The research team is confident that in laboratory conditions, where the images are captured in good lighting conditions and without noise, that the model is capable of correctly differentiating between a wall with no defects and a wall that presents cracks, holes, paint chips, etc.



Figure 16. Tests run on the images of walls with surface defects.



Figure 17. Tests run on both sets of images simultaneously.

The results of the two models tested, using both an online image library and real images captured by the research team, are presented in Table 2. It can be observed that using real images that are captured in a controlled way with a high-resolution camera can not only improve the training and testing accuracy but can also mitigate the loss during the training.

Table 2. Model results.

Model	Loss	Train Accuracy	Test Accuracy	Epochs
ResNet50 online img. library	0.0892	96.94%	92.06%	10
ResNet101 online img. library	0.0411	98.65%	91.63%	15
ResNet101 on real images	0.0208	99.38%	95.00%	40

5. Conclusions

The team mentioned in this paper was made up of researchers from the University of Craiova specializing in the field of Mechatronics and Robotics. Based on previous work, the team developed an application with the aim of transforming the inspection of buildings into a semi-automated task with the help of machine learning. To make the entire system more affordable and cost-effective, the team made use of readily available hardware such as a ZED camera and a standard Linux OS-based laptop/desktop. This strategy was meant to improve the availability of defect detection tools in the construction domain.

In conclusion, even if the ResNet50-based model performed well during training and tests, it was unable to provide satisfying results when used on real images captured by the research team as the images presented unknown inferences. The ResNet10-based model was able to better classify real images and provided better results that can be used in practical applications, as can be seen in Table 2.

A large part of the population now tends to live in cities, and the expansion of urban areas is expected to grow in the future [33]. Therefore, the need for such an application in the future will become more and more present. For future work, the research team foresees the possibility of integrating their progress in the controlling of robotic platforms acquired throughout the previous research papers with the progress presented in the current work in the domain of defect detection in construction.

There are multiple variants of improvement, but the team focuses on developing an integrated automatic detection solution composed of an autonomous robotic platform together with the crack detection system. This solution would be optimal for indoor automatic inspection, which is meant to autonomously scan a building's interior floor by floor.

For the outside of the building, especially in the case of a multi-story building, the research team tends to use a different solution based on an autonomous or semi-autonomous drone that would be the carrier of the crack detection system. Chen et al. also proposed a detection technique using drones to detect cracks on the surface of the buildings [34]. Based on the ideas presented in Hyunkyun Shin et al.'s paper, a low-noise drone is required for use in residential areas; the drone should still be capable of reaching high altitudes with the additional equipment necessary for crack detection in the case of high-rise buildings [35]. Two solutions are considered for the control of the drone: it should be fully autonomous, or it should have a predefined path; in the latter case, the drone would encircle the building story by story or move according to an up-down method.

Author Contributions: Methodology, D.C.; Software, M.M.A.; Validation, A.M.M. and D.C.; Formal analysis, D.C.; Investigation, A.M.M. and M.M.A.; Resources, M.M.A.; Data curation, A.M.M. and M.M.A.; Writing—original draft, A.M.M.; Supervision, D.C. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Data are contained within the article.

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

References

1. Mariniuc, A.M.; Cojocaru, D.; Manta, L.F.; Dragomir, A.; Abagiu, M. Using 3D Scanning Techniques from Robotic Applications in the Constructions Domain. In Proceedings of the 25th International Conference on System Theory, Control and Computing (ICSTCC), Sinaia, Romania, 19–21 October 2022; pp. 170–175, ISBN 978-1-6654-6745-2. [\[CrossRef\]](#)
2. Chen, F.; Jahanshahi, M.R. NB-CNN: Deep Learning-Based Crack Detection Using Convolutional Neural Network and Naive Bayes Data Fusion. *IEEE Trans. Ind. Electron.* **2018**, *65*, 4392–4400. [\[CrossRef\]](#)
3. Pan, Y.; Zhang, G.; Zhang, L. A spatial-channel hierarchical deep learning network for pixel-level automated crack detection. *Autom. Constr.* **2020**, *119*, 103357. [\[CrossRef\]](#)
4. Bacharidis, K.; Sarri, F.; Ragia, L. 3D Building Façade Reconstruction Using Deep Learning. *ISPRS Int. J. Geo-Inf.* **2020**, *9*, 322. [\[CrossRef\]](#)
5. Rocha, G.; Mateus, L.; Fernández, J.; Ferreira, V. A Scan-to-BIM Methodology Applied to Heritage Buildings. *Heritage* **2020**, *3*, 47–67. [\[CrossRef\]](#)
6. Mihić, M.; Sigmund, Z.; Završki, I.; Butković, L.L. An Analysis of Potential Uses, Limitations and Barriers to Implementation of 3D Scan Data for Construction Management-Related Use—Are the Industry and the Technical Solutions Mature Enough for Adoption? *Buildings* **2023**, *13*, 1184. [\[CrossRef\]](#)
7. Manta, L.F.; Dumitru, S.; Cojocaru, D. Computer Vision Techniques for Collision Analysis. A Study Case. In Proceedings of the 22nd International Conference on System Theory, Control and Computing, Sinaia, Romania, 10–12 October 2018; pp. 427–432, ISBN 978-153864444-7.
8. Liang, H.; Fu, W.; Yi, F. A Survey of Recent Advances in Transfer Learning. In Proceedings of the 2019 IEEE 19th International Conference on Communication Technology (ICCT), Xi'an, China, 16–19 October 2019; pp. 1516–1523.

9. Dini, S.F.; Wibowo, E.P.; Iqbal, M.; Bahar, Y.N.; Alfiandy, A. Applying Deep Learning and Convolutional Neural Network System to Identify Historic Buildings: The “Little China” Building in Central Java, Indonesia. *ISVS E-J.* **2023**, *10*, 187–200.
10. Kou, X.; He, Y.; Qian, Y. An improvement and application of a model conducive to productivity optimization. In Proceedings of the 2021 IEEE International Conference on Power Electronics, Computer Applications, ICPECA 2021, Shenyang, China, 22–24 January 2021; pp. 1050–1053.
11. Katsigiannis, S.; Seyedzadeh, S.; Agapiou, A.; Ramzan, N. Deep learning for crack detection on masonry façades using limited data and transfer learning. *J. Build. Eng.* **2023**, *76*, 107105. [\[CrossRef\]](#)
12. Singh, G.H.; Matthews, A.; Tea, R.; George, K. LIDAR-based autonomous wheelchair. In Proceedings of the 2017 IEEE Sensors Applications Symposium (SAS), Glassboro, NJ, USA, 13–15 March 2017; pp. 1–6.
13. Mendes, C.P.; Lim, N.T. EcoLiDAR: An economical LiDAR scanner for ecological research. In Proceedings of the International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences, Volume XLVIII-1/W1-2023, 12th International Symposium on Mobile Mapping Technology (MMT2023), Padua, Italy, 24–26 May 2023. [\[CrossRef\]](#)
14. STEREO LABS. Available online: <https://www.stereolabs.com/zed-2/> (accessed on 19 August 2023).
15. Peralta-López, J.-E.; Morales-Viscaya, J.-A.; Lázaro-Mata, D.; Villaseñor-Aguilar, M.-J.; Prado-Olivarez, J.; Pérez-Pinal, F.-J.; Padilla-Medina, J.-A.; Martínez-Nolasco, J.-J.; Barranco-Gutiérrez, A.-I. Speed Bump and Pothole Detection Using Deep Neural Network with Images Captured through ZED Camera. *Appl. Sci.* **2023**, *13*, 8349. [\[CrossRef\]](#)
16. Cojocar, D.; Manta, L.F.; Pană, C.F.; Dragomir, A.; Mariniuc, A.M.; Vladu, I.C. The design of an intelligent robotic wheelchair supporting people with special needs, including for their visual system. *Healthcare* **2022**, *10*, 13. [\[CrossRef\]](#)
17. VELODYNE'S PUCK, Lidar Sensor-Technical Data. Available online: <https://velodynelidar.com/products/puck/> (accessed on 23 November 2023).
18. Kucak, R.A.; Erol, S.; Isiler, M. Comparative Accuracy Analysis of Lidar Systems. *Turk. J. LIDAR* **2020**, *2*, 30–34.
19. Eligüzel, N.; Çetinkaya, C.; Dereli, T. Comparison of different machine learning techniques on location extraction by utilizing geo-tagged tweets: A case study. *Adv. Eng. Inform.* **2020**, *46*, 101151. [\[CrossRef\]](#)
20. Tulbure, A.A.; Tulbure, A.A.; Dulf, E.H. A review on modern defect detection models using DCNNs—Deep convolutional neural networks. *J. Adv. Res.* **2021**, *35*, 33–48. [\[CrossRef\]](#)
21. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
22. He, H.; Yuan, M.; Liu, X. Research on Surface Defect Detection Method of Metal Workpiece Based on Machine Learning. In Proceedings of the 2021 IEEE 6th International Conference on Intelligent Computing and Signal Processing, ICSP 2021, Xi'an, China, 9–11 April 2021; pp. 881–884.
23. Shu, Y.F.; Li, B.; Li, X.; Xiong, C.; Cao, S.; Wen, X.Y. Deep learning-based fast recognition of commutator surface defects. *Measurement* **2021**, *178*, 109324. [\[CrossRef\]](#)
24. Xu, Y.; Zhang, K.; Wang, L. Metal Surface Defect Detection Using Modified YOLO. *Algorithms* **2021**, *14*, 257. [\[CrossRef\]](#)
25. Gai, X.; Ye, P.; Wang, J.; Wang, B. Research on Defect Detection Method for Steel Metal Surface based on Deep Learning. In Proceedings of the 2020 IEEE 5th Information Technology and Mechatronics Engineering Conference, ITOEC 2020, Chongqing, China, 12–14 June 2020; pp. 637–641.
26. Ooi, J.; Tay, L.C.; Lai, W.K. Bottom-hat filtering for Defect Detection with CNN Classification on Car Wiper Arm. In Proceedings of the 2019 IEEE 15th International Colloquium on Signal Processing and Its Applications, CSPA 2019, Penang, MA, USA, 8–9 March 2019; pp. 90–95.
27. Macarulla, M.; Forcada, N.; Casals, M.; Gangolells, M.; Fuertes, A.; Roca, X. Standardizing housing defects: Classification, validation, and benefits. *J. Constr. Eng. Manag.* **2013**, *139*, 968–976. [\[CrossRef\]](#)
28. Lin, H.; Huang, L.; Chen, Y.; Zheng, L.; Huang, M.; Chen, Y. Research on an Application of CGAN in the Design of Historic Building Facades in Urban Renewal—Taking Fujian Putian Historic Districts as an Example. *Buildings* **2023**, *13*, 1478. [\[CrossRef\]](#)
29. Cichy, R.M.; Kaiser, D. Deep Neural Networks as Scientific Models. *Trends Cogn. Sci.* **2019**, *23*, 305–317. [\[CrossRef\]](#) [\[PubMed\]](#)
30. Mosavi, A.; Faizollahzadeh Ardabili, S.R.; Várkonyi-Kóczy, A. List of Deep Learning Models. *Preprints* **2019**. [\[CrossRef\]](#)
31. Muhtar, Y.; Muhammat, M.; Yadikar, N.; Aysa, A.; Ubul, K. FC-ResNet: A Multilingual Handwritten Signature Verification Model Using an Improved ResNet with CBAM. *Appl. Sci.* **2023**, *13*, 8022. [\[CrossRef\]](#)
32. Xu, H.; Chang, R.; Pan, M.; Li, H.; Liu, S.; Webber, R.J.; Zuo, J.; Dong, N. Application of Artificial Neural Networks in Construction Management: A Scientometric Review. *Buildings* **2022**, *12*, 952. [\[CrossRef\]](#)
33. Berry, B.J.; Marzluff, J.M. Urban Ecology. In *Proceedings of the Urban Ecology: An International Perspective on the Interaction between Humans and Nature*; Springer: Berlin/Heidelberg, Germany, 2008; pp. 25–48.
34. Chen, K.; Reichard, G.; Xu, X.; Akanmu, A. Automated crack segmentation in close-range building façade inspection images using deep learning technique. *J. Build. Eng.* **2021**, *43*, 102913. [\[CrossRef\]](#)
35. Shin, H.; Kim, J.; Kim, K.; Lee, S. Empirical Case Study on Applying Artificial Intelligence and Unmanned Aerial Vehicles for the Efficient Visual Inspection of Residential Buildings. *Buildings* **2023**, *13*, 2754. [\[CrossRef\]](#)

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.