remote sensing

MDPI

Letter

Towards Real-Time Building Damage Mapping with Low-Cost UAV Solutions

Francesco Nex *, Diogo Duarte, Anne Steenbeek and Norman Kerle

Faculty of Geo-Information Science and Earth Observation (ITC), University of Twente, 7500 AE Enschede, The Netherlands; d.duarte@utwente.nl (D.D.); h.a.steenbeek@student.utwente.nl (A.S.); n.kerle@utwente.nl (N.K.)
* Correspondence: f.nex@utwente.nl; Tel.: +31-534896639

Received: 31 December 2018; Accepted: 28 January 2019; Published: 1 February 2019



Abstract: The timely and efficient generation of detailed damage maps is of fundamental importance following disaster events to speed up first responders' (FR) rescue activities and help trapped victims. Several works dealing with the automated detection of building damages have been published in the last decade. The increasingly widespread availability of inexpensive UAV platforms has also driven their recent adoption for rescue operations (i.e., search and rescue). Their deployment, however, remains largely limited to visual image inspection by skilled operators, limiting their applicability in time-constrained real conditions. This paper proposes a new solution to autonomously map building damages with a commercial UAV in near real-time. The solution integrates different components that allow the live streaming of the images on a laptop and their processing on the fly. Advanced photogrammetric techniques and deep learning algorithms are combined to deliver a true-orthophoto showing the position of building damages, which are already processed by the time the UAV returns to base. These algorithms have been customized to deliver fast results, fulfilling the near real-time requirements. The complete solution has been tested in different conditions, and received positive feedback by the FR involved in the EU funded project INACHUS. Two realistic pilot tests are described in the paper. The achieved results show the great potential of the presented approach, how close the proposed solution is to FR' expectations, and where more work is still needed.

Keywords: UAV; building damage; detection; real-time; deep learning; photogrammetric mapping

1. Introduction

When a catastrophic event occurs, timely and efficient response and rescue are the main elements to increase the chances to save trapped victims under collapsed structures. In this scenario, first responders (FR) visually inspect the affected area and decide where and how to start the rescue activities. The inspection is usually performed with ground vehicles or on foot, though this often does not yield a complete overview of the area to allow an efficient prioritization of rescue measures, whilst also exposing rescue forces to unnecessary hazards.

Remote sensing techniques have been increasingly adopted to map disaster areas in recent decades [1–3] as witnessed by a rapidly growing amount of research works dealing with optical [4–6], radar [7,8], and laser [9,10] data with each sensor giving different advantages. Except for subtle damage patterns that deformation-sensitive radar can identify particularly well, optical data are usually preferrable to detect building damage [11] in the early hours after the event as images are more easily interpretable by human operators who still typically carry out the initial damage map production manually [1]. Existing emergency mapping services such as the Copernicus Emergency Management Service [12] have adopted synoptic satellite imagery for this task as they can cover large areas and are available a few hours after a disaster. The main objective of these early impact analyses is to produce georeferenced data of the affected areas, to support search and rescue efforts and humanitarian actions.

Crucial information is the identification of the disaster areas and the estimation of the number of people affected. However, these satellite images typically have limited spatial resolution (0.3–1 m ground sampling distance, GSD) and the uncertainty and subjectivity in the identification of hazard-induced damages can often be solved by using higher resolution images [13]. On the other hand, airborne images [14–16] are often not available in remote places where the delivery of mapping airplanes is not feasible in an emergency timeframe. Therefore, in situ surveys are often preferred.

In this regard, the massive uptake of unmanned aerial vehicles (UAV) in the market has created the possibility to collect very high-resolution images of a given scene, whilst also readily coupling vertical with oblique imaging [14,15]. Their higher portability and their lower instrument cost, compared to manned platforms, represent a good alternative to traditional data sources. This has been witnessed by the increasing number of UAV footages that have been acquired in the very last events and are frequently made available on the internet. Many rescue departments have also added UAVs to the regular equipment of their field teams [17,18]. However, flight execution and image interpretation are still performed manually (using UAVs for inspections from the sky), requiring a dedicated operator. This can be often perceived as a waste of time by first responders given the strong time pressure on their work. This was a common feedback by the first response end users in the recently concluded EU funded project INACHUS [19]. Rescuers require a flexible solution that is: (i) able to deliver timely data (as near real-time as possible) and with frequent update possibility. (ii) Easy to use and that yields data that are quick to interpret also by non-remote sensing experts. The extent of the area, the practical operational problems faced in emergency conditions and possible legal restrictions still in place in the affected area should also be considered. In this regard, (iii) low-cost and (iv) easily deployable (i.e., portable and light) and replicable solutions (i.e., off-the-shelf or easily adaptable devices) would be therefore desirable.

Recent advances in robotics and computer vision [20] have demonstrated that autonomous flights and real-time localization can be reliably implemented on small UAVs. On-board computing and real-time processing have also shown their potential for real-time mapping [21]. However, the replicability of these solutions in real scenarios is still limited because their limited flight autonomy and the high level of customization required for the used platforms. None of the existing solutions is devoted to the quick mapping of building damages to address the FRs' needs.

The development of deep learning algorithms for image classification [22,23] has shown its potential in the remote sensing domain [24–26] providing very promising results in the detection of building damages such as debris or rubble piles [14,27]. These works have demonstrated how the convolutional neural networks (CNN) outperform traditional classification approaches in terms of performances and transferability of the learnt features to new datasets. In these works, however, no specific attention is paid to timely results delivery. The main drawback of these techniques remains the extensive computational time required that is incompatible with FR's demands.

The present work tries to overcome the above-mentioned problems by proposing a solution based on a commercial UAV platform to autonomously map an area and detect building damages in near real-time. In particular, a binary map depicting the extension of the damages (i.e., delineation map) is delivered as output. The developed approach takes advantage of the most recent commercial UAV platforms that are delivered with software development kits (SDKs): these SDKs can customize the flight mode of the UAV and enable the real-time data streaming on a laptop. An efficient photogrammetric process is then implemented to generate a quick orthophoto of the area. A hybrid detection technique was then implemented combining in cascade morphological filters and vegetation indexes with a shallow convolutional neural networks (CNN). The first algorithms identify the areas of potential damages by selecting candidate areas in each image. Subsequently, the implemented CNN only classifies these regions instead of analyzing the whole image.

In the following sections a brief state of the art on the main elements of the developed solution (i.e., deep learning and real-time mapping) is given (Section 2). The developed methodology and

some of the performed tests are presented in Sections 3 and 4, respectively. The results obtained are discussed in Section 5 leading to the conclusions (Section 6).

2. Background

The presented approach builds on two main research topics: (i) real-time mapping and (ii) the automated building damage assessment from images. In the following, a non-exhaustive review of the state of the art in these two domains is given, focusing on the developments that are relevant for the approach presented in this paper.

2.1. Real-Time Mapping

This process refers to the generation of maps (i.e., georeferenced orthophoto) in near real-time. Different steps are followed in this process: image orientation, 3D point cloud generation, and orthorectification. The simultaneous image orientation and sparse point cloud generation of the surrounding environment is often called Visual SLAM (simultaneous localization and mapping) and it is assimilated to the structure from motion (SfM) for real-time applications [28]. Most of the developed approaches have been conceived for autonomous navigation or collision avoidance instead of mapping tasks [28,29]. When the reconstruction of the 3D space needs to be accurate like in UAV applications [30], then the use of bundle adjustments (BA) is adopted to reduce the deformations of large blocks [31]. Unfortunately, BA is a time-consuming procedure especially for large blocks. Different strategies have been implemented to make this process more compatible with real-time needs [27]. A relevant element is the use of local and global BA, acting on a small section or on the whole image block respectively [32–34]. The local adjustment requires limited time while the global adjustment is usually run only few times during the image acquisition process. All Visual SLAM approaches deliver sparse points even when videos and dense tracking algorithms are used [21]. Only very few recent solutions exploiting deep learning [35] are trying to deliver denser clouds in real-time. The generated point clouds are, therefore, too rough to be directly used in the orthorectification process. For that reason, the approaches for real-time orthophoto generation adopt strategies such as homographic assumptions [21] or planar projections [36,37] to ease and fasten the process. However, these methods are only suitable when mostly flat surfaces are surveyed (i.e., fields) and do not provide acceptable results for more complex scenes such as urban environments.

2.2. Automated Building Damage Assessment for First Responders

The automated detection of damages from images has been investigated for several years. Many recent approaches rely on both images and 3D points cloud to assess the damage state of a given area [14,16,38]. However, the generation of 3D information takes a considerable amount of time, thus limiting its use in FR procedures. Recent developments in machine learning (i.e., CNN) for image classification and segmentation have shown their potential in both computer vision [22] and remote sensing domains [24]. These approaches have already been applied to the image classification of building damages, considering data from different platforms and resolutions (satellite, aerial, UAV) [26]. The use of satellite and airborne images during training has also increased the reliability of the image classification of building damages from UAV imagery [27]. In most of these networks, damage cues—such as debris and rubble piles—are identified in a binary classification approach. Despite their promising results [14], little attention has been given to their real adoption in FR search and rescue activities. In particular, the CNN processing is computationally demanding and hardly compatible with real-time needs. In this regard, most recent solutions have already tried to limit the use of the most expensive processes to selected regions defined in advance with faster methods [15].

3. The Proposed Methodology—Materials and Methods

The developed method can be divided into three main components: (i) the app (installed on a smartphone) for customized flight mode and data streaming, (ii) the UAV flight planning toolbox,

and (iii) the incremental map generation and building damage detection (both installed on a laptop). The three components are embedded together in order to optimize the process and assure the timely processing of the data. As shown in Figure 1, the user initially defines an area of interest on the map and the software generates automatically a flight plan. The waypoints are then uploaded on the UAV that can start the image acquisition. During the flight the newly collected images are streamed on the laptop through the remote control and the customized app on the smartphone. The images are progressively processed in the laptop: the new images are sequentially added with a SfM algorithm and the damages are then detected in a classification process. When the image collection is completed, a true-orthophoto displaying the location of building damages is generated, and it is available by the time the UAV lands.

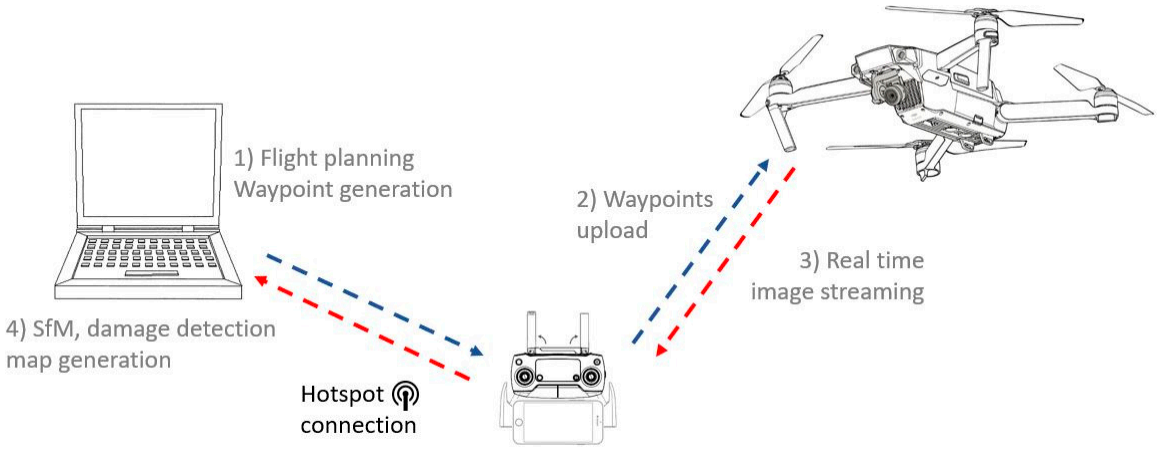
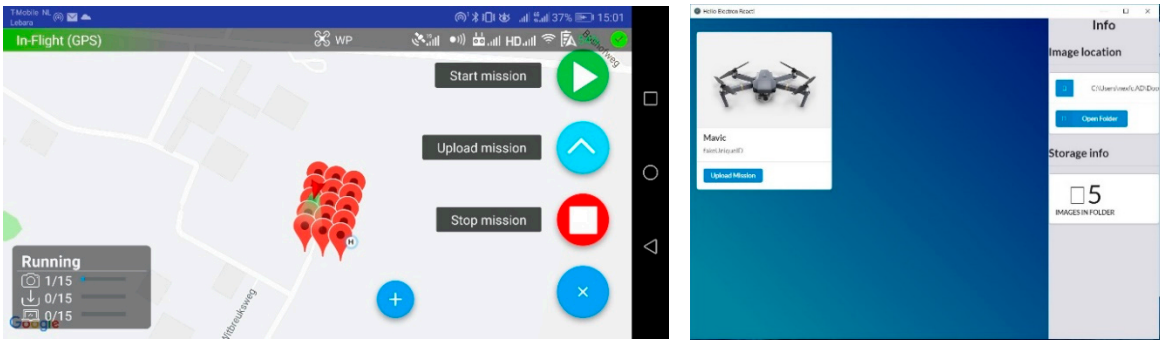


Figure 1. Overview of the developed system composed of a laptop, remote control, and smartphone as well as a UAV platform. Data are streamed from the laptop to the UAV platform (blue dashed line) and vice-versa (red dashed line). The tasks performed in each segment of the system (in grey) are reported according to their sequential order.

3.1. UAV Platform and Customized Flight Mode

Many UAV platforms are delivered with a dedicated SDK to customize their flight modes. In this paper, the adopted UAV platform is the Mavic Pro produced by DJI. This platform provides a reliable and fast connection with the remote control thanks to the Ocusync protocol that has higher communication speeds compared to other existing protocols (i.e., Lightbridge by DJI), as also confirmed by the performed tests. The Mavic Pro can also feature a relatively long flight time in excess of 20 min.

The communication between UAV platform and laptop is provided by an *Android* app installed on the mobile phone. This device is connected to the remote control via USB and to the laptop thanks to a wi-fi hotspot connection. Dedicated Javascript is installed on the laptop to allow the communication with the smartphone. The app has been developed using the mobile SDK of DJI and has two main functions: (i) upload the mission generated in the laptop on the drone, and (ii) receive the images from the platform and stream them to the laptop (see Figure 2a). The Javascript has a graphical interface that allows to verify the connection status with the app, to select the flight plan file to be uploaded and the folder for image storing on the laptop (see Figure 2b). This software interface communicates with the app through a websocket communication protocol, sending the flight plan to the app and receiving the streamed images during the flight. Each platform has a unique ID and the communication with more than one drone is therefore allowed. The UAV acquires the images in a stop and go mode. Although this acquisition mode is slower than in the continuous flight with regular internal shots, it gives higher flexibility that allows the change of camera pitch and yaw for every image.



(**a**) (**b**)

Figure 2. (**a**) The app installed on the smartphone: the mission can be uploaded, started, and stopped with the three buttons on the screen. The position of the waypoints and the UAV is provided in real-time during the mission. Other information such as the number of images that are collected, streamed on the phone or on the laptop is visualized on the bottom left window. (**b**) The connection between app (on the controller) uses the Javascript to upload the file from the laptop and select the folder where the images will be stored. The status of the connection is also given.

3.2. UAV Flight Planning

The goal of the UAV mission is to explore a new area and deliver a map for FR activity planning: no prior information such as the position of buildings or the presence of obstacles is given in advance. In the proposed implementation, the flight has therefore the aim of acquiring only nadir images while flying above obstacles. The flight planning software has been implemented to allow the automated generation of waypoints on an interest area. The user must define a polygon (of any shape) on Google Earth and the flight plan is automatically generated according to predefined overlaps, flight height, and UAV camera specifications. The flight plan parameters can be changed by the user, if needed.

The image location is planned in order to guarantee a minimum overlap (and number of images) in each point of the surveyed area. The position and the acquisition order are optimized in order to minimize the number of acquired images and, therefore, their acquisition time. To do that, the minimum bounding box of the initial polygon is initially determined. The flight strip orientation is set parallel to the longer side of this bounding box as image collection along the same strip is faster than across adjacent strips while the number of strips is also minimized. This strategy assumes locally flat areas to survey. The algorithm can optimize the flight plan for any shape of the area to be surveyed, as shown in Figure 3. The algorithm returns a .csv file with the waypoint coordinates of each image and the information on gimbal pitch (i.e., nadir view, 90°) and heading (i.e., equivalent to flight strip direction). This file is then uploaded from the laptop to the UAV using the Javascript and the smartphone app (see Figure 2).



(**a**) (**b**)

Figure 3. Example of flight plan on irregular shaped area (**a**). In this example the algorithm has been set to collect images with 80% forward and 60% side overlap for each point of the area flying at 50 m high (**b**).

3.3. Incremental 3D Map Generation and Building Damage Detection

The acquired images are streamed to the laptop during the flight and are iteratively processed to generate a building damage map of the area. As shown in Figure 4 two different process are integrated: (i) the incremental 3D map generation and (ii) the building damage detection.

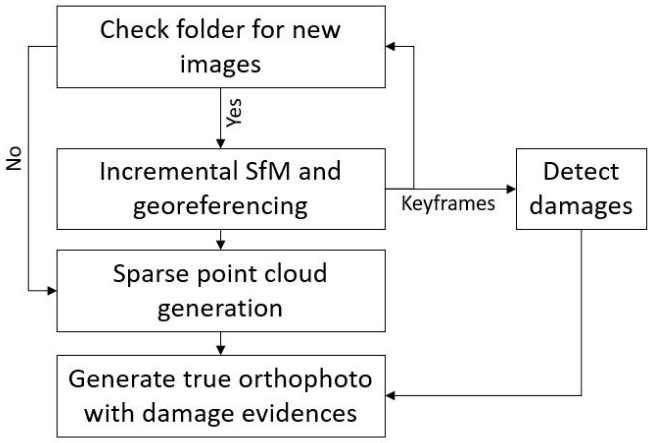


Figure 4. Workflow of the damage map generation.

The algorithm initializes the orientation of images after the first three images arrive on the laptop. Every new image received is added to the initial orientation iteratively enlarging the image block with a SfM process. This orientation process was implemented using the ColMap library [32] that takes advantage of CPU and GPU processing. This library can flexibly modify the parameters of the SfM and get fast and accurate 3D reconstructions of aerial image blocks. The original images (4000×3000 pixels) are downsampled to reduce the computation time needed for the tie-points extraction: ColMap implements a GPU enabled version of the SIFT algorithm. The image sequence concatenation is guided by the geotag coordinates of the images, reducing the search to the 10 closest images. The bundle block adjustment (BBA) is iteratively run to reduce deformations in the image block. A local BBA is run for every new image added to the block: only a few neighboring images are considered in this process. A global BBA is run after several images are added to the block. In the developed implementation, five images are considered as neighborhood of the local adjustment while the global BBA is run for every 10 new images.

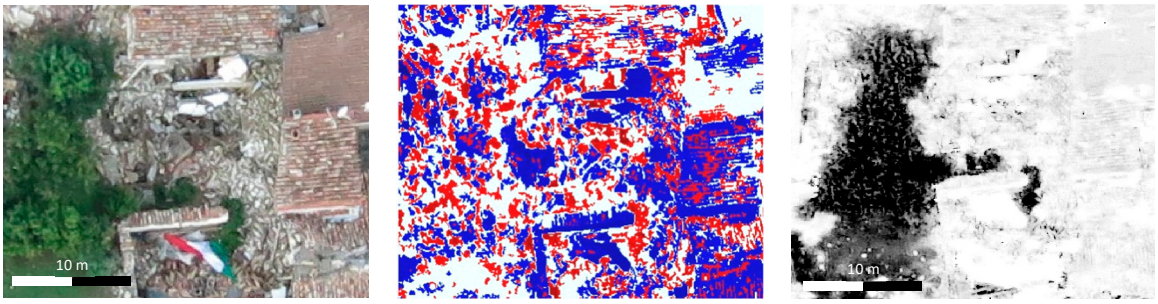
When the image stream stops, a final global BBA is run. The geotag of the images is then used to determine the absolute orientation of the block. The sparse point cloud generated by tie-points is finally used to generate a rough digital surface model (DSM) for the orthorectification process. Damaged areas are usually characterized by heterogeneously textured regions: many tie-points evenly distributed in the area are therefore extracted and used to generate these (quick) DSMs.

Some keyframes are regularly selected during the image collection to detect the presence of building damages (i.e., debris) as described in Section 3.3.1. The classified images are finally used as input in the orthophoto generation. The closest nadir image to each point of the DSM is used in the orthorectification process and a Z-buffer algorithm [39] is then used to avoid double mapping and occlusion problems. No radiometric correction is applied to blend the radiometric values among the different images as this significantly increases the computational time.

3.3.1. Damage Detection Algorithm

The direct use of CNN on the whole image would substantially increase the processing time, thus a two-step cascade approach was implemented. Debris, compared to intact regions, has normally a very fragmented and irregular structure. It is characterized by sudden radiometric variations between several small brighter and darker areas generated by the different illumination on the

fragmented surface. The rough selection of the areas potentially indicating damages can therefore be conducted using a morphological filter and the GRVI index [40]. The first filter excludes the areas corresponding to homogenously textured areas (typically in correspondence of intact regions) while the second eliminates the vegetated regions. The morphological filter is given by a sequence of image closing-by-reconstruction and opening-by-reconstruction as proposed in [41]. The first filter detects brighter areas compared to neighboring objects while the darker areas are detected with the second filter. Compared to other filtering methods, this approach defines a minimum and maximum size of darker and brighter areas that allows to select only relatively small regions (as in the case of debris). Given the approximate GSD of a UAV flight (typically 1–5 cm), it is possible to select regions on the ground with a size interval preserving their shapes in the images. Bright areas larger than 3 pixels and smaller than 25 pixels as well as dark regions smaller than 15 pixels are used to select candidate damages (Figure 5b). These sizes have been defined by experimental tests on the available test datasets. The image is then divided in patches (80 × 80 pixels) to assess the presence of damages. If a patch has a higher percentage of both brighter and darker pixels, this is considered a proxy for potential building damage areas. From experimental tests, 10% of both brighter and darker regions (extracted by morphological filters) is a precautionary value that allows to extract the damaged area without generating false negatives. These patches are further selected excluding those with large extends of GRVI values. The following CNN detection algorithm is run only on these patches to remove false positives, greatly reducing the time needed to classify the complete image.



(**a**) (**b**) (**c**)

Figure 5. Example of (**a**) debris in an image region. (**b**) Morphological filter detection: red and blue areas show darker and brighter regions detected by the morphological filter, respectively. An image patch is considered to correspond to debris if both dark and bright areas are close each other. (**c**) GRVI on the same region: darker pixels correspond to vegetated areas.

A shallow ResNet [22] network (see Figure 6) was developed to allow the fast and reliable detection of the damages. ResNet-like networks have shown very promising results in the damage detection [27] and shallow versions of this architecture have provided reliable results in complex tasks such as autonomous UAV flights [20]. In our application, we use a residual block that is formed as indicated in Figure 6 (RB—residual block) [22]. Six residual blocks are stacked with a low number of filters decreasing the number of parameters in the network (to reduce the overfitting) and also decreasing the time needed for both training and testing on new images. For the classification part of the network, global average pooling followed by a fully connected layer are used.

Building damage detection can be most reliably detected considering the contextual information of the patch: the developed network uses dilated convolutions and downsampled patches to take this into account (Figure 6). From experimental tests, the presented network has shown comparable results to deeper networks used for the same task while reducing the processing time: deeper and more complex networks [14,27] just marginally improved the validation accuracy at the cost of longer running processes.

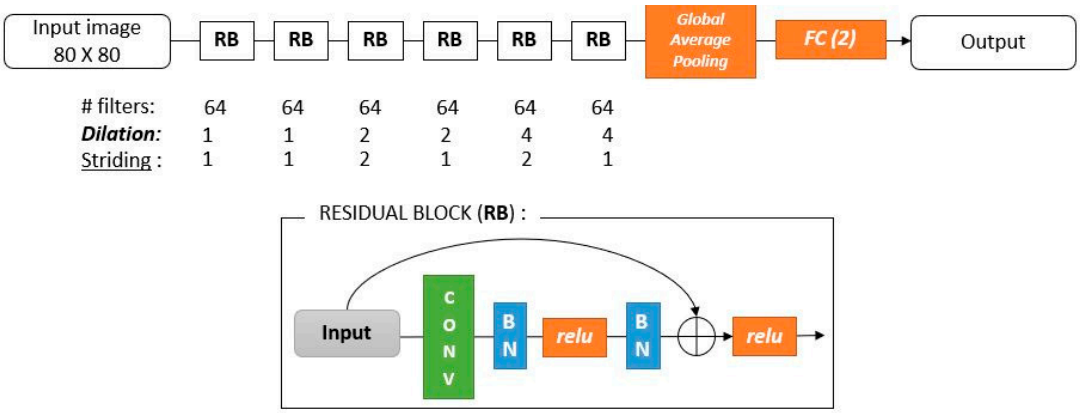
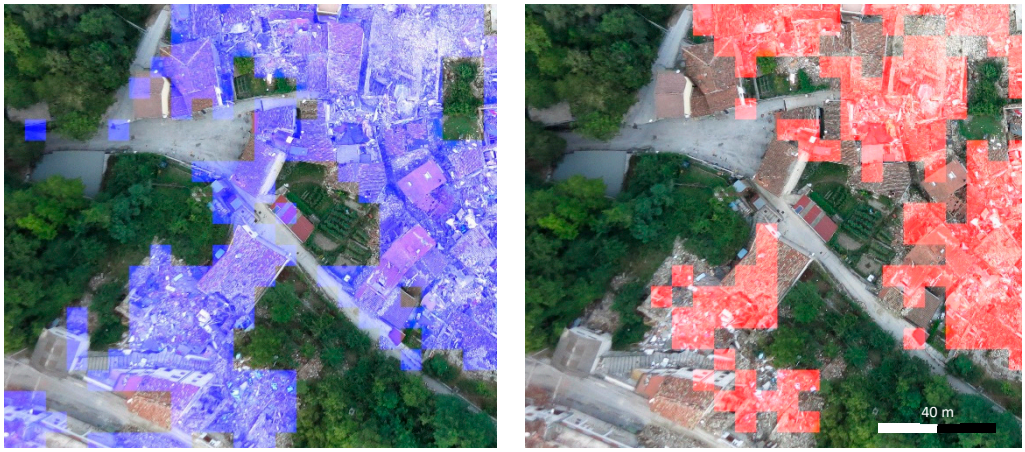


Figure 6. Architecture of the convolutional neural network implemented for the building damage detection. Each residual block is composed by convolutions (CONV), batch normalizations (BN), and rectified linear units (relu). The fully connected layer (FC) is used in the classification section.

The developed network was trained using the datasets presented in [27] for UAV images. Two classes (damaged and intact) have been considered, using about 4000 samples for each of them during the training. In order to prevent overfitting issues, training was extended using data augmentation (translation, rotation, and affine transformations). The training consisted of 100 epochs, with a batch size of 128 and 0.1 initial learning rate and a learning rate drop period of 40 epochs. The validation performed on 20% of the training samples reported an overall accuracy of 92.1%.

As shown in Figure 7, the network is able to classify the patches, reducing the number of false positives in the scene and delivering accurate detections in the images. The algorithm was implemented in MATLAB 2018b using the deep learning toolbox and taking advantage of the GPU and parallel processing.



(**a**) (**b**)

Figure 7. Example of regions detected by the preliminary morphological detection and GRVI (**a**, blue tiles) and the regions confirmed as damages by the CNN (**b**, red tiles).

4. Tests and Results

All components of the proposed approach have been tested in dedicated experiments. In this paper, only the two most relevant experiments are shown. These refer to two realistic scenarios executed during the pilot tests of the INACHUS project. In both cases the full pipeline was used to assess the performance of the methodology. Relatively small areas were surveyed because of legal constraints. The used CNN was trained before these tests using samples from different locations

(as explained above) to test its transferability to new datasets and therefore its practical usability in real scenarios.

The same PC was used for both tests: a Dell 5520 - i7 7820HQ (2.9GH) with 32 GB RAM and a Nvidia Quadro M1200 GPU graphic card. This allowed GPU to be fully exploited for the SfM (in the ColMap library) as well as for the MATLAB code implementing the morphological filters and the damage detection with CNN.

4.1. Lyon (France)

This dataset consists of 91 images that were acquired over a building demolition site (Lyon, France) specifically prepared for this pilot test. The flight height is 50 m while overlaps are 90% (forward) and 80% (side). The complete acquisition was performed in approximately 10 min (including take-off and landing), acquiring an image every 5.5 s on average. The data processing started after receiving the first images. Downsampled images (one-quarter the original size) were used in the orientation procedure, while a full resolution image was used for the damage identification. The damage detection was performed on every third image and this process took about 5 s for each image. In this regard the morphological filtering allowed the number of considered patches to be reduced by about 76% (on average only 397 patches out of 1900 per image).

The image orientation and the damage detection were completed about 50 s after the acquisition of the last image: this is still compatible with the near real-time goals as the UAV only took approximately 30 s to return from the final waypoint and land in its initial position. An additional 55 s were necessary to deliver the final orthophoto in GeoTIFF format to visualize them on external tools such as Google Earth as shown in Figure 8.



Figure 8. Final damage map over the test area in Lyon (France). Yellow and blue pictures show two examples of damages detected on the original images.

The orthophoto was rectified using the tie-points extracted in the image orientation as 3D. The generated DSM is therefore suboptimal and includes considerable distortions in correspondence of the edges of the image block where fewer tie-points are available to correctly reconstruct the scene in 3D. However, the used UAV does not have an accurate GNSS on board and systematic shifts (2–3 m in an easterly direction) are visible when comparing the orthophoto with Google Earth. However, map deformations and systematic errors in georeferencing are acceptable considering the specific purpose of the mapping. The detection algorithm was able to correctly localize the damages. The quantitative

assessment (Table 1) shows recall, precision and F1-score computed on the orthophoto (Figure 8): a manually delineated damage mask was used as ground truth. The achieved values confirm that the only morphological filter overestimates the percentage of true positives while CNN provides better results in all the considered metrics. In general, building damages are overestimated in correspondence of roofs and manmade objects that appear very similar to building debris in the images. Small false negative regions can be detected close to the building in the shadowed area: this misclassification can be explained by considering that these small regions are not visible in many images (due to the building occlusions) as well as by shadows and poor image quality in this part of the scene.

Table 1. Comparison of the results using the only morphological filter (MF) and adding in cascade the CNN (MF + CNN).

	MF	MF + CNN
Recall	0.821	0.831
Precision	0.573	0.839
F1-score	0.659	0.835

4.2. Roquebilliere (France)

This dataset is composed by 24 images acquired in a construction area formerly used as quarry. The flight was performed at 50 m height with 80% forward overlap and 60% side overlap. Part of the scene is occupied by a rubble pile on one side of the construction area (Figure 9).

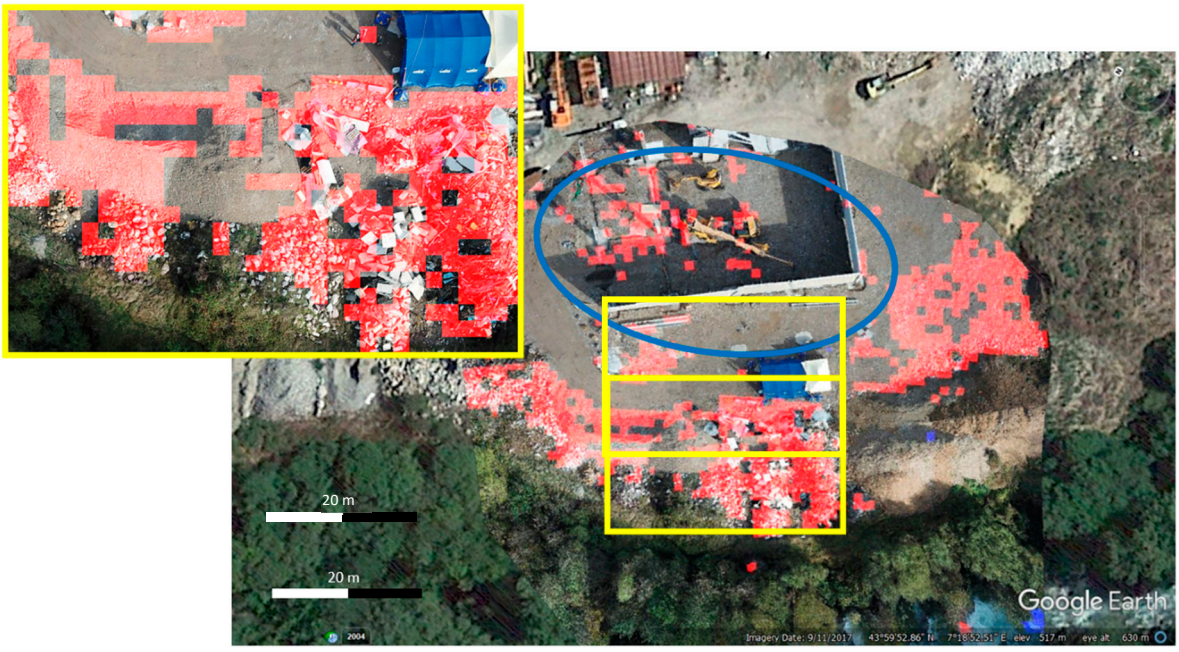


Figure 9. Final damage map over the test area in Roquebilliere (France). The yellow picture shows some examples of damages detected on the original images. The blue circle defines the construction site area.

A large percentage of the scene is also occupied by vegetation and bare soil. The flight was slower than the previous case (probably because of the wind) and images were acquired at 7 s intervals. The complete flight took 4:30 min.

The image orientation and damage detection were performed in real-time, while the generation of the orthophoto was completed few seconds before the UAV landing. In this case, the longer intervals and the lower number of images supported the completion of the work at reduced times.

The damage detection was performed regularly on every third image. As in the previous test, the morphological filters and the detection of vegetation reduced the number of patches processed by the neural network to 24%. The 81% of these patches were then classified as damages. The generated orthophoto has some small shifts compared to the Google Earth map, as discussed in the previous case. Given the difficulty in defining a univocal ground truth (because of the presence of many objects mimicking the debris), no quantitative assessment has been performed in this test. Qualitatively, the number of false positives is relatively high: the algorithm was trained (with real case images) to detect concrete and brick debris that mimic rubble piles made of stones and soil largely present in this scene. Some additional false positives are then visible inside the building under construction. No false negatives are visible in the delivered results.

5. Discussion

The performed tests have shown the suitability of the presented approach for near real-time damage mapping. The stop & go flight mode implemented in the app reduces the productivity of the flight: between 8 and 12 images per minute were normally acquired in the performed tests, depending on the distance among them and the weather conditions. The stop & go however allows a better quality of the images (i.e., no motion blur) and gives a higher flexibility compared to acquisitions with fixed time intervals as the gimbal can be rotated for each image independently. The delay between image acquisition and storing on the laptop is normally less than 2 s. The performed tests showed that the streaming is faster and more reliable if no obstacles are present between the remote control and the UAV.

The approach can execute the orientation of the images and the simultaneous damage detection in real-time for small blocks, as was the case in the second test. Large image blocks (as in the Lyon test) usually determine delays in the processing of the images: this is mainly due to the global BBA in the image orientation as a larger number of observations progressively needs to be processed, requiring more time. How often to perform the global adjustment in the image block is one of the main aspects to be considered to reduce the processing time. Many implementations in the robotics domain only perform local BBA (or similar approaches) to make the orientation progress more efficient, often at the cost of larger block deformations. In the presented application, the geometric consistency of the generated map still represents a priority that cannot be neglected for a slightly faster implementation. Other elements such as the resolution of the images used in the orientation or the performance of the used laptop can also strongly influence the real-time feasibility of the approach.

The use of deep learning algorithms outperforms traditional detection algorithms but requires higher processing times. The adoption of a morphological filter coupled with a vegetation index has greatly reduced the search for building damages in the image, reducing the tiles processed by the neural network. The parameters of the morphological filter are intentionally relaxed to prevent the generation of false negatives. As shown in Section 4.1, few damages can be classified as intact regions (i.e., false negatives), but this problem can be explained considering the presence of shadows, the building occlusions in many images and the poor image quality. It must be noted that the damage detection on these regions fails using both the morphological filter and the CNN. False positives are more advisable than false negatives from a FR's perspective. However, the performed tests (also using other datasets) have demonstrated that this kind of error is limited to very small areas and peculiar conditions (i.e., shadowed areas, blurred images, etc.). In general, the implemented CNN has allowed to generate reliable and complete detections despite its (relatively) limited number of layers. Damages are generally overestimated as many objects look very similar to damaged structures.

6. Conclusions and Future Developments

This paper has demonstrated that building damage maps can be generated in near real-time using low-cost and commercial UAVs. The solution was conceived to give an answer to the practical needs of FR involved in urban search and rescue operations: FR involved in the INACHUS project have

appreciated the performance of this solution. The presented methodology takes advantage of different solutions coming from different domains (robotics, photogrammetry, and computer vision) to deliver an almost completely autonomous solution. The customizations performed with the SDK allowed to stream the images on a dedicated laptop while the integration of photogrammetry and deep learning algorithms allowed the efficient generation of building damage maps.

All the implemented components of the methodology were thoughtfully tested although only two sets of results on the performed pilot tests were presented in this paper. These tests showed that the communication between UAV and remote control as well as the following data streams are generally reliable. However, the use of high-end smartphones, with better hotspot capability and processing speed is preferable to fasten the streaming. As already discussed, a stop & go flight mode is more flexible but less efficient than flights with regular interval acquisitions: for this reason, the app will be further developed to introduce this option only for nadir flights. On the other hand, the presented approach will also be extended to façade inspection planning.

The developed data processing showed real-time capabilities only for small datasets. Larger datasets can be processed with increasing delays in the orientation procedure. Although the registered delays are still acceptable for the specific task, more effective solutions could be tested to further improve this feature.

The two-step approach building damage detection has allowed to achieve reliable results in acceptable times, overcoming the need of more expensive hardware. The full resolution images (12MP in the performed tests) can be efficiently processed in a few seconds. The presented network is largely a state-of-the-art network customized for building damage assessment. Although its performance is comparable with existing deeper networks [14,27], additional studies will be undertaken to test different architecture and improve the current results. The migration from MATLAB to more efficient programming languages could further fasten the process, although recent developments of this software have returned comparable processing times to other tools and libraries [42].

Author Contributions: F.N. conceived the technical solution, developed the SfM and drafted the paper. F.N. and D.D. developed and tested the damage assessment algorithm. A.S. developed the app. F.N., D.D. and N.K. improved the methodology and the paper structure.

Funding: This work was funded by INACHUS (Technological and Methodological Solutions for Integrated Wide Area Situation Awareness and Survivor Localization to Support Search and Rescue Teams), an EU-FP7 project with grant number 607522.

Acknowledgments: We thank the anonymous reviewers for their insights and constructive comments, which helped to improve the paper.

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

References

1. Dell'Acqua, F.; Gamba, P. Remote sensing and earthquake damage assessment: Experiences, limits, and perspectives. *Proc. IEEE* **2012**, *100*, 2876–2890. [CrossRef]
2. Eguchi, R.T.; Huyck, C.K.; Ghosh, S.; Adams, B.J.; McMillan, A. Utilizing new technologies in managing hazards and disasters. In *Geospatial Techniques in Urban Hazard and Disaster Analysis*; Showalter, P.S., Lu, Y., Eds.; Springer: Dordrecht, The Netherlands, 2009; pp. 295–323. ISBN 978-90-481-2237-0.
3. United Nations INSARAG Guidelines. Available online: https://bit.ly/2HHLcoX (accessed on 31 December 2018).
4. Curtis, A.; Fagan, W.F. Capturing damage assessment with a spatial video: An example of a building and street-scale analysis of tornado-related mortality in Joplin, Missouri, 2011. *Ann. Assoc. Am. Geogr.* **2013**, *103*, 1522–1538. [CrossRef]
5. Ishii, M.; Goto, T.; Sugiyama, T.; Saji, H.; Abe, K. Detection of earthquake damaged areas from aerial photographs by using color and edge information. In Proceedings of the 5th Asian Conference on Computer Vision, Melbourne, Australia, 22–25 January 2002.
6. Vu, T.T.; Matsuoka, M.; Yamazaki, F. Detection and animation of damage using very high-resolution satellite data following the 2003 Bam, Iran, earthquake. *Earthq. Spectra* **2005**, *21*, 319–327. [CrossRef]

7. Balz, T.; Liao, M. Building-damage detection using post-seismic high-resolution SAR satellite data. *Int. J. Remote Sens.* **2010**, *31*, 3369–3391. [CrossRef]
8. Brunner, D.; Schulz, K.; Brehm, T. Building damage assessment in decimeter resolution SAR imagery: A future perspective. In Proceedings of the Joint Urban Remote Sensing Event, Munich, Germany, 11–13 April 2011; pp. 217–220.
9. Armesto-González, J.; Riveiro-Rodríguez, B.; González-Aguilera, D.; Rivas-Brea, M.T. Terrestrial laser scanning intensity data applied to damage detection for historical buildings. *J. Archaeol. Sci.* **2010**, *37*, 3037–3047. [CrossRef]
10. Khoshelham, K.; Oude Elberink, S.; Xu, S. Segment-based classification of damaged building roofs in aerial laser scanning data. *IEEE Geosci. Remote Sens. Lett.* **2013**, *10*, 1258–1262. [CrossRef]
11. Gerke, M.; Kerle, N. Automatic structural seismic damage assessment with airborne oblique Pictometry© imagery. *Photogramm. Eng. Remote Sens.* **2011**, *77*, 885–898. [CrossRef]
12. Copernicus Emergency Management Service. Available online: https://bit.ly/2Giqg5H (accessed on 30 December 2018).
13. Kerle, N. Satellite-based damage mapping following the 2006 Indonesia earthquake—How accurate was it? *Int. J. Appl. Earth Obs. Geoinf.* **2010**, *12*, 466–476. [CrossRef]
14. Vetrivel, A.; Gerke, M.; Kerle, N.; Nex, F.; Vosselman, G. Disaster damage detection through synergistic use of deep learning and 3D point cloud features derived from very high resolution oblique aerial images, and multiple-kernel-learning. *ISPRS J. Photogramm. Remote Sens.* **2018**, *140*, 45–59. [CrossRef]
15. Duarte, D.; Nex, F.; Kerle, N.; Vosselman, G. Towards a more efficient detection of earthquake induced facade damages using oblique UAV imagery. In Proceedings of the ISPRS—International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Bonn, Germany, 4–7 September 2017; Volume XLII-2/W6, pp. 93–100.
16. Rupnik, E.; Nex, F.; Toschi, I.; Remondino, F. Contextual classification using photometry and elevation data for damage detection after an earthquake event. *Eur. J. Remote Sens.* **2018**, *51*, 543–557. [CrossRef]
17. Xu, Z.; Yang, J.; Peng, C.; Wu, Y.; Jiang, X.; Li, R.; Zheng, Y.; Gao, Y.; Liu, S.; Tian, B. Development of an UAS for post-earthquake disaster surveying and its application in Ms7.0 Lushan Earthquake, Sichuan, China. *Comput. Geosci.* **2014**, *68*, 22–30. [CrossRef]
18. Boccardo, P.; Chiabrando, F.; Dutto, F.; Tonolo, F.; Lingua, A. UAV deployment exercise for mapping purposes: evaluation of emergency response applications. *Sensors* **2015**, *15*, 15717–15737. [CrossRef] [PubMed]
19. INACHUS Project. Available online: www.inachus.eu (accessed on 31 December 2018).
20. Palossi, D.; Loquercio, A.; Conti, F.; Flamand, E.; Scaramuzza, D.; Benini, L. Ultra low power deep-learning-powered autonomous nano drones. *arXiv* **2018**, arXiv:1805.01831.
21. Hinzmann, T.; Schönberger, J.L.; Pollefeys, M.; Siegwart, R. Mapping on the fly: Real-time 3D dense reconstruction, digital surface map and incremental orthomosaic generation for unmanned aerial vehicles. In *Field and Service Robotics*; Hutter, M., Siegwart, R., Eds.; Springer International Publishing: Cham, Switzerland, 2018; Volume 5, pp. 383–396. ISBN 978-3-319-67360-8.
22. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the CVPR, Las Vegas, NV, USA, 26 June–1 July 2016.
23. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. In Proceedings of the ICLR, San Diego, CA, USA, 7–9 May 2015; pp. 1–13.
24. Maggiori, E.; Tarabalka, Y.; Charpiat, G.; Alliez, P. Convolutional neural networks for large-scale remote-sensing image classification. *IEEE Trans. Geosci. Remote Sens.* **2017**, *55*, 645–657. [CrossRef]
25. Fu, G.; Liu, C.; Zhou, R.; Sun, T.; Zhang, Q. Classification for high resolution remote sensing imagery using a fully convolutional network. *Remote Sens.* **2017**, *9*, 498. [CrossRef]
26. Wei, Y.; Wang, Z.; Xu, M. Road structure refined CNN for road extraction in aerial images. *IEEE Geosci. Remote Sens. Lett.* **2017**, *14*, 709–713. [CrossRef]
27. Duarte, D.; Nex, F.; Kerle, N.; Vosselman, G. Multi-resolution feature fusion for image classification of building damages with convolutional neural networks. *Remote Sens.* **2018**, *10*, 1636. [CrossRef]
28. Taketomi, T.; Uchiyama, H.; Ikeda, S. Visual SLAM algorithms: A survey from 2010 to 2016. *IPSJ Trans. Comput. Vis. Appl.* **2017**, *9*, 16. [CrossRef]

29. Cadena, C.; Carlone, L.; Carrillo, H.; Latif, Y.; Scaramuzza, D.; Neira, J.; Reid, I.; Leonard, J.J. Past, present, and future of simultaneous localization and mapping: Towards the robust-perception age. *IEEE Trans. Robot.* **2016**, *32*, 1309–1332. [CrossRef]
30. Lu, Y.; Xue, Z.; Xia, G.-S.; Zhang, L. A survey on vision-based UAV navigation. *Geo-Spat. Inf. Sci.* **2018**, *21*, 21–32. [CrossRef]
31. Strasdat, H.; Montiel, J.M.M.; Davison, A.J. Visual SLAM: Why filter? *Image Vis. Comput.* **2012**, *30*, 65–77. [CrossRef]
32. Schonberger, J.L.; Frahm, J.-M. Structure-from-Motion revisited. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 26 June 26–1 July 2016; pp. 4104–4113.
33. Klein, G.; Murray, D. Parallel tracking and mapping for small AR workspaces. In Proceedings of the 2007 6th IEEE and ACM International Symposium on Mixed and Augmented Reality, Nara, Japan, 13–16 November 2007; pp. 1–10.
34. Wu, C. Towards linear-time incremental structure from motion. In Proceedings of the 2013 International Conference on 3D Vision—3DV 2013, Seattle, WA, USA, 29 June–1 July 2013; p. 8.
35. Kendall, A.; Martirosyan, H.; Dasgupta, S.; Henry, P.; Kennedy, R.; Bachrach, A.; Bry, A. End-to-end learning of geometry and context for deep stereo regression. *arXiv* **2017**, arXiv:1703.04309.
36. DroneDeploy. Available online: https://bit.ly/2S2wTzQ (accessed on 31 December 2018).
37. LiveDroneMap. Available online: http://www.gaia3d.com/en/?p=2304 (accessed on 31 December 2018).
38. Cusicanqui, J.; Kerle, N.; Nex, F. Usability of aerial video footage for 3D-scene reconstruction and structural damage assessment. *Nat. Hazards Earth Syst. Sci. Discuss.* **2018**, *18*, 1583. [CrossRef]
39. Habib, A.F.; Kim, E.-M.; Kim, C.-J. New methodologies for true orthophoto generation. *Photogramm. Eng. Remote Sens.* **2007**, *73*, 25–36. [CrossRef]
40. Motohka, T.; Nasahara, K.N.; Oguma, H.; Tsuchida, S. Applicability of green-red vegetation index for remote sensing of vegetation phenology. *Remote Sens.* **2010**, *2*, 2369–2387. [CrossRef]
41. Dalla Mura, M.; Benediktsson, J.A.; Waske, B.; Bruzzone, L. Morphological attribute profiles for the analysis of very high resolution images. *IEEE Trans. Geosci. Remote Sens.* **2010**, *48*, 3747–3762. [CrossRef]
42. Matworks & Deep Learning. Available online: https://bit.ly/2Wy3T1K (accessed on 31 December 2018).



© 2019 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (http://creativecommons.org/licenses/by/4.0/).