APPLIED RESEARCH

Computer Vision Based Inspection on Post-Earthquake With UAV Synthetic Dataset

MATEUSZ ŻARSKI[1,2]**, BARTOSZ WÓJCIK**[1,2]**, JAROSŁAW A. MISZCZAK**[2]**, BARTLOMIEJ BLACHOWSKI**[3]**, AND MARIUSZ OSTROWSKI**[3]

[1]Faculty of Civil Engineering, Silesian University of Technology, 44-100 Gliwice, Poland
[2]Institute of Theoretical and Applied Informatics, Polish Academy of Sciences, 44-100 Gliwice, Poland
[3]Institute of Fundamental Technological Research, Polish Academy of Sciences, 02-106 Warsaw, Poland

Corresponding author: Mateusz Żarski (mateusz.zarski@polsl.pl)

ABSTRACT The area affected by the earthquake is vast and often difficult to entirely cover, and the earthquake itself is a sudden event that causes multiple defects simultaneously, that cannot be effectively traced using traditional, manual methods. This article presents an innovative approach to the problem of detecting damage after sudden events by using an interconnected set of deep machine learning models organized in a single pipeline and allowing for easy modification and swapping models seamlessly. Models in the pipeline were trained with a synthetic dataset and were adapted to be further evaluated and used with unmanned aerial vehicles (UAVs) in real-world conditions. Thanks to the methods presented in the article, it is possible to obtain high accuracy in detecting buildings defects, segmenting constructions into their components and estimating their technical condition based on a single drone flight.

INDEX TERMS Structural health monitoring, machine learning, defect detection, synthetic dataset.

I. INTRODUCTION

Earthquakes are sudden and violent disasters that cover huge areas of land in a very short period of time. They have been known to mankind since ancient times and invariably pose one of the most serious threats to the lives of people concentrated in large cities. The scale of their destructive power can be seen in the number of nearly two million earthquake victims in the 20th century alone [1], or in the most devastating events, which could claim up to nearly a million lives [2]. At the same time, the map of seismically active areas largely overlaps with densely populated areas, particularly in North America, Europe and Asia [3], which focuses researchers on this type of hazard and methods of its mitigation.

Studies conducted to date have assessed the effects of earthquakes both in terms of the impact on housing and infrastructure, and the performance of public services in repairing

The associate editor coordinating the review of this manuscript and approving it for publication was Filbert Juwono.

damage or improving traffic flow in the affected area [4], [5]. These works have led to concepts of cities in which such events will no longer have a critical impact on the lives of residents, but with the cost of monitoring the condition of structures even after seemingly harmless, small earthquakes to take corrective action immediately after damage occurs [6]. This, however, requires the use of modern methods of construction monitoring to reduce the labor intensity of the entire process, without which the end goal is impossible to achieve.

In this paper, we present our step towards building autonomous systems that can bring this goal closer. Using a synthetic dataset containing models of earthquake-damaged buildings observed from unmanned aerial vehicle (UAV) like perspective, we created a robust, yet easily modifiable pipeline featuring multiple machine learning models that can be applied in real-life scenarios. The models we have trained allow us to detect close objects, segment them into their components, and finally detect their defects and evaluate their condition. In our work, we also described the specifics of working with a synthetic dataset and the

possibilities for extending our solution that could improve its accuracy.

The rest of this paper is organized as follows. In Section II we present works related to the main topics of the article, including earthquake-induced structural defects, defect detection and the use of UAVs along with synthetic dataset. In Section III we describe our approach starting with dataset description. Then we cover in detail each task of our approach together with our solution and develop our final pipeline. Section IV provides discussion of our solution. in which we suggest ways to improve it. Lastly, in Section V we summarize the progress achieved in the work.

II. RELATED WORKS

Although the topic is extremely broad, four primary themes present in the related literature and research papers can be isolated from it. These topics include earthquake-related damage to structures, detection and management of this damage, the use of UAVs for this purpose, and the use of synthetic datasets for machine learning algorithms.

On the subject of damage to buildings caused by earthquakes, many scientific works focus on the analysis of specific cases of disasters [7], [8], or individual constructions [9] affected by earthquake. Related to them are works on estimating possible future damage in a given area [10] and building general models of damage caused by earthquakes [11], [12] or management strategies. There are also works describing the assessment of the accuracy of currently used methods of seismic measurements in relation to the damage recorded on buildings [13], or novel systems of seismic data collection [14].

The field of detection and management of identified faults is also rich in research. In the detection of defects, methods employing, among others, dynamic response of the structure [15] or laser scanning are used [16], but for a long time there has also been a significant increase in the number of works devoted to the use of computer vision, also regarding earthquakes [17]. In this sub-field, classical methods of computer vision [18] are currently being replaced by methods that derive from machine learning, using convolutional [19] and fully convolutional [20] neural networks, LSTM [21] networks or other techniques combining [22] or improving [23] upon these methods. However, it should be noted that only a few systems have been dedicated to detecting more than a single type of defect. Similarly, the field of identified damage management is still evolving, using, among other things, BIM models [24] and mobile applications [25].

The use of UAVs is extremely closely related to the detection of damage to buildings, as they have been used for this purpose for a long time, not only in the form of flying vehicles, but also self-propelled rovers [26]. Flying vehicles were used to detect damage on various surfaces, such as pavements [27], railroads [28] and public infrastructure facilities [29], also with additional sensors [30].

The last field – the use of synthetic data sets in training machine learning algorithms also has a long history

of research related to it. Synthetically generated data sets do not necessarily have to be images and have been used in many areas, ranging from sociology [31], finance [32], medicine [33], to the issues related to computer vision. Since sets of correctly labeled data are necessary in the training of ML algorithms, and their manual collection is extremely time-consuming, automatic generators of synthetic data were also developed, thus further reducing the laboriousness of building a data sets [34].

III. OUR APPROACH

In our approach, we focused on putting various computer vision and machine learning techniques to the test to find the optimal solution to the problem at hand. We used various models of Convolutional Neural Networks for the task of image recognition and semantic segmentation, and decision trees, random forests and naive Bayes algorithm for the classification task. While developing our solution, we used ready-made state-of-the-art models with transfer learning technique for feature recognition as well as our own models and algorithms, developed exclusively for the task and trained from scratch.

In conclusion, we managed to develop a robust pipeline that can perform the tasks of segmenting construction components, their defects and assessing each of the elements' condition in the single run of the algorithm, while delivering satisfactory accuracy. It has to be noted, however, that the presented solution was validated only with the given, synthetic dataset, and for the practical usage, the models should be fine-tuned also on the real-world data points.

All the scripts needed to replicate the obtained results are provided in our repository [35]. The experiments described in this paper were performed on machine equipped with Intel Core i7 3.80 GHz CPU, Nvidia GeForce RTX 3080Ti GPU and 64 GB of RAM.

A. DATASET DESCRIPTION AND INITIAL MANAGEMENT
1) DATASET DESCRIPTION
The provided dataset [36] consisted of 4808 images of $1920 \times 1080px$ size from artificially generated drone flight in 3D urban surroundings and depicted multi-story apartment buildings with hardly noticeable defects along with similarly looking backgrounds. Out of all images, 3 804 were labeled into tasks of component, defect (cracking, spalling and exposed rebar), and damage state recognition. Additionally, separate labels were provided for depth channel of the images.

However, we found the direct usage of the dataset proved to be difficult, as it came with a set of problems that had to be resolved in the first step of data preparation. The most important problem was the presence of conflicting labels in the task of defect recognition. The total number of images with colliding labels along with label-to-label collision marking is summarized in Tab. 1. This problem prevented the use of a single model for defect recognition, as it would have to yield the probabilities of occurrence of three separate labels.

TABLE 1. Occurrences of label collisions in the dataset.

Collision:	Number of images where collisions occurred:
Cracking - Exposed rebar	559
Cracking - Spalling	3660
Exposed rebar - Spalling	1931

TABLE 2. Data points extracted for defects classes after the dataset modification.

Class	Training set	Testing set	Total damaged components
Cracking	11771	5072	16843
Spalling	15681	5030	20711
Exposed rebar	1394	355	1749

This problem was solved by training three separate models for each of the defect class.

Next, we discovered that the defect labels themselves make up only a small fraction of the whole labeled image, which was most noticeable for the *exposed rebar* class. The pixels marked as exposed rebar occupied a maximum of only 1% of the label image, with occurrences of as few as only one labeled pixel per image (less than one hundredth of a percent of all pixels in the image). Furthermore, it was found out that many of the images do not contain any of the class labels, again, with *exposed rebar* class not present on nearly half of the training images. To mitigate this problem, we introduced a step-by-step approach, in which at first the building in image was split into its components and only then the defects were sought in the area of segmented out components.

The last identified problem concerned the buildings visible in the background. While such image alignment is realistic, it does not coincide with the assigned labels, as they cover only the building in the foreground, and thus would be misleading for the classifier during its training (*eg.* wall in the more distant building would be considered *background* while similarly looking wall in the building nearby would be classified accordingly to its class). Moreover, the provided depth channel cannot be used directly by the models, as this data is not provided for the evaluation dataset (although RGB-D sensors are already in use with UAVs [37]). To avoid this problem along with the need to estimate depth maps with ML algorithms like in already existing works [38], we decided to train a simple segmentation model to differentiate between background and foreground in the initial step of the image analysis.

2) INITIAL DATASET MANAGEMENT
Before developing, training and testing our algorithms, a series of modifications to the dataset had to be done, accordingly to the issues described above. The first one was to split it into training and testing datasets to avoid overfitting of the trained algorithms and allow for reliable metrics checking. We split the dataset randomly in 4:1 ratio, where 20% of the dataset was intended for testing, thus yielding 3 043 datapoints for training and 761 for testing. To preserve the split and be able to repeat it with the same outcome, it was initiated with the known seed for the random algorithm.

After splitting the dataset, we prepared it for the task of detecting objects in the foreground and masking unwanted background ones. To include this step in the single, continuous pipeline of performing all the given tasks in one algorithm

run, we added it as *Task 0*. To prepare the dataset, we provided simple, binary image mask labels for classes *background* and *foreground*, where all the objects from component segmentation task were counted as *foreground* class. This way, the data prepared for classifier could differentiate only between whole, close and distant set of objects instead of each object distance type separately. The sample images from the dataset and their reworked counterparts are depicted in the Fig. 1. Apart from *Task 0*, images reworked for this dataset, albeit with different labels, would be then later used for the rest of the tasks as well.

Next, for the task of detecting defects, we prepared a new dataset that could help us with the problem of small number of labeled pixels for the defects and their complete absence from some of the images. To do so, we used labels for detecting components from *Task 2* to make rectangular crops containing a single element and its immediate surroundings. We repeated the same operation on label images, thus obtaining a dataset containing a higher number of smaller than initial images, but to greater extent filled with labeled pixels. In this step, we also excluded all the image parts, where the defects were not visible, to slightly counteract biasing the classifiers towards *background* class. The final yield of images for every class in both training and testing datasets for defect detection along with the total sum of damaged elements is summarized in Tab. 2.

For the component segmentation task, we mostly used the initial dataset provided with the task. The only modifications that we made were the use of images modified for the background-foreground segmentation and re-coding the labels' numbers to integers – this way, as image labels we obtained images containing only a single channel.

Lastly, we also modified the dataset for the last task – the damage state assessment. However, this task turned out to be more demanding, as additional problems occurred. Some of the elements, while having their state decreased (by implication by the defects occurring on them), had no visible defects that were indicated by provided labels. Others had conflicting damage state designations even though the entire component should have been classified to a single state. Furthermore, the dataset containing structure elements did not separate their individual segments (for instance segmentation task), even though two segments of the same element may have differed in condition. The latter two of the problems are depicted in Fig. 2.

The problems described are beyond the ability to solve them using dataset manipulation or even an elaborate ML algorithm, and are a direct result of the way the dataset was
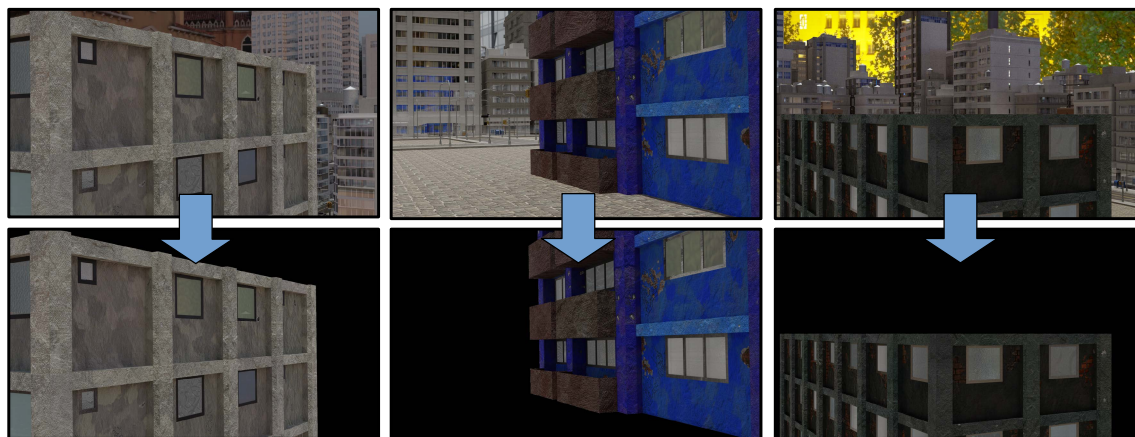
FIGURE 1. Initial dataset images (top) and reworked images by background removal (bottom).
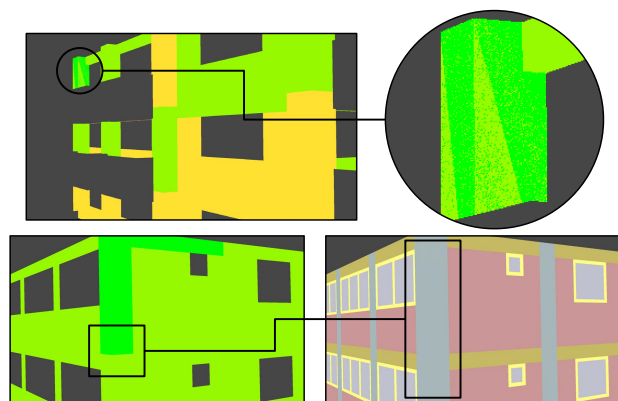

FIGURE 2. Example of damage state marking problems: contrasting markings on a single element (top), no clear division of component segments (bottom).

constructed. As these were human errors made during the development of the dataset, possibly due to work fatigue, this draws all the more attention to the need for careful preparation of training data. However, we still attempted to solve them by approaching them in a more general way. Rather than focusing on elements' defects themselves, we decided to focus on their surfaces, and assess them by their general appearance.

To do so, we once again focused on extracting surface images from single elements. However, this time, to retain as much information about the surface of a single element as there was, we extracted them with their minimum area, rotated rectangular bounding boxes and warped them into square shape of $224 \times 224px$. An example of such transformation is depicted in Fig. 3. Such rectangular images were then used as an input to an image classifier, where the class was indicated by elements' damage state. The final yield of textures used for training is summarized in Tab. 3. While considering the dataset, it is important to notice that in this task there are large inequalities between classes that, when mishandled, can affect the classifier.

FIGURE 3. Surface images warped to square shape.

TABLE 3. Datapoints extracted for damage state classes after the dataset modification.

Damage state	Training dataset	Testing dataset	Total extracted components
No damage	1025	254	**1279**
Light damage	21811	5325	**27136**
Moderate damage	35762	8825	**44587**
Severe damage	2462	613	**3075**

3) THE IMPACT OF DATASET MANAGEMENT

While the purpose of the modifications that we did to the dataset may be unclear at first, they have greatly assisted us in achieving high accuracy of the models we trained. Although in the next sections we'll be focusing on our most successful models developed with the data modified as described above, we will still provide the results from our other models that used the dataset directly or through simple extraction of data for additional comparison.

TABLE 4. Metrics of foreground detection model.

Metric	Value [%]
mean IoU	98.88
mean pixel accuracy	99.28
IoU - foreground	99.46
IoU - background	98.30
Foreground pixel accuracy	99.87
Background pixel accuracy	98.70

B. TASK 0 – DETECTING FOREGROUND OBJECTS

As mentioned previously in Section III-A, we added the initial task marked as *Task 0* to differentiate between objects in background and foreground more easily. To perform the segmentation, we used the previously prepared dataset along with model trained with Detectron 2 framework, taken from Detectron 2 model ZOO repository [39]. Our choice of framework was based on the multiplicity of training options, choices of available architectures, and the ease of inference with the model. Detectron also allows for changing the type of task (*eg.* object detection, semantic segmentation etc.) with only minimal changes to the code, for the ease of testing different approaches.

During our tests we found out that the model giving us optimal results considering both accuracy and inference speed was the Faster R-CNN [40] based on ResNet 50 model [41] with additional fully convolutional head as feature proposal network, trained initially for 12 epochs on ImageNet [42] dataset. We fine-tuned the model for 15 additional epochs using learning rate of $2.5e - 4$ with batches containing two images each and no additional learning rate decay.

The metrics of the model we obtained after training, including both – mean values and values per class of IoU and accuracy per pixel, are presented in Tab. 4. We found them satisfactory, as masks resulting from inference with the model hardly differ from the ones we manually prepared for training, and can be used in subsequent tasks without additional modifications.

C. TASK 1 – DETECTING DEFECTS

Similarly to *Task 0*, *Task 1* also focuses on image semantic segmentation with intention to detect cracks, spalling and exposed rebar on the surface of the structure. This time, however, the objects sought in the image are much smaller and more sparse across the dataset. As stated previously in Section III-A, we attempted to mitigate this problem by extracting only those parts of the dataset, where defects were present. Also, due to the collisions of labels described in the description of the dataset, we had to train one model per defect separately.

With such prepared dataset and task, we used the same architecture as in *Task 0* for three models, as it again proved to provide the best balance between accuracy and inference time. To achieve the best results, we changed training parameters – this time each model was fine-tuned for additional
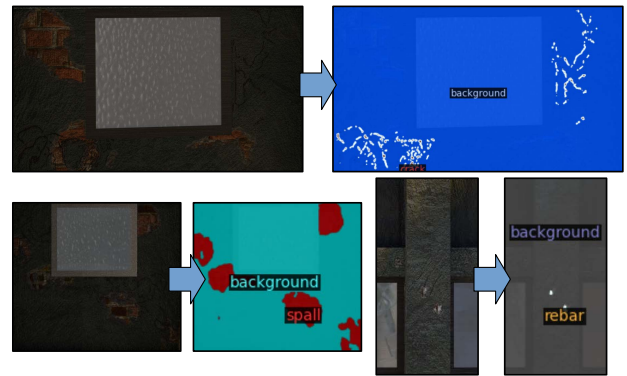
FIGURE 4. Example predictions for each model: cracking (top), spalling (bottom left) and exposed rebar (bottom right).

50 epochs with learning rate decaying from initial $2.5e - 4$ by $\frac{2.5e-4}{50}$ every epoch. The resulting metrics (IoU and pixel accuracy per defect) calculated on the testing dataset after training are presented in Tab. 5, and sample images from model inference are depicted in Fig. 4. Tab. 5 also provides comparison between models trained with modified and unmodified dataset.

As seen in Tab. 5, our modifications to the dataset helped with the problem of model bias towards detecting *background* class and made them more balanced, especially in *Exposed rebar* task, where pixel accuracy increased over eightfold. In Fig. 4 it can also be seen how small an area the searched defect could be, especially considering the initial, much greater size of the input image, what once again justifies the best result achieved by the largest type of defect (spalling). The results however still are not perfect, and other means such as random undersampling of dominant class or training performed for image patches could also be considered.

D. TASK 2 – SEGMENTING OUT COMPONENTS

Task 2 once again focuses on image semantic segmentation, so we used methods similar to *Task 1* and *2*. We used Detectron 2 framework to fine tune ResNet 50 with Feature Proposal Network on dataset with background objects removed. We achieved satisfactory results when the model was trained for additional 15 epochs and learning rate an order of magnitude greater than for the previous tasks – $2.5e - 3$. We also did not use learning rate decay schedule this time.

The resulting metrics we obtained after the training for both datasets – the initial one and after background removal with model from *Task 0* – are summarized in Tab. 6. Moreover, Figs. 5 and 6 show the example results for component detection for both cases – when background is and is not removed.

As can be seen in Fig. 5, 6 and Tab. 6, removing the background has significantly improved the performance of the model, especially in the case of detecting slabs and backgrounds, whereas the accuracy with visible background was negligible. The only elements whose pixel-by-pixel

TABLE 5. Metrics of defects detection models.

| Metric	After dataset modification			Before dataset modification		
	Cracking	Spalling	Exposed rebar	Cracking	Spalling	Exposed rebar
Mean IoU [%]	**64.16**	**89.42**	**64.86**	63.83	88.30	51.81
Defect IoU [%]	**29.73**	**80.12**	**29.81**	28.57	77.45	3.62
Background IoU [%]	98.59	98.73	99.91	**99.08**	**99.13**	**99.99**
Mean pixel accuracy [%]	**67.51**	**93.69**	**69.58**	67.39	92.58	52.25
Defect pixel accuracy [%]	**35.34**	**87.96**	**39.19**	35.02	85.51	4.50
Background pixel accuracy [%]	99.68	99.43	99.97	**99.76**	**99.64**	**99.99**
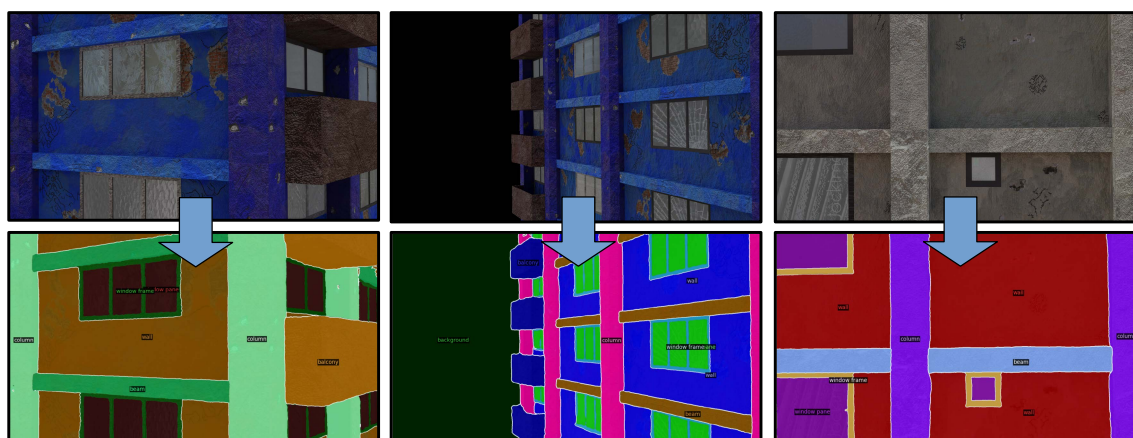
FIGURE 5. Example results obtained after performing initial background removal.
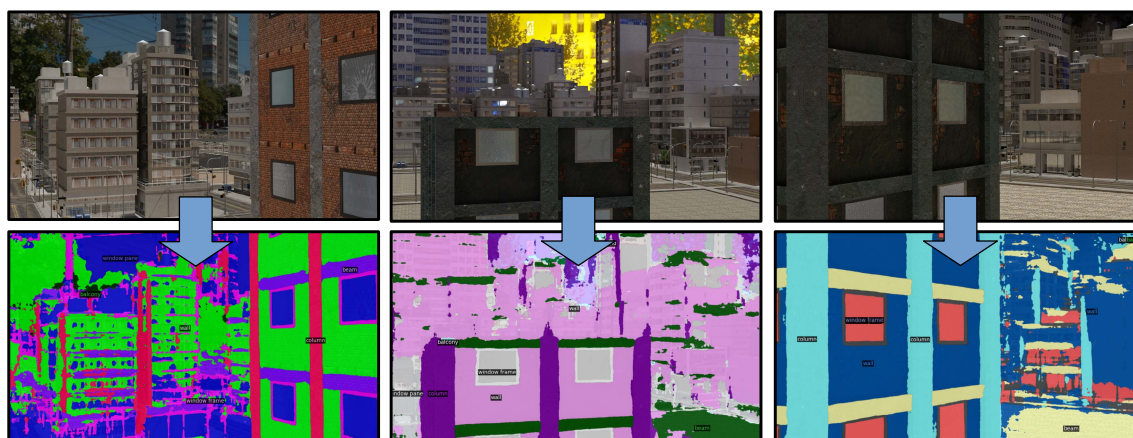

FIGURE 6. Example results without background removal.

detection accuracy slightly increased without background removal were parts of windows, and this may have been due to their specific shape in the dataset, which windows found in buildings in the background did not have.

At the same time, it should be noted that the model working on images without first removing the background still correctly recognizes the elements of objects in the background to some extent, even though they differ significantly in appearance from those in the dataset. This indicates, on the one hand, that the model was correctly chosen for the task, since it was able to generalize the acquired knowledge beyond the provided dataset, and on the other hand, the necessity to apply the first step of the analysis – removing the background, since the results of the actually conducted measurements would be muddied by the occurrence of false-positive detections.

TABLE 6. Metrics of component segmentation model.

Metric	Wall	Beam	Column	Window frame	Window pane	Balcony	Slab	Background	Mean
	After background removal								
IoU [%]	**91.42**	**89.63**	**84.25**	**82.14**	**97.31**	**96.84**	**90.81**	**99.89**	**91.54**
Pixel accuracy [%]	**98.81**	**93.73**	**86.27**	87.38	97.86	**98.64**	**94.51**	**99.96**	**94.65**
	Without background removal								
IoU [%]	70.53	57.31	69.69	59.35	76.31	87.47	0.82	0.35	52.73
Pixel accuracy [%]	98.70	92.79	86.05	**87.48**	**98.06**	97.88	0.86	0.35	70.27

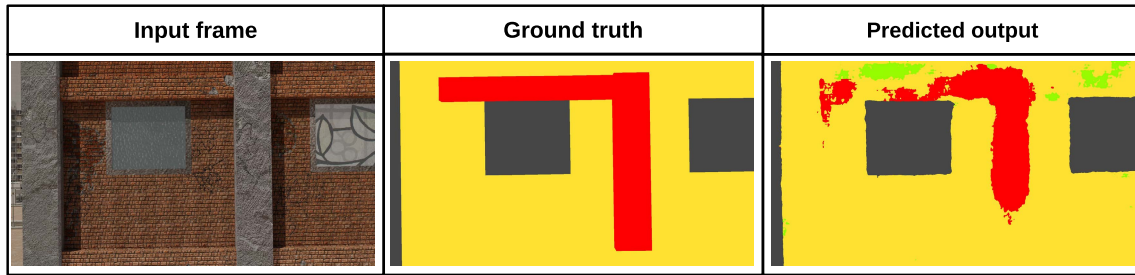| Input frame	Ground truth	Predicted output

FIGURE 7. Single model approach compared to ground truth label.

TABLE 7. Metrics of damage state detection model.

Model	Accuracy per class [%]				Average accuracy [%]	Average F1 score [%]
	No damage	Light damage	Moderate damage	Severe damage		
VGG16	28.74	74.25	83.13	59.11	78.06	77.83
Densenet	28.35	78.14	86.33	61.17	81.40	**81.13**
ResNet	28.34	77.56	**86.93**	58.30	**81.42**	81.09
Xception	19.29	75.42	85.44	57.21	79.59	79.18
Decision tree	12.89	70.61	76.61	**61.96**	72.98	73.19
Random forest	0.00	81.55	77.50	38.36	76.00	75.29
Naive Bayes	**98.05**	21.36	39.31	52.65	34.45	43.66
Naive Bayes normed	0.00	**98.62**	13.97	28.73	35.73	35.87
Single model baseline	*23.68*	*68.66*	*80.40*	*53.02*	*64.67*	*69.12*

E. TASK 3 – ASSESSING THE DAMAGE STATE

In the first attempt to solve *Task 3* we intended to use the relationship between an element, its total area, and the relative area of each defect as a one-dimensional input vector for a shallow machine learning algorithm. We used decision trees with maximal depth of 59 splits, random forest with total of 200 models and naive Bayesian algorithm with and without data normalization. The training data we collected came from the partitioning of the structure into elements by the model trained for *Task 2* and the damage detected using the model from *Task 1*. This way we managed to gather 61 058 data points for training and 15 036 for testing. The final vector along with its exemplary data presented itself in this way:

$$\{E_t, E_{sr}, C_r, R_r, S_r\} = E_s \qquad (1)$$

eg. $\{3.0, 0.0245, 5.90e-5, 3.94e-5, 9.60e-3\} = 3.0$

where:

E_t: *element type (mapped to int value)*
E_{sr}: *element size to image size ratio*
C_r: *crack size to element size ratio*
R_r: *exposed rebar size to element size ratio*
S_r: *spalling size to element size ratio*
E_s: *element damage state*

However, despite a vast dataset for model development, this approach proved to be inaccurate. It was mostly due to the component defect not visible in the frame of the image and occasional false negative indications of the model from *Task 1*. This led to many instances of data points where even though the element had no visible defect, its state was described as damaged. The results of models trained using this approach are shown in Tab. 7.
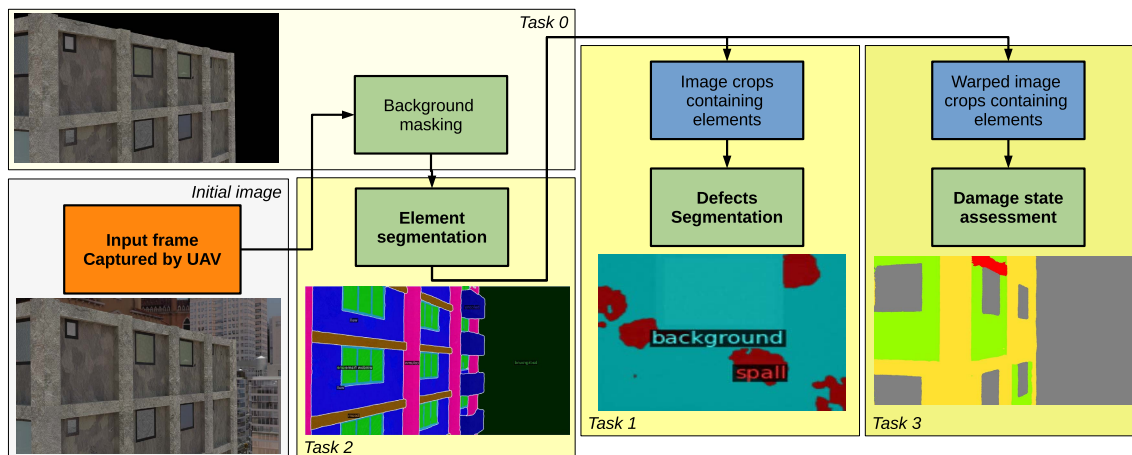
FIGURE 8. The final pipeline of our solution.

To increase the accuracy of classification, we turned to methods based on a single model that uses computer vision to determine the damage state of elements. We called this approach *single model baseline*, where we used Detectron 2 directly on the visual data. Unfortunately, this approach turned out to be ineffective, not achieving the best result in any of the categories, as shown in Tab. 7. In Fig. 7, we also show the result of the model inference compared to the label.

To improve damage state assessment, as stated previously in Section III-A, we chose the method that is approaching the problem in a more general way – by image recognition rather than inference based on defects found by other models. Again, the part of the task related to the semantic segmentation of the structure components has been done before, and we can continue on the segmentation done with high accuracy in *Task 2* to focus on image recognition task.

In order to perform the image recognition we used KrakN framework [43] to train 4 separate CNN models using transfer learning technique with 3-fold cross validation on dataset containing warped images of elements' surfaces. Models we used as feature extractors for single layer logistic regression classifier were: VGG16 [44], Densenet [45], ResNet and Xception [46], all of which were pre-trained on the ImageNet dataset. The results of the training are summarized in Tab. 7 – note that this time the only measured metrics were accuracy per class, average accuracy and average F1 score which describes balance between precision and recall. This time IoU or pixel accuracy will be derived from the segmentation performed in *Task 2* and have no impact on the overall score.

As seen in Tab. 7, while shallow machine learning techniques can provide near perfect accuracy for single defect types, they tend to be highly biased and thus resulting in low average accuracy and F1 score. On the other hand, deep CNN models provide more balanced results across classes and do not tend to over-promote particular predictions. At the same time, once again, it is important to account for highly imbalanced dataset. However, even with properly performed random undersampling with the best performing models

(ResNet and Densenet), we did not manage to achieve better results.

It is also possible to consider using model ensembles with only slightly biased members and use a voting algorithm to pick the most probable prediction.

F. THE FINAL PIPELINE
In Fig. 8 we present the final pipeline of our solution.

First, the image retrieved by the UAV is stripped of background objects using the model trained for *Task 0*. Then, the model from *Task 2* performs structure segmentation to its components. These components after additional processing are used in further tasks. For *Task 1*, the segmented parts of the structure come as rectangular frames of the image, and then damage is detected on them using the model for semantic segmentation. For *Task 2*, they are further warped to a square shape, after which the CNN model performs an image recognition task on them.

While the tasks performed by the pipeline are backwards dependent, the great advantage of this approach is to divide one large task into a number of smaller tasks that can be performed with greater accuracy. At the same time, this approach allows for wider freedom in the manipulation of the models that make up the workflow – models can be freely exchanged or fine-tuned with new batches of data as single nodes responsible for individual tasks, thus maintaining the integrity of the solution.

IV. DISCUSSION
Considering the phasing of successive tasks performed by with our pipeline from Section III-F, we can expect marginally lower accuracy of the whole workflow, albeit thanks to nearly perfect accuracy of the first stages, the reduction will not be significant. Also, some of the models we presented in the pipeline can certainly be improved in the terms of their particular accuracy. However, due to the specific nature of the dataset, much higher prediction accuracy cannot be expected.

Still, there are methods that can aid the accuracy of the algorithm. One is the previously mentioned use of small

image patches to train a model that performs damage detection. This way the problem of highly imbalanced dataset can be avoided, and models would not be as much biased towards *background* class. This problem can also be mitigated with the use of random oversampling techniques that could help by expanding underrepresented classes or with the use of weighted loss function – focal loss [47]. However, while the latter could help with the training process (although during the performed tests, the gain was insignificant), the former could lead to model biased towards a specific shape of defect of single class.

The last method that can have a positive impact on the accuracy of the algorithm is to transform the flat representation of objects in the image into a 3D point cloud. This way, the problem of defects not visible on an element in a single frame may no longer be relevant, since the defect detection model would operate on the entire 3D object. The transformation could be performed with photogrammetric methods employing RGB-D sensors like in *eg.* [48]. However, this approach also has some drawbacks. First of them is the need of depth data associated with RGB images, that was not available for the task. Second is the necessity of heavy computations performed during preprocessing of the data, excluding this approach from on-site usage. Additionally, machine learning models trained in 3D point cloud environment require significantly more computing power than those operating on RGB images.

V. CONCLUSION

The detection of earthquake caused defects to buildings is an extremely important issue, affecting both the safety of building occupants and the efficient management and restoration of the affected area. The use of UAVs for this purpose is an important step in the desired direction by which the effectiveness of immediate post-earthquake response can be increased. Moreover, fully autonomous vehicles armed with algorithms that enable automatic and immediate assessment of structure technical condition will allow more efficient management of emergency services human resources and labor intensity reduction of the whole process. However, for that purpose, vast datasets and robust machine learning models are needed.

While real-world datasets like PEER Hub ImageNet [49] or Mexico City [50] earthquake datasets are available, they focus on high-magnitude earthquakes, which is beyond the scope of continuous monitoring of structures after minor earthquakes. Other datasets that contain individual defect types, on the other hand, do not consider a holistic approach to the structure monitoring problem. For this reason, until actual imaging data is collected from low magnitude earthquakes in highly urbanized areas, synthetic datasets are one of the best ways to pretrain neural algorithms.

In this paper, we proposed a workflow along with dataset transformations and models trained for the tasks of foreground objects, defects, and structure components detection, as well as image recognition methods for elements' damage state assessment. In the work we presented, we used a synthetic dataset that can serve as a base for training machine learning models and achieved satisfactory results in all four tasks.

During our work, we also identified some recurring issues with synthetic datasets that can significantly affect the trained models. Although synthetic data, especially those that simulate extremely rare and large-scale events, are essential in building training datasets, they must be free of errors in the objects represented and the classes given. At the same time, even a flawlessly prepared dataset cannot be the only source of information for a machine learning algorithm – for this, one must also be exposed to real data in the final training phase. With this in mind, a worthwhile concept to consider while creating a synthetic dataset is instead of a single, homogeneous dataset, creating multiple smaller but more diverse datasets for which the real-world data would always fit closer within the spectrum of variants.

Our work has also pointed possible directions for further development of the proposed methods. As we mentioned in Section IV, a next step with a holistic approach to assessing the condition of a building would be a much-needed improvement to the overall process of structure maintenance. However, it requires creating a three-dimensional model of the building that is faithful to the original, using RGB-D cameras, photogrammetry methods or multi-scale approach. At the same time, the effort should be put into developing diverse synthetic datasets along with their real-life counterparts.

REFERENCES

[1] C. Wai-Fah and S. Charles, *Earthquake Engineering*. New York, NY, USA: CRC Press, 2003.

[2] Infoplease and Sandbox Networks. (2017). *Deadliest Earthquakes on Record*. [Online]. Available: https://www.infoplease.com/world/earthquakes/deadliest-earthquakes-record

[3] J. Oluwafemi, O. Ofuyatan, O. Sadiq, S. Oyebisi, J. Abolarin, and K. Babaremu, "Review of world earthquakes," *Int. J. Civil Eng. Technol.*, vol. 9, pp. 440–464, Sep. 2018.

[4] D. Koren and K. Rus, "Assessment of a city's performance under different earthquake scenarios," in *Proc. 1st Croatian Conf. Earthq. Eng. (1CroCEE)*, Zagreb, Croatia, Mar. 2021. [Online]. Available: https://crocee.grad.hr/event/1/contributions/52/attachments/11/88/009_CroCEE_ID_155.pdf, doi: 10.5592/CO/1CroCEE.2021.155.

[5] P. Hua, Z. Feng-Xin, and G. Meng-Tan, "Study on the characteristics of earthquake's impact on cities," *Acta Seismol. Sinica*, vol. 17, pp. 222–229, Mar. 2004.

[6] A. Wada, I. Towhata, K. Tamura, and Z. Qu, "Moving toward cities where earthquakes will not cause a grievous disaster," *Jpn. Archit. Rev.*, vol. 1, no. 4, pp. 410–418, Oct. 2018.

[7] S. Eem, B.-J. Yang, and H. Jeon, "Earthquake damage assessment of buildings using opendata in the Pohang and the Gyeongju earthquakes," *J. Earthq. Eng. Soc. Korea*, vol. 22, no. 3, pp. 121–128, Apr. 2018.

[8] M. Erener, C. Hızal, D. Erdogan, A. Sezer, A. Nuhoglu, E. Ercan, C. Kıncal, M. Akgun, and O. Ozdag, "Geotechnical and structural assessment of the damages in Bayrakli region after Samos 2020 earthquake," in *Proc. 6th Int. Conf. Earthq. Eng. Seismol. (6ICEES)*, Gebze, Turkey, Oct. 2021. [Online]. Available: https://www.researchgate.net/publication/355888035_GEOTECHNICAL_AND_STRUCTURAL_ASSESSMENT_OF_THE_DAMAGES_IN_BAYRAKLI_REGION_AFTER_SAMOS_2020_EARTHQUAKE

[9] M. Maeda and H. Alwashali, "Damage due to earthquakes and improvement of seismic performance of reinforced concrete buildings in Japan," in *Science and Technology Research Partnership for Sustainable Development*. Japan, Mar. 2018, ch. 6, pp. 361–372, doi: 10.1007/978-3-319-74724-8_24.

[10] K. A. R. V. D. Kahandawa, N. D. Domingo, K. S. Park, and S. R. Uma, "Earthquake damage estimation systems: Literature review," *Proc. Eng.*, vol. 212, pp. 622–628, 2018.

[11] D. Lang, "Earthquake damage and loss assessment—Predicting the unpredictable," Ph.D. thesis, Univ. Bergen, Norway, May 2013, p. 334, doi: 10.13140/RG.2.2.30518.80966.

[12] G. Amiri and E. Rajabi, "Effects of consecutive earthquakes on increased damage and response of reinforced concrete structures," *Comput. Concrete*, vol. 21, pp. 55–66, Jan. 2018.

[13] M. Astroza and R. Astroza, "Comparison of the damages produced by chilean earthquakes with different seismogenic sources," in *Proc. 33rd Sud-Amer. Conf. Struct. Eng.*, Santiago, Chile, May 2008. [Online]. Available: https://www.researchgate.net/publication/258407195_Comparison_of_the_Damages_Produced_by_Chilean_Earthquakes_with_Different_Seismogenic_Sources

[14] H. Stone, V. Putrino, and D. D'Ayala, "Earthquake damage data collection using omnidirectional imagery," *Frontiers Built Environ.*, vol. 4, p. 51, Sep. 2018.

[15] Z. Shang, L. Sun, Y. Xia, and W. Zhang, "Vibration-based damage detection for bridges by deep convolutional denoising autoencoder," *Struct. Health Monit.*, vol. 20, no. 4, pp. 1880–1903, Jul. 2021.

[16] M. Rabah, A. Elhattab, and A. Fayad, "Automatic concrete cracks detection and mapping of terrestrial laser scan data," *NRIAG J. Astron. Geophys.*, vol. 2, no. 2, pp. 250–255, Dec. 2013.

[17] M. Rezaeian, "Assessment of earthquake damages by image-based techniques," Doctor Sci., Dept. Elect. Eng., Institut für Geodäsie und Photogrammetrie an der Eidgenössischen Technischen Hochschule, Zürich, Switzerland, Jan. 2010.

[18] R. S. Adhikari, O. Moselhi, and A. Bagchi, "Image based retrieval of concrete crack properties," *Gerontechnology*, vol. 11, no. 2, pp. 180–194, Jun. 2012, doi: 10.4017/gt.2012.11.02.458.00.

[19] H.-X. He, J.-C. Zheng, L.-C. Liao, and Y.-J. Chen, "Damage identification based on convolutional neural network and recurrence graph for beam bridge," *Struct. Health Monit.*, vol. 20, no. 4, pp. 1392–1408, Jul. 2021, doi: 10.1177/1475921720916928.

[20] Q. Mei, M. Gül, and M. R. Azim, "Densely connected deep neural network considering connectivity of pixels for automatic crack detection," *Autom. Construct.*, vol. 110, Feb. 2020, Art. no. 103018.

[21] A. Zhang, K. C. P. Wang, Y. Fei, Y. Liu, S. Tao, C. Chen, J. Q. Li, and B. Li, "Deep learning–based fully automated pavement crack detection on 3D asphalt surfaces with an improved CrackNet," *J. Comput. Civil Eng.*, vol. 32, no. 5, pp. 1–14, Sep. 2018.

[22] F. T. Ni, J. Zhang, and Z. Q. Chen, "Pixel-level crack delineation in images with convolutional feature fusion," *Struct. Control Heal. Monit.*, vol. 26, no. 1, pp. 1–18, 2019.

[23] M. Meng, K. Zhu, K. Chen, and H. Qu, "A modified fully convolutional network for crack damage identification compared with conventional methods," *Model. Simul. Eng.*, vol. 2021, pp. 1–14, Nov. 2021.

[24] Y.-C. Lin, J.-X. Chang, and Y.-C. Su, "Developing construction defect management system using BIM technology in quality inspection," *J. Civil Eng. Manag.*, vol. 22, no. 7, pp. 903–914, Jul. 2016.

[25] M. Sadi, F. Nateghi, F. Mohammadi, and A. Abdullah, "Mobile application for building defect management system," *Indian J. Sci. Technol.*, vol. 7, no. 2, pp. 174–179, Feb. 2013.

[26] B. Li, K. Ushiroda, L. Yang, Q. Song, and J. Xiao, "Wall-climbing robot for non-destructive evaluation using impact-echo and metric learning SVM," *Int. J. Intell. Robot. Appl.*, vol. 1, no. 3, pp. 255–270, Sep. 2017.

[27] L. Zhang, W. Xu, L. Zhu, X. Yuan, and C. Zhang, "Study on pavement defect detection based on image processing utilizing UAV," *J. Phys., Conf. Ser.*, vol. 1168, Feb. 2019, Art. no. 042011.

[28] Y. Wu, Y. Qin, and L. Jia, "Research on rail surface defect detection method based on UAV images," in *Proc. Prognostics Syst. Health Manag. Conf. (PHM-Chongqing)*, Oct. 2018, pp. 553–558.

[29] S. Yang, Z. Shen, X. Wang, T. Bai, Y. Ji, Y. Jiang, X. Liu, X. Dong, C. Li, Q. Han, J. Lu, and G. Xiong, "UAV assisted bridge defect inspection system," in *Proc. Int. Conf. Intell. Sci.*, Jan. 2018, pp. 401–411.

[30] P. Addabbo, A. Angrisano, M. L. Bernardi, G. Gagliarde, A. Mennella, M. Nisi, and S. Ullo, "A UAV infrared measurement approach for defect detection in photovoltaic plants," in *Proc. IEEE Int. Workshop Metrol. Aerosp. (MetroAeroSpace)*, Jun. 2017, pp. 345–350.

[31] J. P. Burgard, J.-P. Kolb, H. Merkle, and R. Münnich, "Synthetic data for open and reproducible methodological research in social sciences and official statistics," *AStA Wirtschafts-und Sozialstatistisches Archiv*, vol. 11, nos. 3–4, pp. 233–244, Dec. 2017.

[32] E. A. Lopez-Rojas and S. Axelsson, "Poster abstract: Using financial synthetic data sets for fraud detection research," in *Proc. 17th Int. Symp. Res. Attacks, Intrusions Defenses*, Gothenburg, Sweden, vol. 8688, Sep. 2014. [Online]. Available: https://www.researchgate.net/publication/265736247_Poster_Abstract_Using_Financial_Synthetic_Data_Sets_for_Fraud_Detection_Research

[33] T. N. Arvanitis, S. White, S. Harrison, R. Chaplin, and G. Despotou, "A method for machine learning generation of realistic synthetic datasets for validating healthcare applications," *Health Informat. J.*, vol. 28, no. 2, Jan. 2022, Art. no. 146045822210770, doi: 10.1177/14604582221077000.

[34] S. D. P. Mendonca, Y. P. D. S. Brito, C. G. R. D. Santos, R. D. A. D. Lima, T. D. O. D. Araujo, and B. S. Meiguins, "Synthetic datasets generator for testing information visualization and machine learning techniques and tools," *IEEE Access*, vol. 8, pp. 82917–82928, 2020.

[35] M. Żarski and B. Wójcik. (2021). *IC SHM Repository*. [Online]. Available: https://github.com/MatZar01/IC_SHM_P2

[36] SAIL. (2021). *IC-SHM 2021 the 2nd International Competition for Structural Health Monitoring*. [Online]. Available: https://sail.cive.uh.edu/ic-shm2021/

[37] R. TaoZhang, "UAV 3D mapping with RGB-D camera," in *Proc. Chin. Autom. Congr. (CAC)*, Oct. 2017, pp. 2727–2731.

[38] F. Khan, S. Salahuddin, and H. Javidnia, "Deep learning-based monocular depth estimation methods—A state-of-the-art review," *Sensors*, vol. 20, no. 8, p. 2272, Apr. 2020.

[39] Y. Wu, A. Kirillov, F. Massa, W.-Y. Lo, and R. Girshick. (2019). *Detectron2*. [Online]. Available: https://github.com/facebookresearch/detectron2

[40] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 1–9. [Online]. Available: https://proceedings.neurips.cc/paper/2015/file/14bfa6bb14875e45bba028a21ed38046-Paper.pdf

[41] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *CoRR*, vol. abs/1512.03385, pp. 1–12, Dec. 2015.

[42] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 248–255.

[43] M. Żarski, B. Wójcik, and J. A. Miszczak, "KrakN: Transfer learning framework and dataset for infrastructure thin crack detection," *SoftwareX*, vol. 16, Dec. 2021, Art. no. 100893.

[44] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, *arXiv:1409.1556*.

[45] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 2261–2269.

[46] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1800–1807, doi: 10.1109/CVPR.2017.195.

[47] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollar, "Focal loss for dense object detection," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 318–327.

[48] M. Zollhöfer, P. Stotko, A. Görlitz, C. Theobalt, M. Nießner, R. Klein, and A. Kolb, "State of the art on 3D reconstruction with RGB-D cameras," *Comput. Graph. Forum*, vol. 37, no. 2, pp. 625–652, 2018.

[49] University of Berkeley. (2020). *Peer Hub Imagenet (Ø-Net): A Large-Scale Multi-Attribute Benchmark Dataset of Structural Images*. [Online]. Available: https://peer.berkeley.edu/news/new-peer-report-201907-peer-hub-imagenet-ø-net-large-scale-multi-attribute-benchmark-dataset

[50] DesignSafe-CI, Texas. (2017). *Buildings Surveyed After 2017 Mexico City Earthquakes*. [Online]. Available: https://doi.org/10.17603/ds2-4b2n-at54

MATEUSZ ŻARSKI received the B.S. and M.S. degrees in civil engineering from the Silesian University of Technology, Gliwice, Poland, in 2015 and 2017, respectively, where he is currently pursuing the Ph.D. degree in civil engineering.

Since 2021, he has been a Research Assistant with the Institute of Theoretical and Applied Informatics, Polish Academy of Sciences, Gliwice. His research interests include applied artificial intelligence in the field of civil engineering and the Internet of Things application.

BARTOSZ WÓJCIK received the B.S. and M.S. degrees in civil engineering from the Silesian University of Technology, Gliwice, Poland, in 2014 and 2016, respectively, where he is currently pursuing the Ph.D. degree in civil engineering.

Since 2021, he has been a Research Assistant with the Institute of Theoretical and Applied Informatics, Polish Academy of Sciences, Gliwice. His research interest includes applied informatics in the context of civil engineering.

BARTLOMIEJ BLACHOWSKI received the M.Sc. degree from the Silesian University of Technology, in 2000, and the Ph.D. and Habilitation degrees from the Institute of Fundamental Technological Research (IPPT), Polish Academy of Sciences (PAN), in 2005 and 2020, respectively. He is currently a member of the Scientific Staff of the IPPT, PAN. His research interests include wide-sense structural dynamics, in particular strategies of vibration control, algorithms for optimal structural design, and the use of sensor networks for structural health monitoring. He is equally interested in modern computational methods of structural analysis, experimental techniques such as operational modal analysis, and hardware solutions for wireless acceleration measurements of engineering structures.

JAROSŁAW A. MISZCZAK received the M.Sc. degree with specialization in theoretical physics from the University of Silesia, Katowice, Poland, in 2003, the Ph.D. degree in computer science from the Institute of Theoretical and Applied Informatics, Polish Academy of Sciences, Gliwice, Poland, in July 2008, and the D.Sc. degree from the Silesian University of Technology, in 2014.

He is currently a Researcher at the Institute of Theoretical and Applied Informatics, Polish Academy of Sciences. His research interests include quantum information theory, foundations of quantum mechanics, scientific computing, complex systems, and theory of programming languages.

MARIUSZ OSTROWSKI received the M.Sc. degree from the Lublin University of Technology, in 2016. He worked with space industry (UV2VIS project, which is related with Sentinel-5 mission), from 2017 to 2018. He also dealt with modeling of electromechanical systems and multiphysics problems for industry with aid of Matlab® software and other similar tools. He took sixth place in IC-SHM 2021 by development of a framework for computer vision-based structural health monitoring. He currently works at the Institute of Fundamental Technological Research (IPPT), Polish Academy of Sciences (PAN), Warsaw, Poland. His current research covers the fields of semi-active modal control of vibration, optimal sensors placement, and structural health monitoring. His research interests include modal and optimal control, mathematical modeling, model updating, optimization, and energy harvesting.

• • •