BUILDING FACADE INSPECTION: A SYSTEM BASED ON AUTOMATED DATA ACQUISITION, MACHINE LEARNING, AND DEEP LEARNING IMAGE CLASSIFICATION METHODS

Gabriel B. Bouzan, Paulo F. P. C. Fazzioni, Renata G. Faisca and Carlos A. P. Soares  
Pós-Graduação em Engenharia Civil, Universidade Federal Fluminense, Niterói, Brazil  
E-Mail: renatafaisca@id.uff.br

ABSTRACT

Building facade inspection usually uses time-consuming, expensive data collection processes, with a risk level above the desirable, mainly because it is based on visual inspection. In this context, studies that address the acquisition of automated data and the post-processing of digital images have increasingly aroused researchers’ interest. In this work, we developed a low-cost system for identifying and classifying building facades pathologies formed by a drone for capturing high-resolution images and a web application containing three components: a) a set of machine learning and deep learning algorithms for the pathologies classification; b) a dataset of façade pathologies; c) an interface that allows an environment of interaction and selection of information that contributes to the user's decision-making process. To select the best method for identifying the type of pathology, we also performed a comparative analysis of algorithms effectiveness that uses decision trees, random forests, and convolutional neural networks. The results contribute to improving inspection processes on buildings facades and reducing costs and the risk of accidents.

Keywords: facade pathologies; facade inspection; decision trees; random forests; convolutional neural networks; unmanned aerial vehicles.

1. INTRODUCTION

Facade pathologies cause severe property damage, in addition to economic devaluation and environmental risks, considerably reducing the building’s durability and useful life [1]. Identifying these anomalies is essential for assessing the degree of facade deterioration and, consequently, improving decision-making on the best intervention strategy.

Facade inspections usually require equipment such as scaffolding and stairs in places with difficult access and a risk level above the desirable. Besides, the inspection process is time-consuming, expensive, and requires a lot of human effort, mainly because it is based on visual inspection. In this context, Eschman et al. [2] addressed the building inspection methodology using data processing and drones, reducing human participation in activities of high complexity and risk, as well as the costs associated with these activities. Thus, what is usually done by visual inspection and processing of digitized photographic material can be replaced by automated data acquisition processes and post-processing of digital images.

In automated data acquisition processes, drones use has proved to be an interesting strategy due to its ability to collect large amounts of high-resolution images [3] efficiently and with low cost and risk. Also, techniques and models based on artificial intelligence (AI) [4] and its subsets, such as machine learning (ML) [5], deep learning (DL) [6], and computer vision (CV) [7], has proven to be essential tools for improving image recognition and increasing the efficiency of data processing and analysis. However, the literature on façade pathology inspection still lacks studies that address its automation, especially concerning increasing efficiency and attractiveness. To collaborate to filling this gap, in this study, we propose a low-cost system for identifying and classifying building facades pathologies formed by a drone for capturing high-resolution images and a web application containing three components: a) a set of machine learning and deep learning algorithms for the pathologies classification; b) a dataset of façade pathologies; c) an interface that allows an environment of interaction and selection of information that contributes to the user's decision-making process.

A crucial element for the effectiveness of the data processing result is the method used for image recognition. To select the best method for identifying the pathology type, we also performed a comparative analysis of the effectiveness of algorithms based on decision trees [8], random forests [9], and convolutional neural networks [10].

Facade inspections, as they are expensive, are usually relegated to the background. The most common situation is when pathological manifestations are in huge quantity and require corrective maintenance at an advanced stage. In this context, this work also collaborates so that inspections can become more routine by proposing an effective, low-cost, and easy-to-operate system. This work also contributes to reducing accident risks during the data collection process and the costs associated with building maintenance since preventive maintenance has lower costs than corrective ones.

2. FACADE PATHOLOGY RECOGNITION SYSTEM

For the system development, we designed an approach in six steps: Drone Selection, Processing device definition, Datasets construction, Algorithms elaboration, Selection of the best method for pathology type identification, and web application development.
2.1 Drone Selection

The adopted strategy was to use low-cost equipment, with few legal operating restrictions in Brazil. The National Civil Aviation Agency (ANAC) [11] does not require licensing of remotely piloted aircraft (RPA) up to 250g. However, the operator must comply with the airspace usage rules issued by the Airspace Control Department (DECEA) [12] and the requirements issued by the National Telecommunications Agency (ANATEL) [13] and the Ministry of Defense (MD) [14]. Thus, we selected a quadcopter type drone, model DJI Ryz Tello [15], featured in Table 1.

<table>
<thead>
<tr>
<th>Specifications</th>
<th>DJI Ryze Tello</th>
</tr>
</thead>
<tbody>
<tr>
<td>Weight</td>
<td>Approximately 80 g (Propellers and Battery Included)</td>
</tr>
<tr>
<td>Dimensions</td>
<td>98×92.5×41 mm</td>
</tr>
<tr>
<td>Propeller</td>
<td>3 inches</td>
</tr>
<tr>
<td>Built-in Functions</td>
<td>Range Finder, Barometer, LED, Vision System, 2.4 GHz 802.11n Wi-Fi, 720p Live View</td>
</tr>
<tr>
<td>Port</td>
<td>Micro USB Charging Port</td>
</tr>
<tr>
<td>Detachable Battery</td>
<td>1.1Ah/3.8V</td>
</tr>
<tr>
<td>Max Flight Distance</td>
<td>100m</td>
</tr>
<tr>
<td>Max Speed</td>
<td>8m/s</td>
</tr>
<tr>
<td>Max Flight Time</td>
<td>13min</td>
</tr>
<tr>
<td>Max Flight Height</td>
<td>30m</td>
</tr>
<tr>
<td>Built-in Camera</td>
<td>Yes</td>
</tr>
<tr>
<td>Photo</td>
<td>5MP (2592×1936)</td>
</tr>
<tr>
<td>FOV</td>
<td>82.6°</td>
</tr>
<tr>
<td>Video</td>
<td>HD720P30</td>
</tr>
<tr>
<td>Format</td>
<td>JPG(Photo); MP4(Video)</td>
</tr>
<tr>
<td>EIS</td>
<td>Yes</td>
</tr>
</tbody>
</table>

The DJI Ryze Tello Drone supports the development of artificial intelligence applications. It has embedded firmware with advanced commands, improved data interfaces, supports electronic image stabilization and Python programming [16]. Access to video stream data allows image processing by performing AI functions, such as object recognition, tracking, 3D reconstruction through programming, computer vision, and deep machine learning technologies.

The selected drone operates with software embedded in smartphones and algorithms developed on open source platforms. The drone's control and the capture of images in flight are carried out using software designed for a smartphone-compatible with the Android operating system [17]. They enable drone control operations, reading sensors embedded drone, images capturing using a high definition RGB camera [18], and georeferencing information managing.

2.2 Processing Device Definition

The minimum requirements for image processing and execution of ML and DL algorithms comprise hardware capable of supporting large computational operations in memory and processing. It is essential to consider portability, processing speed, graphics processing capacity, and compatibility between hardware and ML and DL libraries. We recommend that the hardware has at least the following characteristics:

- Central Processing Unit (CPU) - a minimum of 7th generation (Intel Core i7 processor [19]). It is used in low-computing machine learning tasks that can be easily manipulated through complex sequential processing without the need for a GPU.
- AI-based Graphics Processing Unit (GPU), such as CuDNN [20], and parallel computing APIs [21] such as NVIDIA's CUDA [22]. These frameworks and APIs take advantage of GPU parallelism for deep learning tasks, such as TensorFlow [23] and PyTorch [24]. GPUs are microprocessor chips developed primarily to handle graphics. GPUs have become popular in deep learning due to their ability to handle simultaneous calculations faster than CPUs.
- Memory equal to or higher than 16 GB of RAM is recommended for most deep learning tasks.
- Solid State Device (SSD) storage is due to the increasing size of deep learning data sets requiring higher storage capacity. It is also recommended for its speed and efficiency.

2.3 Datasets Construction

For the dataset construction, we used images obtained during the authors professional performance, images obtained on the Internet using the web scraping technique, and contributions from the works of Mundt et al. [25] and Hüthwohl et al. [26]. Pathology models were trained in four classes called: deepcrack, efflorescence, reinforcement, and ruststain. Figure-1 presents examples of these classes. The dataset consists of 1000 images with different resolutions, with 250 images for each class.
2.4 Algorithms Elaboration

The algorithms and tools using AI were developed in Python programming language, with the following packages and CV libraries:

a) **OpenCV [27]**: high-efficiency computer vision library that facilitates the processing of images in real-time;

b) **Scikit-learn [28]**: application programming interface (API) for machine learning in Python;

c) **Scikit-image [29]**: a collection of algorithms for image processing used in addition to OpenCV;

d) **TensorFlow**: open-source machine learning library, used in neural networks (NN) [30], deep learning, and as a computational backend for API Keras [31];

e) **Keras**: application programming interface of high-performance neural networks. They were written in Python, offering integration with TensorFlow.

In this step, we train ML and DL learning models with Python to recognize pathologies images. The ML model creation using the Python programming language was systematically aligned with the automated inspection approach to building facades. To select the best method for identifying the type of pathology, we compared the results of using algorithms based on decision trees (DT) [32], random forests (RF) [33], and convolutional neural networks (CNN) [34].

2.5 Selection of the Best Method for the Pathology Type Identification

Initially, we extracted the data set characteristics from the four pathologies categories. To test each method (decision trees, random forests, and convolutional neural networks - CNN), we use training and test sets made up of data and labels. 75% of this set was used for the algorithm's training, and 25% was used for testing.

The decision tree classifier was built based on Scikit Learn DT parameters [35]. The image classifier in random forests was also built based on Scikit Learn RF parameters [36] and used 20 decision trees.

For the classifier construction using Convolutional Neural Networks, we import convolutional layers, maximum pooling operations, different types of activation and leveling functions, Adam optimizer, and Scikit-Learn processing selection models [37]. Instead of using a vector of color statistical characteristics, we operate on the pixels themselves. Thus, we add the resized and scaled images to the data list, scaling to the vector range and dividing by the maximum value of one pixel. We used the Scikit Label Binarizer [38] to encode the labels, which were converted into character strings of

![Figure-1. Examples of dataset classes.](image-url)
whole numbers. The CNN image was classified with Keras, using TensorFlow as a backend.

Table-2 presents the pathology classification report of the DT, RF, and CNN algorithms issued by the Scikit learn [39] tool, which summarizes the class’s main classification metrics. The metrics were defined in terms of true and false positives and true and false negatives. A true positive occurs when a real class and an estimated class are positive. A false positive occurs when a real class is negative, but the estimated class is positive.

Table-2. Pathology classification report for DT, RF, and CNN algorithms.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Pathology</th>
<th>Precision</th>
<th>Recall</th>
<th>F1-Score</th>
<th>Support</th>
</tr>
</thead>
<tbody>
<tr>
<td>DT</td>
<td>Deepcrack</td>
<td>0.67</td>
<td>0.75</td>
<td>0.71</td>
<td>60.00</td>
</tr>
<tr>
<td>DT</td>
<td>Efflorescence</td>
<td>0.53</td>
<td>0.55</td>
<td>0.54</td>
<td>62.00</td>
</tr>
<tr>
<td>DT</td>
<td>Reinforcement</td>
<td>0.62</td>
<td>0.63</td>
<td>0.62</td>
<td>67.00</td>
</tr>
<tr>
<td>DT</td>
<td>Ruststain</td>
<td>0.71</td>
<td>0.59</td>
<td>0.64</td>
<td>61.00</td>
</tr>
<tr>
<td>RF</td>
<td>Deepcrack</td>
<td>0.68</td>
<td>0.82</td>
<td>0.74</td>
<td>60.00</td>
</tr>
<tr>
<td>RF</td>
<td>Efflorescence</td>
<td>0.55</td>
<td>0.61</td>
<td>0.58</td>
<td>62.00</td>
</tr>
<tr>
<td>RF</td>
<td>Reinforcement</td>
<td>0.76</td>
<td>0.55</td>
<td>0.64</td>
<td>67.00</td>
</tr>
<tr>
<td>RF</td>
<td>Ruststain</td>
<td>0.73</td>
<td>0.72</td>
<td>0.73</td>
<td>61.00</td>
</tr>
<tr>
<td>CNN</td>
<td>Deepcrack</td>
<td>0.84</td>
<td>0.78</td>
<td>0.81</td>
<td>63.00</td>
</tr>
<tr>
<td>CNN</td>
<td>Efflorescence</td>
<td>0.89</td>
<td>0.82</td>
<td>0.85</td>
<td>68.00</td>
</tr>
<tr>
<td>CNN</td>
<td>Reinforcement</td>
<td>0.65</td>
<td>0.64</td>
<td>0.64</td>
<td>55.00</td>
</tr>
<tr>
<td>CNN</td>
<td>Ruststain</td>
<td>0.71</td>
<td>0.83</td>
<td>0.76</td>
<td>64.00</td>
</tr>
<tr>
<td>DT</td>
<td>Micro Avg</td>
<td>0.63</td>
<td>0.63</td>
<td>0.63</td>
<td>250</td>
</tr>
<tr>
<td>DT</td>
<td>Macro Avg</td>
<td>0.63</td>
<td>0.63</td>
<td>0.63</td>
<td>250</td>
</tr>
<tr>
<td>DT</td>
<td>Weighted Avg</td>
<td>0.63</td>
<td>0.63</td>
<td>0.63</td>
<td>250</td>
</tr>
<tr>
<td>RF</td>
<td>Micro Avg</td>
<td>0.67</td>
<td>0.67</td>
<td>0.67</td>
<td>250</td>
</tr>
<tr>
<td>RF</td>
<td>Macro Avg</td>
<td>0.68</td>
<td>0.68</td>
<td>0.67</td>
<td>250</td>
</tr>
<tr>
<td>RF</td>
<td>Weighted Avg</td>
<td>0.68</td>
<td>0.67</td>
<td>0.67</td>
<td>250</td>
</tr>
<tr>
<td>CNN</td>
<td>Micro Avg</td>
<td>0.77</td>
<td>0.77</td>
<td>0.77</td>
<td>250</td>
</tr>
<tr>
<td>CNN</td>
<td>Macro Avg</td>
<td>0.77</td>
<td>0.77</td>
<td>0.77</td>
<td>250</td>
</tr>
<tr>
<td>CNN</td>
<td>Weighted Avg</td>
<td>0.77</td>
<td>0.77</td>
<td>0.77</td>
<td>250</td>
</tr>
</tbody>
</table>

- **Classes**: identification of each facade pathology class.
- **Precision**: measure the accuracy of a classifier as a positive occurrence.
- **Recall**: the measure of the classifier's integrity, which represents a classifier's ability to find all positive instances correctly.
- **F1 score**: weighted harmonic mean of precision and recall, so that the best score is 1.0 and the worst is 0.0.
- **Support**: number of actual occurrences of the class in the specified data set.
- **Micro average**: the sum of dividends and dividers that compose the metrics by class to calculate a general quotient. Instead of adding the metric by class, the micro avg provides each sample-class pair with an equal contribution to the overall metric [40].
- **Macro average**: calculates the average of the binary metric, assigning equal weight to each class.
- **Weighted average**: is responsible for the class imbalance, from calculating the average of the binary metric. The score of each class is weighted by its presence in the real data sample.

Figure-2 summarizes the classification accuracy by pathology type and algorithm type. Regarding deep crack, rust stain, and efflorescence pathologies, the DT and RF type algorithms presented similar precision values for each pathology, lower than the CNN. Regarding the reinforcement pathology, the DT and RF algorithms also showed similar precision values. However, when using the RF algorithm, the precision reached values slightly above the previous ones, disagreeing with the previous results trends.
Figure 2. Precision of DT, RF, and CNN algorithms for the classification of deep crack, rust stain, efflorescence, and reinforcement pathologies.

Figure 3 shows the average precision of the three algorithms considering all classified pathologies. The CNN algorithm was more accurate than the others and, therefore, used in experimental validation.

Figure 3. Average precision of algorithms.

2.6 Web Application Development

The BFIS web application consists of three modules. The first module extracts the images set characteristics to be analyzed, consolidating them in a spreadsheet. The second module uses this spreadsheet to identify the pathologies types through the algorithm that uses Convolutional Neural Networks. The third performs the user interface, enabling an environment of information interaction and selection that contributes to the decision-making process. As a low-cost proposal, the BFIS was developed with dynamic web pages using a local web server, eliminating extra expenses with specialized website hosting. Figure 4 shows BFIS's initial dashboard. In the next section, you can view the BFIS functions and reports.

Figure 4. BFIS dashboard.

3. SYSTEM EVALUATION

To evaluate the system, we used two buildings known as Block D and E of the Praia Vermelha Campus of Universidade Federal Fluminense, located in the municipality of Niterói in the state of Rio de Janeiro, Brazil. The "D" building was built 27 years ago, has five floors, and the following dimensions: 23 m wide, 116 m length and 25 m in height. The "E" building was built 46
years ago, has four floors, and the following dimensions: 16 m wide, 106 m length and 18 m in height. Figure-5 shows the building’s location using the Google Earth Pro tool [41].

The inspections took place between June and September 2020. We identified the number and position of trees, shrubs, gardens, transmission lines, and power poles from the site inspection. Besides, we identified the temperature and climate conditions on the inspection days and the access of people and vehicle traffic, which during the inspection days were almost non-existent.

During the Proof of Concept (POC), we use the following equipment and applications:

- DJI Ryze Tello drone;
- High precision 3D joystick control, T1S GameSir model [42];
- Drone flight control software, Tello FPV [43];
- Samsung Galaxy S7 Edge Smartphone [44];
- Dell Inspiron 7580 notebook [45] with the following configuration: Intel Core i7-8565U CPU CPU @ 1.80GHz 1.99GHz; Memory 16 GB RAM BCC 16GB (2x8GB) 2666MHz DDR4; NVIDIA GeForce MX150 2GB GDDR5 GPU;
- ML and DL algorithms, developed in Python programming language, executed in a virtualized environment running on Linux Ubuntu 18 system [46].

Using the drone to collect the images, we scan the facades from bottom to top and from left to right, as shown in Figure-6. To capture images with adequate quality, in addition to the proper positioning of the drone, we adopted an overlapping of consecutive scan images.
The operating time had to be optimized since the battery of this type of drone has an average duration of 13 minutes. In this time limitation scenario, we use four batteries, considering an average time for recharging in batches of approximately 2 hours. Figure-7 shows the BFIS Dashboard containing the coordinates identification map of Blocks D and E. On the dashboard left side is the pathologies selections panel per facade that allows reports with information filtering considering the pathology type and the building's location, and the facade to be analyzed.

Each measurement point is georeferenced and represented on a map, thus facilitating the pathologies visualization in the evaluated paths. The locations with the highest or least need for repair or maintenance prevention are identified, as illustrated in Figures 7 and 8.

Figure-7. Coordinates identification map of blocks D and E.

Figure-8. Coordinate identification map for blocks D and E.

Figure-9 shows the dashboard with pathologies quantification. The results show that Block D contains 88.4% of the identified pathologies.
Figure-9. Pathologies quantification of blocks D and E.

Figure-10 shows the dashboard with pathologies quantification of per facade. The results show that in Block D, the pathologies occurred mainly in the north and south facades.

Figure-10. Pathologies quantification by facades in blocks D and E.

Around the west facade of Block D, we identified many manifestations of efflorescence and rust stain due to many trees and proximity to the slope (Figures 9 and 10). Figures 11 and 12 show the dashboard with the quantification of each pathology type. The results show a higher incidence of reinforcement and rust stain pathologies.
In addition to the wind, the drone also suffered interference from wifi networks around it, as the smartphone's wifi signal controls it through a private communication network. The drone model does not have internal storage for archiving videos and photos transmitted in real-time to the smartphone via its wifi network. Thus, interference in the wifi network can compromise the data collection from height, speed, and location coordinates of the drone, the transmission and quality of the photos and videos, and loss of control of the drone causing accidents and loss of equipment.

Data collection was carried out in two moments, in which the communication behavior with the drone varied. In the period of social distance demanded by the pandemic, the UFF buildings were empty and without wireless networks in classrooms, laboratories, and offices. In this context, the drone operated satisfactorily. During the flexibilization of working hours, wireless networks of some laboratories and offices in the UFF buildings were activated. In this case, the drone had problems transmitting videos, photos, and telemetry. When using a wifi signal booster, the drone started to operate satisfactorily. Therefore, we recommend the use of signal reinforcers.

Regarding the algorithms, we can improve the models precision results acting in the dataset. After a first test of the algorithms, we analyzed each image in the dataset and found some lower image quality. The algorithms precision has increased by having the entire dataset with adequate image quality, reinforcing the need to check all the dataset images quality.

A strict criterion must be established to select each image attributed to each pathology class to avoid inconsistencies during the training and testing process. Another important proof is that the inclusion of new images with adequate resolution also increased the accuracy of the algorithms results, which was also to be expected. Bearing in mind that the construction of a dataset is a laborious and costly process, a possible
outcome for this work is determining the number of images that provide the best cost-benefit ratio.

In addition to the network training process to learn the filters used in ML and DL algorithms, there are pre-processing steps and manual adjustment parameters that generally do not perform well in captured images. These parameters are influenced by the test environment’s conditions, where the lighting and shading conditions are potentially unknown or simply uncontrollable, being interesting variables to be explored in future studies.

When testing the use of computational resources of the central processing unit (CPU) and graphics processing unit (GPU), measuring the response time of machine learning without running the risk of “freezing” the system, we concluded that a significant number of data sets require large computational operations in memory. In this context, the GPU is the best option for calculating data efficiently.

We found no difficulties in using it regarding the web interface, mainly because it is intuitive and low complexity.

4. CONCLUSIONS

In this study, we developed a low-cost system for the identification and classification of building facades pathologies formed by a drone for capturing high-resolution images and a web application containing three components: a) a set of machine learning and deep learning algorithms for the pathologies classification; b) a dataset of façade pathologies; c) an interface that allows an environment of interaction and selection of information that contributes to the user’s decision-making process.

Based on the experiments carried out and the implementations developed, it was possible to verify the machine learning efficiency and deep learning techniques. It was also possible to confirm the adequacy of resources and goals for creating a low-cost solution in survey activities and inspection of pathologies on building facades.

It should be noted that although the system developed in this work has been applied to buildings with five floors, it is a scalable solution for buildings up to 30 meters, considering that this is the drone operation limit provided by legislation by ANAC. However, it will be necessary for buildings above 30 meters to use more sophisticated drones, with a higher cost. However, the methodology using AI presented in this article does not depend on this fact. It is possible to implement it for such buildings, including being used for other construction types, such as inspections on bridges, viaducts, towers, and dams. We hope that this work can contribute to the new digital transformation trends in civil construction and make the façade inspection process more efficient and routine.

REFERENCES


[31] Keras - The Python deep learning API. Available online: https://keras.io (accessed on 09 June 2020).


[46] Ubuntu 18.04.5 LTS (Bionic Beaver) Linux OS. Available online: https://releases.ubuntu.com/18.04/ (accessed on 02 July 2020).