POLITECNICO DI TORINO
Repository ISTITUZIONALE

DEcay Classification using Artificial Intelligence

Original
DEcay Classification using Artificial Intelligence / Giovannini, E.C., Tomalini, A., Pristeri, E., Bergamasco, L., Lo Turco, M.. - In: INTERNATIONAL ARCHIVES OF THE PHOTOGRAMMETRY, REMOTE SENSING AND SPATIAL INFORMATION SCIENCES. - ISSN 2194-9034. - ELETTRONICO. - XLIII-B2-2021:(2021), pp. 847-854. [10.5194/isprs-archives-XLIII-B2-2021-847-2021]

Availability:
This version is available at: 11583/2910572 since: 2021-07-01T16:05:13Z

Publisher:
Copernicus Publ.

Published
DOI:10.5194/isprs-archives-XLIII-B2-2021-847-2021

Terms of use:

This article is made available under terms and conditions as specified in the corresponding bibliographic description in the repository

Publisher copyright

(Article begins on next page)

12 October 2026

DECAY CLASSIFICATION USING ARTIFICIAL INTELLIGENCE

E. C. Giovannini [1], A. Tomalini [1], E. Pristeri [2], L. Bergamasco [2], M. Lo Turco [1]*

[1] DAD, Dept. of Architecture and Design, Politecnico di Torino, Turin, Italy
(massimiliano.loturco, andrea.tomalini, elisabettacaterina.giovannini)@polito.it
[2] LINKS Foundation, Leading Innovation & Knowledge for Society, Turin, Italy
(letizia.bergamasco, edoardo.pristeri)@linksfoundation.com

Commission II, WG II/8

KEY WORDS: Machine Learning, decay map, H-BIM, data analysis, pattern recognition

ABSTRACT:

The paper presents DECAI - DEcay Classification using Artificial Intelligence, a novel study using machine learning algorithms to identify materials, degradations or surface gaps of an architectural artefact in a semi-automatic way. A customised software has been developed to allow the operator to choose which categories of materials to classify, and selecting sample data from an orthophoto of the artefact to train the machine learning algorithms. Thanks to Visual Programming Language algorithms, the classification results are directly imported into the H-BIM environment and used to enrich the H-BIM model of the artefact. To date, the developed tool is dedicated to research use only; future developments will improve the graphical interface to make this tool accessible to a wider public.

1. INTRODUCTION

The paper proposes a novel study that involves the use of digital tools and applications for the management and analysis of decays in historical buildings. Recently, many studies and methodologies have tried to digitise processes of decay annotations, but to date, a methodological approach is missing.
Diverse technologies and tools are available, but they are generally employed by users with different backgrounds and purposes. The presented research aimed at creating a novel collaborative process between ICT scientists and the AEC sector, their common methodologies and interoperable tools.
Building Information Modelling (BIM) and its application at architectural heritage (H-BIM) methodologies are increasingly being investigated in the management of architecture (Logothetis et al., 2015; Murphy et al., 2013). The use of these methodologies has opened research to investigate novel approaches considering a multidisciplinary perspective in the collection, management, and analysis of different types of data (Tsilimantou et al., 2020).
Computer science and technological advancement enabled the creation of a large amount of data in the architectural heritage field. These data acquired by diverse instruments are no longer collected for the unique purpose of measuring or verifying metric accuracy but can be analysed with specific purposes. The efforts made in recent years in this direction have led to the development of many techniques useful to generate new forms of knowledge analysing data and information through pattern recognition algorithms (Dulli et al., 2009).
Despite the large amount of data acquired and available, the architectural field still has to become confident with novel ICT techniques, such as machine and deep learning, encouraging and promoting the development of good practices for recording, documentation and information management of architectural heritage.
In the field, finding diverse methodologies to map the decay of walls is an ongoing area of research, where diverse approaches are available. To date, the operations of cataloguing and assessing degradation and materials are carried out manually by specialised operators. The images collected of the artefact are studied individually: efflorescence, exfoliation, biological patina, cracks, black crusts, detachments, and stains due to rising damp; they are all identifiable "on sight". This information can be also digitised using BIM software (Bruno and Roncella, 2019; Malinverni et al., 2019) where the digitisation process consists in the creation of objects that are computerised in the H-BIM environment and that can be created by parametric modelling (Brumana et al., 2017; Chiabrando et al., 2017) or using algorithms (Giovannini and Tomalini, 2021; Lo Turco et al., 2021).
Some studies focus on the development of workflows and methodologies that can create H-BIM models from point clouds (Fryskowska and Stachelek, 2018) or by using point clouds as reference data (Grilli and Remondino, 2019; Lo Turco and Santagati, 2016; Quattrini et al., 2015).
Analysis and use of this data are usually focused on the 3D modelling phase instead of on the recognition of decay elements.
The paper presents preliminary results of the ongoing project DECAI - DEcay Classification using Artificial Intelligence, a collaboration between Links Foundation and the Department of Architecture and Design of Politecnico di Torino. DECAI proposes a machine learning-based pipeline that uses an orthophotos “intelligent” analysis to classify and recognise materials, degradations, surface gaps of elevations in a semi-automatic way. The classification results can be imported into the H-BIM environment by using Visual Programming Language algorithms.

2. THE CASE STUDY

2.1 The Castle of Bonavalle

The Castle of Bonavalle is located in the Italian Province of Cuneo and it is nowadays in a state of abandonment since the early 20th century. The fief of Bonavalle was claimed in the 19th

* E. C. Giovannini is author of par. 1 and 2, E. Pristeri is author of par. 3, L. Bergamasco is author of par. 4.1 and A. Tomalini is author of par. 4.2. Lo Turco is author of par. 5

century, by the Municipality of Racconigi, for historical reasons, but there is no concrete evidence of an actual annexation, even if the reason for its abandonment is certainly related to its property transfer to the public body.
The settlement, with its original defensive function, has towers on the main south front and a perimeter moat that is no longer recognizable. Following the loss of the function of fortification, the castle was used as a "noble" residence outside the city. The presence of diverse transformation of the building reflects the succession of different lord owners during diverse past years. The transformations have interested both the change of use (first fortress, immediately after Consignoria and finally farm) and sometimes the structural modification of the building that nowadays appears to be a stratified set of interventions. However, the more drastic and invasive interventions (for example the creation of chimneys), over the years have caused the formation of cracks and various instabilities that, not having been monitored, have become widespread and now involve serious consequences from a structural point of view (Comoli Mandracci, 1988; Cravero, 2010).
The ruinous state of the castle, however, is not only due to the repeated change of ownership, but also to the subsequent abandonment of the Seigniory, which occurred after the death of the last owner Giuseppe Augusto Levis at the beginning of the 20th century, who, before his death, left one-third of his property to the Municipality of Chiomonte, and the remaining two thirds to the Municipality of Racconigi. The lack of care on the part of the Municipality led the building is in a state of abandonment, and this situation could even worsen leading to the collapse of the entire property. Intending to avoid this, around the year 2000, the Municipality of Racconigi opened a public auction to assign the management of the ruin to private individuals.

Figure 1. Drone Phantom 4 Pro picture of the ruins of Castle.

The ruin consists of a square body, with two angular towers facing south and two small towers surmounted by domes facing north. The north, east and west fronts rose to three floors above ground; the south one has a further two-storey elevation in the central part. The load-bearing structure of the Castle is composed of solid brick walls. Their development in height varies from floor to floor and it starts from a useful section of 1.20 m up to 0.50 m approximately. In addition to the three floors above ground, the building also has a basement consisting of rooms with vaulted brick. The main element of coverage, which ran on the four sleeves, is no longer present.

2.2 The digital acquisition

The digital acquisition and survey campaign were performed using tools and methodologies to obtain an accuracy that allowed a precise analysis of both the degradation of walls and the surface characteristics. Furthermore, the dataset allowed a comprehensive understanding of the artefact.
During the survey campaign, traditional topographic, image-based and range-based technologies have been employed. The performed survey operations have respected the precise and defined order going "from general to particular".
For a correct georeferencing of the acquired and produced data, a topographic network has been created using GPS/GNSS technologies; every subsequent elaboration has been referred to this topographic network.
The architectural components have been acquired using terrestrial and aerial tools. 135 pictures were made using a Canon EOS 5DS R with a focal length of 25 mm and a constant focal ratio of f/11. The camera acquisition had a trajectory circle clockwise starting from the facade facing south. Obtained RAW images were post-processed to obtain .jpeg format with max. resolution of 8688x579 pixels.
The aerial acquisition was made using the DJI Phantom 4 Pro drone. Its camera, the FC6310 has a CMOS 1" sensor (13,2x8,8 mm) and allows it to obtain 20 MP pictures (Figure 1). The drone's path showed in Figure 2 was made manually through a radio control connected to the smartphone, which normally reaches a maximum transmission distance of 3.5 km. 283 images were acquired in .jpg format of 5472x3648 pixels and the height of the flight was defined using a pre-calculated Ground Sample Distance (GSD) (Cusniriuc and Martillo Flores, 2019).

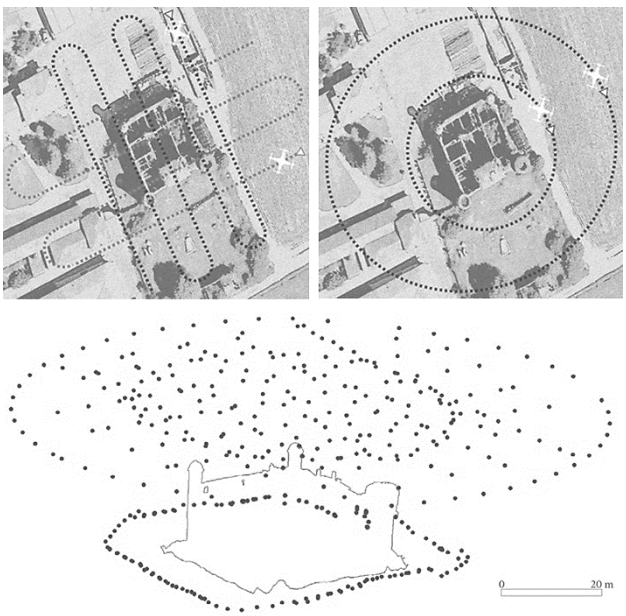

Figure 2. From the top: drones trajectories; position of cameras used for architectural component's acquisition.

2.3 H-BIM digitisation process

Autodesk Revit 2020 software was used to create the H-BIM model. The point clouds obtained by the post-processing of collected pictures were imported in Revit. For more suitable management, the clouds were translated from the original coordinates (x, y, z) to coordinates close to the origin of the axes (x', y', z') in the Revit coordinates system.
The perimeter walls were modelled using conceptual masses obtained from the extrusion of sections generated by the intersection between numerous section planes and the point cloud. The distance between the section planes ranges from 1m to 1,5m, then refinements section planes have sometimes 0,5 m

of distance. The section planes, therefore, allow obtaining a more accurate H-BIM model as shown in Figure 3.
Once the perimeter walls were defined, the modelling of all architectural elements, functional and decorative, were developed using ad-hoc parametric families (Figure 4).

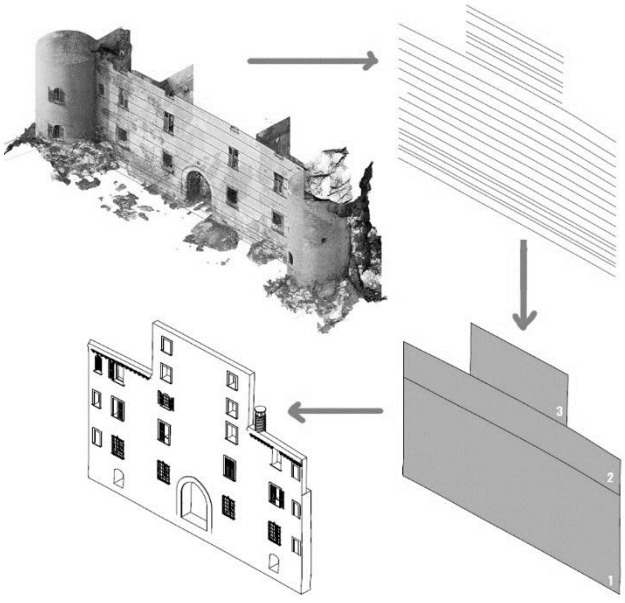

Figure 3. Digitisation workflow to obtain an H-BIM model starting from a point cloud.

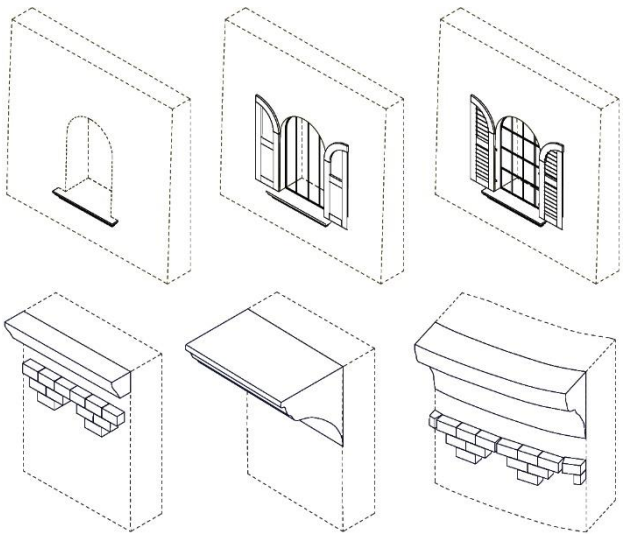

Figure 4. From the top: parametric family for a window, parametric family for wall decorative elements.

3. A DECAY CLASSIFICATION METHODOLOGY

3.1 Digital tools and environments

3.1.1 Machine Learning (ML) algorithms: Image classification and pattern recognition represent a very explored research field. Numerous algorithms are nowadays available, along with their implementations, to perform these tasks. ML algorithms allow creating models that can be trained on a dataset and learn how to classify images from the data itself. These models can be conceptualised as the collection of the memories the algorithms recall to perform the classification tasks. It is also worth noting that one of the differences between classic ML and Deep learning (DL) algorithms lies in how much data is needed to train the models: usually, solutions using DL algorithms require considerably more data. In recent years, these algorithms have become easily accessible and performing enough to allow developers to integrate these solutions into the most different fields, and architecture is no exception. In this project, we use the ML algorithms implementation available in the scikit-learn open-source Python library (Pedregosa et al., 2011).
Since each algorithm brings advantages and disadvantages, more than one algorithm should be tested when approaching a new classification problem. Thus, when facing the challenge described in this paper, our approach has been to consider each image as a different classification problem, with a set of algorithms to try and models to be created explicitly for that image. The advantage of this approach is that only a small amount of training data is needed to perform the classification, with respect to solutions using DL algorithms.

3.1.2 Visual Programming Language (VPL): The use of VPL software became quite common in the architectural field because it allows simplifying the complexity of the parametric design. VPL is a programming language that uses graphical elements and figures to develop a program instead of coding. In DECAI, it is used to integrate the ML classification results in the H-BIM environment. The ML part, instead, is developed in a traditional programming environment since it allows shorter and more flexible workflows; moreover, the available libraries are much more performing and developed than those ready for use in the VPL environment. The VPL application chosen for this research is Grasshopper. It was chosen for its flexibility and for the numerous plugins already developed that allow connecting different applications with the BIM environment.

3.1.3 H-BIM environment: The ML software produces a classified orthophoto, where each class is represented by a different colour and allows exporting the results in a VPL compatible file. Once imported within the VPL environment, the results are geo-referenced and projected onto the H-BIM model of the architectural artefact. Using VPL as a connector between traditional modellers and the H-BIM environment allows to easily import and manage data that have been analysed and processed in the external environment. The libraries for this type of operation in the VPL environment are already well developed.

3.2 DECAI overall workflow

Figure 5 presents DECAI general workflow, describing the process that leads from the survey phase to the creation of the final H-BIM model, and the corresponding data flow. Once all the images from the survey (.JPG files) have been collected, they are processed by the Agisoft Metashape software to create a point cloud. On the one side, this point cloud data is used to manually build the H-BIM model in Revit. On the other side, the same point cloud allows the generation of orthophotos (.TIF file) that will be the input of DORA, the ML classification software. DORA employs ML algorithms to classify the different parts of the orthophoto as belonging to one of the classes (materials, finish, gaps, degradations) that the user wants to identify. These results are exported in a VPL compatible format (text files) so that they can be easily imported in Grasshopper. Thanks to the developed VPL algorithms and the integration of Grasshopper with Revit, the information about the identified types of decay can be automatically projected onto the manually built H-BIM model, thus building the final H-BIM model, enriched with the results coming from the automatic ML classification of areas presenting signs of decay.

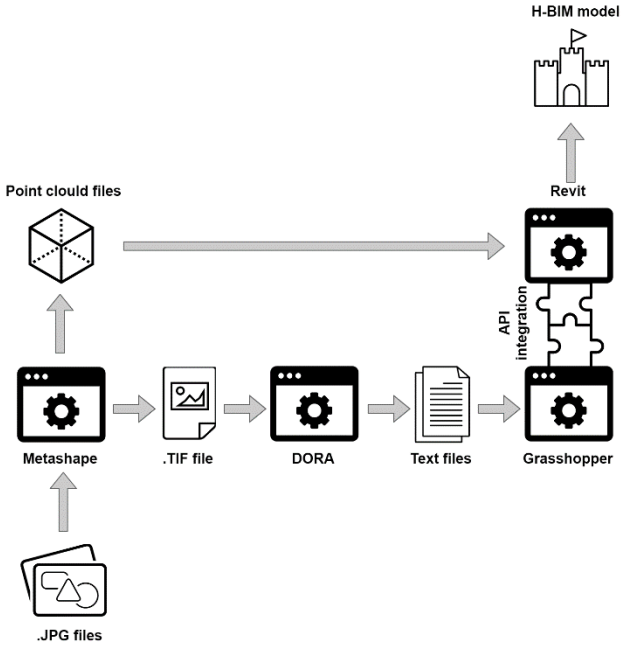

Figure 5. Scheme of DECAI general workflow, highlighting the data flows from survey to H-BIM environment.

4. THE APPLICATIVE SCENARIO

4.1 The ML algorithm workflow

4.1.1 Preprocessing: Preprocessing techniques may be helpful to facilitate image classification, especially in cases where there is noise or bad illumination conditions. The employed preprocessing phase currently includes contrast enhancement through histogram equalization.

4.1.2 ML models training: In the field of supervised learning, to learn how to distinguish data belonging to different classes, ML models have to be trained with some labelled data, where the class of each data point is known. In this work, we leverage the expertise of a cultural heritage operator to select these training data from the orthophoto exported from the Agisoft Metashape software. Guided by the user interface, the operator has to select rectangular areas on the displayed orthophoto (a couple of rectangular areas for each class that they want to identify). A moving window system is implemented to divide the training areas into smaller parts, which are more suitable for feature extraction. Indeed, as happens in a typical ML pipeline, the so defined training data are used to extract features that will be given as input to the ML models during the training phase. Once trained, the models are ready to receive features coming from data that they have never seen before and classify them as belonging to one of the considered classes.

4.1.3 Feature extraction: Both the training data and the new unseen data undergo feature extraction before being given as input to the ML models. In ML, features represent some characteristics of the data, which allow the models to discriminate among different classes. In particular, in this work, we have employed features quantifying shape (Hu Moments), texture (Haralick texture) and colour (Histograms, saturation, hue, brightness). To be significant, the features are not calculated from the whole areas selected by the user (for the training data) or from the whole image (for unseen data); instead, they are calculated on smaller areas, defined according to the implemented moving window mechanism.

4.1.4 Moving window mechanism: To define small portions of the image that are suitable for feature extraction, we have implemented a moving window mechanism that works as follows. In the beginning, we define the size of the window that will delimit the image area where to compute the features from; this is done according to the user settings. In the case of training data, for each area selected by the user (see example in Figure 9), we shift the window along that area, and at every step, we calculate the features for the part of the area marked by the position of the window. This procedure is illustrated in Figure 6. In the case of the classification of the whole image, when the trained ML models have to classify unseen data, we apply again the moving window system in an analogous way, this time shifting the defined window along with the whole image. For every position of the window, the set of features is computed and given as input to the trained ML models, which have to classify the considered area of the image.

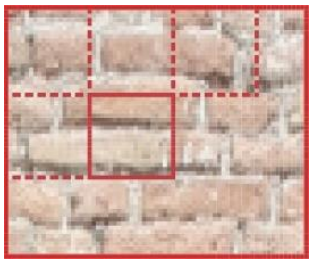

Figure 6. Illustration of the moving window mechanism to define the areas used for feature extraction from the training areas selected by the user.

4.1.5 Classification: As written before, the classification step requires that the whole image is divided into smaller areas, according to the window mechanism, and each of them is assigned one of the classes (selected by the user) by the ML models. Once all the smaller areas have been processed by the ML models, the classification process is complete. It is important to note that the same set of features is calculated only once, and given as input to all the models. Each model acts independently from the others, this means that the user can have as output one orthophoto for each selected model and pick the best looking one. To build the employed models, we have tested (and included in the list of available choices for the user) the following algorithms: Random Forest, Support Vector Machine, Logistic Regression, Extremely Randomized Trees, AdaBoost Decision Trees, Bagging Decision Trees, Linear Discriminant Analysis and K-Nearest Neighbors. As a remark, in this work, we do not utilize pre-trained models. For each experiment, we generate new models according to the user's choices and train them on the training data selected by the user. Since textures may vary a lot in different use cases, this solution provides more flexibility, because for each different case the operator can choose any set of classes to be identified, and optimize the models based on the available data.

4.1.6 Postprocessing: Postprocessing techniques may be used to improve the quality of the result, i.e. the classified orthophoto. Currently, the postprocessing phase includes image filtering with the implemented median filter. This filter is characterized by a parameter named window size, which can be set to the desired value by the user. Higher values of this parameter usually bring a higher "blurring" effect.

4.1.7 Orthophoto classification workflow: A scheme of the workflow for the orthophoto classification with ML algorithms performed by DORA is illustrated in Figure 7.

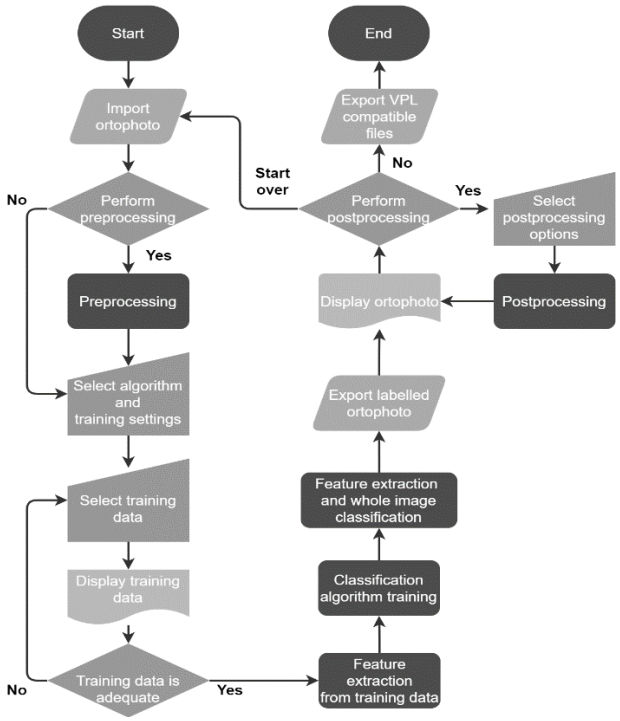

Figure 7. Scheme of DORA classification workflow.

The classification workflow includes the following steps:
1) The user selects the desired orthophoto exported from the Agisoft Metashape software in the TIFF format and then proceeds to import it in the developed classification software.
2) The user can now decide whether to perform preprocessing on the orthophoto or not.
3) The user has then to select the ML models that they want to use, among the ones available in the provided list. Then the user may also choose some parameters relative to the classification phase:
(a) the size in pixels of the window used in the moving window mechanism for feature extraction and classification;
(b) the different classes that they want to distinguish in the image, e.g. plaster, bricks, frescos, gaps, defects…;
(c) the colours that they want to assign to each class, among the ones available in the provided list.
4) The user, following the instructions provided by the user interface, should select some training data for each class. This is done by selecting rectangular areas on the displayed orthophoto. Each rectangular area is then divided into smaller parts, according to the moving window mechanism.
5) After the selection, it is possible to visualize again all the training data, to check that they are consistent with the instructions and that there are no accidental mistakes.
6) The user is finally asked for a confirmation. If the user deems the training data as adequate, the software proceeds with the next steps, otherwise the user is brought back to the training data selection (step 5).
7) The software proceeds to extract the features from the training data, computing the set of features explained in "Feature extraction" for each window of training data.
8) The models are now trained on the previously extracted features.
9) The next step is the classification of the whole image. As explained, here it is necessary as well to extract the features that will be used to feed the trained ML models, and this is done according to the moving window mechanism. Therefore, for every position of the window, the set of features is computed and used by each model to classify the considered part of the image as belonging to one of the classes and to assign it the corresponding colour. At the end of this step, we get a classified orthophoto for each ML model.
10) Afterwards, the user can visualize the classified orthophoto (for each selected ML model) and decide if they are satisfied with it or if they want to perform post-processing. The user can try postprocessing many times with different settings, or even decide to start over and try a different approach in some other steps.
11) If the post-processing option is selected, the user has to configure some settings, e.g. a stronger or weaker effect of the median filter. The result of this step is again an orthophoto for each ML model.
12) Once they are satisfied with the resulting orthophoto, the user can export the results in a VPL compatible format (text files).

4.1.8 ML results and discussion: The image classification results for the Castle of Bonavalle are reported in Figures 8-13. Figure 9 illustrates which training areas have been selected by the user for each class, starting from the orthophoto in Figure 8. The chosen classes were plaster, fresco, brick, brick with lacks, stone, vegetation and others. Given the good illumination conditions and general aspect of the image, no preprocessing has been considered necessary in this particular case. The Random Forest classifier was the model that provided the classified orthophoto (Figure 10) which was the most similar to the ground truth in Figure 12, i.e. an orthophoto that has been manually labelled by an operator for results comparison and evaluation. The choice to perform a post-processing phase has led to the image in Figure 11, where a lot of noise has been filtered out and the areas representing the different classes are more homogeneous.

Figure 8. Original orthophoto of the Castle of Bonavalle.

Figure 9. Training areas selected by the user through the user interface. Legend: **magenta**=plaster, **navy**=fresco, **red**=brick, **brown**=brick with lacks, **blue**=stone, **lime**=vegetation, **cyan**=other.

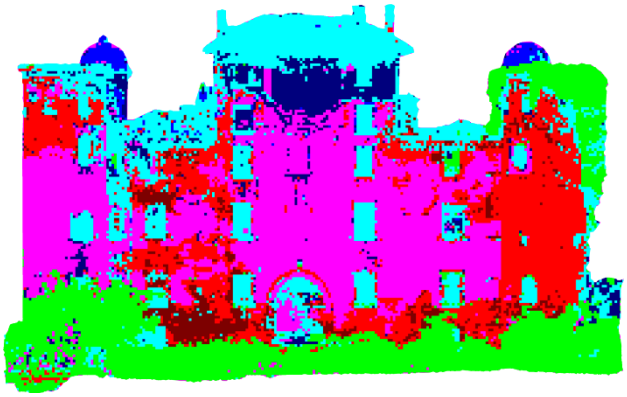

Figure 10. Orthophoto resulting from the image classification with ML, using Random Forest classifier.

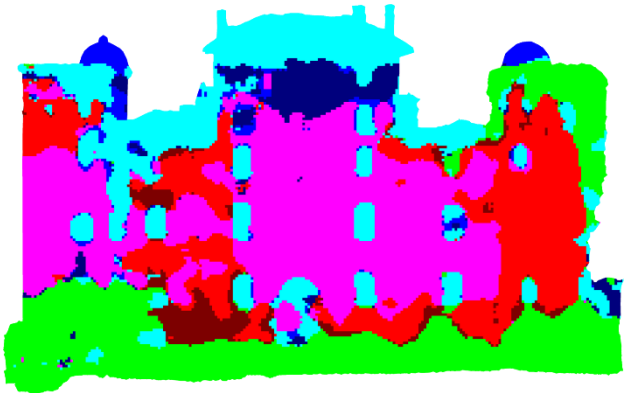

Figure 11. Classified orthophoto after postprocessing.

Figure 12. Orthophoto manually labelled by an operator, serving as ground truth to evaluate the results.

Since the user is free to choose the training areas that they consider as the most representative of each class, the final results will highly depend on this choice. However, the graphical interface provides guidelines that can help the user in this phase, e.g. suggesting how many areas to select, how big they should be, or which details to avoid, to facilitate the ML models' learning and possibly optimise the classification results.
At first glance, the result in Figure 11 can be already considered a good approximation of the manually labelled image in Figure 12. Having such an approximation allows to easily calculate the size of the surface corresponding to each class, and this is crucial especially in the case of restoration work. In that situation, an estimate of the needed amount of materials is calculated according to the size of the surface to be renovated. According to the professional sector, in Italy, it is sufficient to obtain an approximation of this area to ensure proper management of the materials needed to complete the restoration work.
Besides the visual comparison, we also defined some metrics to quantitatively evaluate the performance of the proposed system. Considering the manually labelled orthophoto (e.g. the one in Figure 12) as ground truth, and the resulting orthophoto (e.g. the one in Figure 11) as the prediction, we compute a confusion matrix. Each column of the matrix represents a class of the ground truth image, while the rows represent the classes in the predicted image; each element of the matrix indicates how many pixels of that column's (actual) class has been classified as that row's (predicted) class. The values are normalised by columns so that each column's values sum up to one.

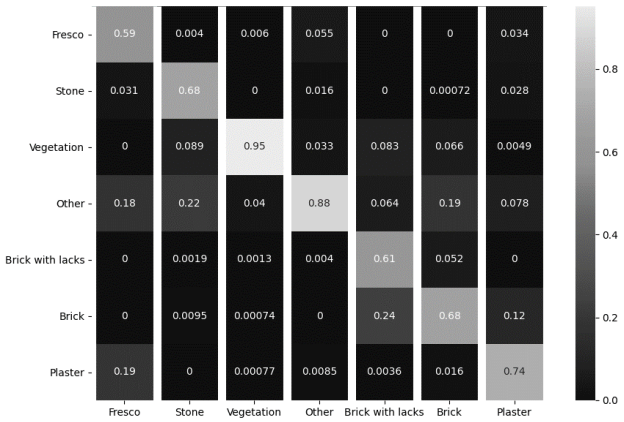

Figure 13. Confusion matrix related to the results of the classification results of figures 8-11.

In Figure 13 we report the confusion matrix obtained from the results shown in Figures 8-12. We can see that the class with the highest percentage of correctly classified pixels is vegetation, where 94% of the ground truth pixels have been predicted as vegetation. It is important to observe that also for the other classes the majority of pixels have been correctly classified. We can notice that classes that are more similar among them are sometimes confused. For example, 59% of fresco pixels are classified as such, but 19% are classified as plaster; regarding brick with lacks, 61% pixels are correctly predicted, while 24% are confused with simple bricks; and so on. This happens because many areas belonging to those pairs of confused classes look so similar that they may be difficult to distinguish even for an operator.

4.2 VPL and H-BIM approach

As mentioned above, the output of the ML processes includes text files in a VPL compatible format. The .txt extension can be, in fact, imported without any formatting request by grasshopper plug-in. In particular, there are two text files for each selected class (material or degradation) that the user has chosen to identify in the orthophoto.
One file contains a list of all the points coordinates that are useful for identifying the perimeter of those portions of the elevation surface characterized by homogeneous finish characteristics.
The other file collects a list of numbers corresponding to the number of points that generates each surface perimeter: a pattern with which the points of the first list are divided. The purpose of the developed VPL algorithm is to use these textual files to automatically generate a set of surfaces that will enrich the previously constructed H-BIM model with further details.
The proposed VPL workflow can be summarized in the following

six phases and was developed using grasshopper and Revit environments. The whole procedure can be repeated for each necessary class identified into an orthophoto.

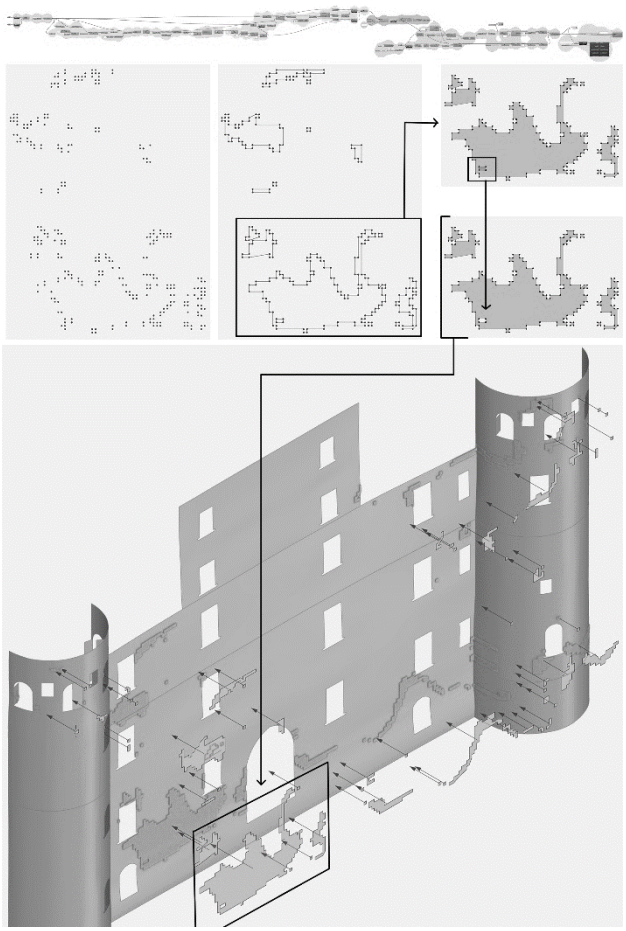

Figure 14. Graphic display of the phases 1 to 5, performed in the VPL environment.

1) The first phase consists of the importing of the two text files corresponding to the identified classes into the VPL environment. The list including all perimeter points is divided into a list of lists (data structure used within Grasshopper to manage data). Using the components "Polyline" and "Boundary", all the surfaces belonging to the considered class are automatically identified.
2) After the check that identifies possible overlaps between the generated surfaces, if the function finds that within a larger area there is a smaller area that does not belong to that category, the overlap is identified. The surfaces are then arranged on the Z-axis. Their orders are constrained to have the smallest one at the top and the largest one at the bottom. The centre point of each surface is projected downwards and if there is an intersection the upper surface is subtracted from the below one.
3) During the phases described above, diverse functions are performed to obtain all the surfaces resulting from subtraction operation and characterized by the same finish characteristics. As reported in Figure 14 the resulting dataset is composed of diverse trimmed surfaces.
4) The coordinate system of the preliminary phases is defined by the ML system and corresponds to the pixel resolution of the selected orthophoto. The fourth phase, then, has to perform a change to the previous coordinate system to make it compatible with the BIM environment. But first, the alignment with the coordinate system of the point cloud referenced file is necessary. Surfaces are scaled and roto-translated to the correct position using data from the *.tfw file (a file generated by the Agisoft Metashape software that contains a 2x2 matrix for roto/scaling and two values for a translation vector).
5) The fifth phase consists in visualizing the H-BIM model within the VPL environment and projecting generated surfaces on the model as shown in Figure 15.

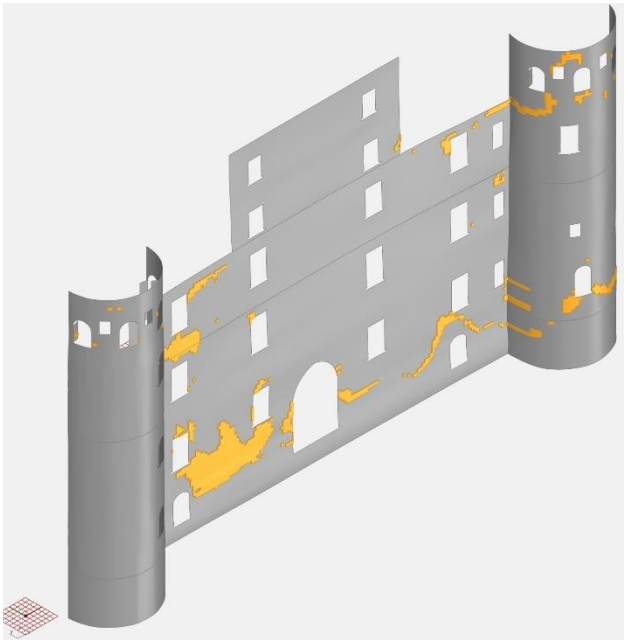

Figure 15. Grasshopper environment: surfaces projected on imported model using VPL.

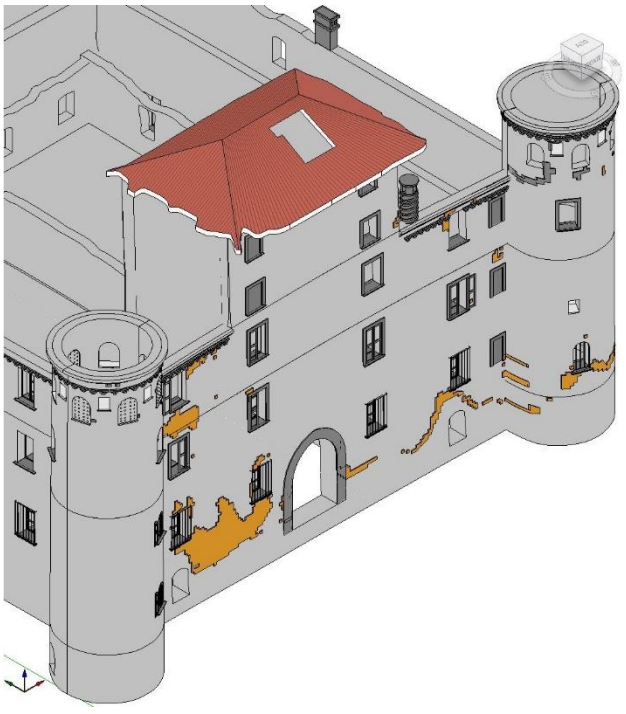

Figure 16. Revit environment: surfaces imported as a wall building component in the BIM environment.

6) Finally, the algorithm imports the surfaces into Revit software (Figure 16). These elements are displayed and imported as walls with a thickness of 1mm. The creation of these families allows the assignment of a material's parameter that can be counted and

managed through an abacus. The import process was tested using two different Grasshopper add-ons, Grevit and Rhino.Inside®.Revit. The most stable workflow was obtained with Rhino.Inside®.Revit, where the "Add Geometry DirectShape" component allows to import any geometry inside Revit and assign it a specific Category, Name and Material.

5. CONCLUSIONS

In this paper, we have presented the DECAI project, which combines decay recognition with machine learning algorithms, VPL programming and H-BIM modelling. It must be emphasised that the proposed pipeline is still under development, but the results are promising. Processing times are still short and accuracy results are satisfying. However, there may be improvements in both the ML procedures and the algorithm construction in the VPL environment.

In the ML environment, the implementation of more specific preprocessing techniques could further improve the aspect and noise of the orthophotos, leading to higher classification accuracy. Moreover, the integration of a previously built texture dataset with the training data selected by the operator could be explored, both to minimize possible mistakes on the operator's side and to enhance the classification performance. Another path that could be explored is the implementation of contouring algorithms on the segmented orthophotos, to obtain a more detailed drawing that supports also a larger scale of representation.

In the VPL environment, the management of surface overlaps must certainly be improved, currently being the bottleneck of this phase of the general workflow. It is essential to be able to link more information to the geometries imported into the BIM environment. Furthermore, it is necessary to develop a labelling system linked to this information, which can streamline the procedure of drawing up the tables.

The flexibility of the proposed workflow needs to be verified as well. To date, the output of the ML procedures has been formatted in the most general way, with the aim that systems other than Grasshopper can also integrate and process this kind of output. Future development will be to transpose the same algorithm also within Dynamo visual programming platform, to test the actual flexibility of the VPL workflow and the opportunities connected to this different tool.

ACKNOWLEDGEMENTS

The digital acquisition's dataset and the H-BIM model used for the DECAI research are parts of outputs developed by Cusniriuc, Victor Theodor and Martillo Flores, Jorge Washington for their thesis (MSc degree program in Architecture Construction and City), at Politecnico di Torino entitled "Bonavalle. 3D metric survey, HBIM and VPL for the protection of architectural heritage." The thesis supervisors were Professors M. Lo Turco, F. Chiabrando and M. Naretto.

REFERENCES

Brumana, R., Dellatorre, S., Oreni, D., Previtali, M., Cantini, L., Barazzetti, L., Franchi, A., Banfi, F., 2017. HBIM challenge among the paradigm of complexity, tools and preservation: The basilica Di collemaggio 8 years after the earthquake (L'aquila). *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci. - ISPRS Arch.* 42, 97–104.

Bruno, N., Roncella, R., 2019. HBIM for Conservation: A New Proposal for Information Modeling. *Remote Sens.* 11, 1751.

Chiabrando, F., Lo Turco, M., Rinaudo, F., 2017. Modeling the decay in an hbim starting from 3d point clouds. A followed approach for cultural heritage knowledge. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* 42, 605–612.

Comoli Mandracci, V., 1988. *PIEMONTE*. Laterza, Roma-Bari.

Cravero, P., 2010. Il Castello di Bonavalle: problemi di restauro e conservazione. (Dissertation) Politecnico di Torino, Turin.

Cusniriuc, V.T., Martillo Flores, J.W., 2019. Bonavalle. 3D metric survey, HBIM and VPL for the protection of architectural heritage. (Dissertation) Politecnico di Torino, Turin.

Dulli, S., Furini, S., Peron, E., 2009. *Data mining*, UNITEXT. Springer Milan, Milano.

Fryskowska, A., Stachelek, J., 2018. A no-reference method of geometric content quality analysis of 3D models generated from laser scanning point clouds for hBIM. *J. Cult. Herit.* 34, 95–108.

Giovannini, E.C., Tomalini, A., 2021. An algorithmic information model (AIM) for the map of decay: the Church of San Giuliano. *DN* 7, 130–140.

Grilli, E., Remondino, F., 2019. Classification of 3D digital heritage. *Remote Sens.* 11, 38121.

Lo Turco, M., Calvano, M., Giovannini, E.C., Tomalini, A., 2021. AIM! Algorithmic Information Modeling: New Strategies for a Fully Integrated Approach in the Field of Cultural Heritage, in: Springer Tracts in Civil Engineering. Springer, pp. 143–156.

Lo Turco, M., Santagati, C., 2016. From SfM to semantic-aware BIM objects of architectural elements, in: Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). Springer Verlag, pp. 600–612.

Logothetis, S., Delinasiou, A., Stylianidis, E., 2015. Building information modelling for cultural heritage: A review. *ISPRS Ann. Photogramm. Remote Sens. Spat. Inf. Sci.* 2, 177–183.

Malinverni, E.S., Mariano, F., Di Stefano, F., Petetta, L., Onori, F., 2019. Modelling in HBIM to document materials decay by a thematic mapping to manage the cultural heritage: The case of "chiesa della pietà" in fermo. *ISPRS Ann. Photogramm. Remote Sens. Spat. Inf. Sci.* 42, 777–784.

Murphy, M., McGovern, E., Pavia, S., 2013. Historic Building Information Modelling - Adding intelligence to laser and image based surveys of European classical architecture. *ISPRS J. Photogramm. Remote Sens.* 76, 89–102.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E., 2011. Scikit-learn: Machine Learning in Python. *J. Mach. Learn. Res.* 12, 2825–2830.

Quattrini, R., Malinverni, E.S., Clini, P., Nespeca, R., Orlietti, E., 2015. From TLS to HBIM: high quality semantically-aware 3D modeling of complex architecture. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* XL-5/W4, 367–374.

Tsilimantou, E., Delegou, E.T., Nikitakos, I.A., Ioannidis, C., Moropoulou, A., 2020. GIS and BIM as Integrated Digital Environments for Modeling and Monitoring of Historic Buildings. *Appl. Sci.* 10, 1078.