Building Change Detection in Aerial Images

Fatima Mroueh
Computer Science Department
Lebanese University
Beirut, Lebanon
fatima.mroueh249@gmail.com

Ihab Sbeity
Computer Science Department
Lebanese University
Beirut, Lebanon
ihab.sbeity@gmail.com

Mohamad Chaitou
Computer Science Department
Lebanese University
Beirut, Lebanon
mohamad.chaitou@ul.edu.lb

***Abstract*—In this paper, we provide an approach that detects the changes in buildings between two multi-temporal aerial images of different sources. Since the images in most cases are not perfectly aligned, our approach takes into consideration the differences in the geometric aspects of the images. Differences in scale, view point or overlapping regions may be present between the pair of images. Our approach relies on segmentation to extract building masks from the original aerial images. Changes are then found by comparing the features of the pair of masks using image matching algorithms. This procedure is applied on a set of 80 pairs of aerial images of different sizes and with different applied transformations, and an evaluation has been considered in comparison with the corresponding ground truth references. The evaluation yields buildings change detection rate of 92.7%. The results of our proposed approach suggest that automatic building change detection is possible, but further research should include improvement of the segmentation phase to better distinguish buildings and enhancement of the change detection method. Real time application of the process is also a challenging perspective.**

***Index Terms*—change detection, aerial images, image segmentation, image matching algorithms, SIFT, feature detection, feature description**

I. Introduction

Aerial imagery is - as it sounds - the process of taking images from the air. It is a subset of a larger domain called Remote Sensing. It consists of acquiring data without making physical contact with the objects in study [1].

Aerial images such as satellite imagery or drone imagery are considered one of the richest sources of data that can be used in various applications. Change detection in aerial images is detecting new or disappeared objects in images registered at different moments of time and possibly in various lighting, heights and camera calibrations [2]. Detecting the changes in aerial images of the same region and taken at different times is useful and important in many domains such as: automatic map updating, field change after catastrophic events, detecting illegal buildings areas and undeclared refugees camps, analysis of urban and suburban areas, a base for automatic monitoring system and some other military applications. For these reasons, detecting changes in aerial images has thus become an important research topic [3].

In fact, several techniques and approaches are designed and implemented to detect changes in aerial images. However, all these techniques were motivated by the availability and fusion of different types of useful and profitable remote sensing data, such as data generated from Digital Elevation Models (DEM), Light Detection and Ranging (LiDAR) technology and other kinds of remote sensing technologies [1] [4], or limited to work with specific types of images such as GeoTiff images that contain accurate geographic information such as coordinates is the global coordinates system. Furthermore, they are also limited to aligned images that are of the same scale and view point (same height, same camera calibrations, same coordinates...).

The main problem with these techniques is that they rely too much on the information provided with the images, and therefore they cannot be applied to any image that is not enriched with any information such as geo-spatial information.

Nowadays, the automatic analysis techniques of images are very essential. Machine Learning and Computer Vision techniques, and more specifically the image matching algorithms, has proven to be very efficient in the field of image processing and comparison. Furthermore, there are still poor scientific methodologies for detecting changes in aerial images, especially those that differ in geometric aspects such as scale and orientation, and without being limited to additional information about the images. Going deeper into the topic is essential to introduce new efficient insights in the field of change detection in aerial images.

Accordingly, this research provides a complete procedure for building change detection in aerial images using machine learning and computer vision techniques and algorithms.

The main advantage of our approach is that it does not take benefit from any of the information derived from the aerial images. It deals with aerial images as simple PNG or JPG formats without any enrichment. More importantly, it can detect changes in aerial images that differ in scale and view point and images that have overlapped regions. This way, our approach can be applied to any pair of aerial images despite of their related information or their geometric aspects.

II. Previous Studies

Detecting the changes in aerial images has been an old and long journey. In particular, changes in buildings, is an essential part of this journey.

Looking at the previous studies related to our topic, one can see that most of these studies rely on data fusion; they integrate multiple data sources to produce more consistent and accurate information than that provided by any individual data source.

For example, in their work, Nebiker et al. used image-based dense digital surface models (DSMs) in order to compute a depth value for every pixel of an image, combined with the aerial images for the detection of individual buildings. They used these models with object-based image analysis to detect changes [5]. As well, the study of Chen Lin was based on multi-source data. They pre-processed the data using triangulation of an irregular network of data points collected by Light Detection And Ranging (LiDAR) technology, and then, the changes were detected by finding differences in height by comparing the LiDAR point measurements and the estimates of the building models [6]. Furthermore, Alonso et al. applied the support vector machine (SVM) classification algorithm to a joint satellite and laser data set for the extraction of buildings. For change detection, they suggested to compare an old map with more recent spatial information instead of comparing a pair of images [7]. Many other studies took benefit from data sources other than the aerial image itself such as Digital Elevation Models (DEMs), laser scanner data, indicator of vegetation (NDVI), the relationship between the buildings and their shadows, and high resolution aerial images in order to detect changes in buildings [8] [9] [10] [11] [12]. Most of these studies suffered from significant problems with small buildings and with buildings surrounded by high trees.

Talking about extracting the buildings before detecting changes, this step was included in numerous studies. Some of them went for region-based classification where each small region was classified to “building” or “no-building” based on a decision tree induced from training data (edge recording of the buildings), and then classified to “change” or “no change” based on some conditions [13]. Other used the indicator of vegetation NDVI for distinguishing buildings from trees since both have similar height information [14]. Neural network classifier also was employed in order to classify the regions in an aerial image into multiple classes (grove, building, tree, shadow ...) by feeding the neural network with many inputs such as area, average gray level, shape factor and compactness [3]. Region-based segmentation was also applied using a decision tree that rely on the geometric properties of the land cover objects such as elevation, spectral information, texture and shape [15]. The most important and precise segmentation was applied using Convolutional Neural Networks where the large imagery is divided into small patches, and then CNN is trained with those patches and their corresponding three-channel map patches (building, road and background) [16]. However this work was not including change detection.

As for detecting changes in aerial images that have different views, Bourdis et al. stated that camera motion and viewpoint differences introduce parallax effects. Therefore, in order to be robust to viewpoint differences, they introduce an algorithm to distinguish between real changes and parallax effects based on optical flow constrained with epipolar geometry [17]. In other works concerning this point, knowing the calibration of the camera or the spatial information about the geographic area were essential in order to achieve the goal [13] [17].

Furthermore, ArcGIS Pro offers a tool that detect feature changes by finding where the update line features spatially match the base line features and detects spatial changes, attribute changes, or both, as well as no change. However, all inputs to this tool must be in the same coordinates system [18]. While in our case, we aim to detect changes even if we do not know the spatial location of the geographic region we are working on.

To the best of our knowledge, we did not find any study that processes aerial images independently from any other source of information to extract buildings from. Moreover, computer vision techniques such as image matching algorithms are not employed to detect changes although they proved to be very efficient in the comparison of images.

To overcome the two problems cited above, our approach works in three steps. As we are interested in small-scale change detection (buildings), the first step is the segmentation phase, in which we eliminate a large part of the scene without losing any actual building. This is possible by extracting buildings’ footprints from the aerial images. Second, we use the SIFT image matching algorithm to check the correspondence of the pair of images, i.e. to make sure that the images taken correspond to the same geographic region. Third, we detect the type of transformation applied to one of the images with respect to the other (scale, rotation, overlap). The detected transformation is then reversed to get two images of the same scale and view. The last step, the difference image can be computed and post-processed. In this step, the changes in the buildings are detected.

III. Backgroud

A. Image Segmentation

Computer vision is a field that is intended to make computers accurately understand and efciently process visual data like images. Extracting information from images and understanding image information is very critical in many applications in this domain. Computer vision helps in extracting features of an image in order to simplify image analysis [19].

In several cases, we may not be interested in all the components of the image, but only for some areas or objects that have certain characteristics related to our task. Image segmentation is one of the best techniques to handle this issue. This technique works by isolating objects from the rest of the image [20] [21] [22] [23] [24]. Image segmentation mainly has the role of classifying each pixel of an image into meaningful classes that refer to specific objects. It involves grouping of the elements of an image by certain criteria of homogeneity [4]. It does not only make a prediction for an input providing classes, but also provides additional information regarding the location of such classes.

Deep learning techniques have proven to be very efficient in solving such problems. These techniques can learn patterns in order to predict classes. The main deep learning architecture that is used for image segmentation, and generally speaking for image processing, is the Convolutional Neural Network (CNN).

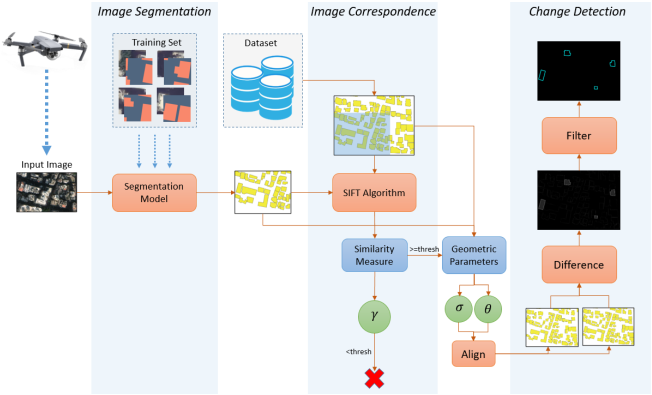

Fig. 1. Worflow of our approach.

Frameworks like MaskRCNN, RetinaNet allow to apply image segmentation using deep learning. However the domain of application of some of them is restricted to scene images, and they cannot be used in case of aerial images [25] [26]. Other frameworks that work with aerial images such as ENVI, ERDAS Imagine, eCognition and others are also available [27] [28]. Nevertheless they have many limitations. Some of them do not have any vectorization tool to convert the segmented result to use them in further analysis, others are confused with images where the building roofs were dark and having intensities very less as compared to other building objects [29].

B. *Image Matching*

In order to compare the images, we look for specific patterns or specific features that are unique in the images and that can be easily compared. A feature is a relevant piece of information. It is a specific structure in the image such as a point, an edge or a corner. The operation of finding the features of an image is called Feature Detection.

Feature detection is the process of transforming the visual information of the image into the vector space. It is basically finding keypoints (or interest points) in the image. A keypoint is a unique point in the local area around it. A keypoint can be matched to a corresponding point in another image. The main purpose of detecting features is giving us the possibility to perform mathematical operations on them, and thus to find similar vectors that lead us to similar objects or scenes in the images. Ideally, this information should be invariant under image transformations, so we can find the same features again even if the image is transformed in some way.

Using a specific feature detection algorithm, we search for such features in the first image and then we look for the same features in the other image. As a result, we get a set of points (x_i, y_i) for each image, where x_i and y_i are the coordinates of the point i detected as a feature in the image. After detecting interest points, we continue to compute a descriptor for each one of them. The regions around the features should be described so that the algorithm can find the similar features in the other image. This is called the Feature Description.

The local appearance around each feature point is described in some way that is invariant under changes in translation, scale and rotation. Therefore, we end up with a descriptor vector for each feature point. Feature descriptors encode interesting information into a series of numbers and act as a sort of numerical ‘fingerprint’ that can be used to differentiate one image from another. Once the features and the descriptors are extracted and computed, some preliminary feature matches between these images will be established.

Feature matching, or more generally image matching, is the task of establishing correspondences between two images. Keypoints between two images are matched by identifying their nearest neighbors. This is achieved by comparing the descriptors across the images to identify similar features. For any two images, we get a set of pairs $(x_i, y_i), (x'_i, y'_i)$ where (x_i, y_i) is a feature in one image and (x'_i, y'_i) is its matching feature in the other image. We can summarize the process of image matching as follows: 1- Find a set of distinctive keypoints. 2- Define a region around each keypoint. 3- Extract and normalize the region content. 4- Compute a local descriptor from the normalized region. 5- Match local descriptors.

Many comparative studies have been published assessing the performance of the image matching algorithms. The real challenge is to achieve true invariant feature detection under any image transformation. It seems that the selection of the adequate algorithm to complete the matching task significantly depends on the type of the image to be matched and in the variations within an image and its matching pair in scale, orientation or other transformations. Most of these studies has stated that Scale Invariant Feature Transform SIFT algorithm performs the best against different image transformations [30] [31] [32] [33].

IV. Methodolody

Fig. 1 represents the overall process of our approach: First, the buildings footprints from the acquired aerial image will be extracted in order to use them in detecting changes instead of the original aerial images. To achieve this step, a segmentation model for extracting buildings masks from aerial images is built. Second, we suppose that a database is already prepared containing preprocessed aerial images’ masks of the region of interest. In this step, we look into in the database for the mask that corresponds to our input mask. This step is achieved by computing a similarity measure between each couple of images using SIFT image matching algorithm. Finally, after aligning the couple of masks, we detect changes by filtering their difference image.

A. *Buildings’ Footprints Extraction*

Extracting buildings’ footprints from aerial images is a kind of preprocessing of the images before matching. It helps us to get better results in detecting changes, since by segmenting the images, we get rid of every element that is considered noise for us (not an object of interest).

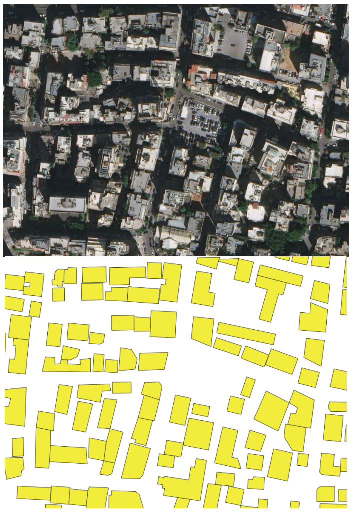

Fig. 2. Aerial image and its corresponding buildings mask.

A segmentation model is needed for this purpose. Many tools that implement this techniques are available. The used tool to achieve our goal is RoboSat [34]. RoboSat is an end-to-end pipeline written in Python3 for feature extraction from aerial and satellite imagery. Features can be anything visually distinguishable in the imagery such as buildings, roads or cars [34]. We chose to work with RoboSat since it is specially designed to work with aerial images and it has shown important results in this domain.

The data preparation tools in RoboSat help us to create and prepare the dataset for training feature extraction models. Also, the modelling tools in RoboSat help with training fully convolutional neural networks for segmentation [34].

Fig. 2 represents an aerial image with its corresponding buildings mask.

a) Data Preparation: We first walk through creating a dataset for training feature extraction model. Such dataset consists of satellite imagery combined with their corresponding masks for the feature we want to extract, which is building in our case. We can think of these masks as binary images which take the value zero where there is no building and one for building areas.

This dataset will serve as training set for the segmentation model. The goal is to have a model that accepts an aerial image and outputs its corresponding buildings footprints. As mentioned before, the footprints will be used to detect changes instead of original aerial images, and this is to reduce all kinds of noise that may affect the accuracy of our application. Our objects of interest are only buildings.

We start by extracting geometries from OpenStreetMap (OSM) project. We try then to figure out where we need satellite imagery in order to complete the training set [35]. OpenStreetMap (OSM) project creates and provides free geographic data. The OpenStreetMap Foundation is an international not-to-profit organization supporting the OpenStreetMap project. This project maintains data about roads, buildings, trails, railway stations and much more, all over the world. OSM maps are saved on the internet and they are totally free. But the most important thing is that OSM is accurate and up to date (normally updated every day) [35].

There are two reasons for which we are building our own segmentation model instead of using OSM data directly. The first reason is that OSM data do not cover all the regions we are interested in. In Lebanon for example, buildings masks are not provided for all the country. So, we take benefit from the available geometries provided by OSM in order to build the segmentation model. Later, this model will provide us with the buildings footprints for regions that are not covered by OSM. The second reason, which is the most important one, is that we may not be aware of the exact location of the image in the global coordinates system. In such case, we cannot use OSM extracts.

GeoFabrik server from OSM provides a convenient and updated extracts which we can work with [36]. GeoFabrik team extract, select and process free geodata for everyone. They create shape files, maps, and map tiles with a free of charge download service. The geometries extracted from GeoFabrik server are shape files of extension .shp. A shape file is a simple format that is used for storing the geometric location and attribute information of geographic features that can be represented by points, lines or polygons. We are only interested in polygon representation of the buildings. These shape files can be visualized as vector layer in GIS tools, which help us to decide at what locations we need to download satellite imagery to complete the dataset.

Although the masks are not always perfect, but a slightly noisy dataset will still work fine with training the model on thousands of images and masks.

The next step is to download the corresponding aerial imagery. Our aerial imagery is downloaded from Mapbox [37]. Mapbox satellite is a full global base map. It uses global satellite and aerial imagery from commercial providers such as NASA and USAS. Mapbox provides an API that allows us to download the needed satellite imagery [37].

RoboSat works with the Slippy Map tile format to abstract away georeferenced imagery behind tiles of the same size. A Slippy Map is, in general, a term referring to modern web maps which let you zoom and pan around. By default, the Slippy Map renders tiles. Tiles are 256 x 256 pixel PNG files. Each tile is a file in a directory representing a column, and each column is a subdirectory that represents the zoom level. RoboSat offers the tool that is responsible for tiling the collected aerial images as well as extracted geometries.

With downloaded satellite imagery and rasterized corresponding masks, our dataset is complete and ready. Fig.3 shows the downloaded aerial imagery tiles with their corresponding buildings footprints.

b) Training and Modelling: The RoboSat segmentation model is a kind of fully convolutional neural network which we train on pairs of aerial images and corresponding masks. The training process takes place within a GeForce GTX 1080 platform. When picking up the best checkpoint, the model

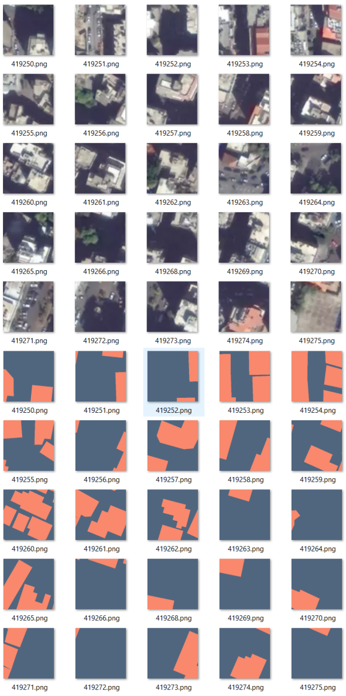

Fig. 3. Aerial images tiles with their corresponding footprints.

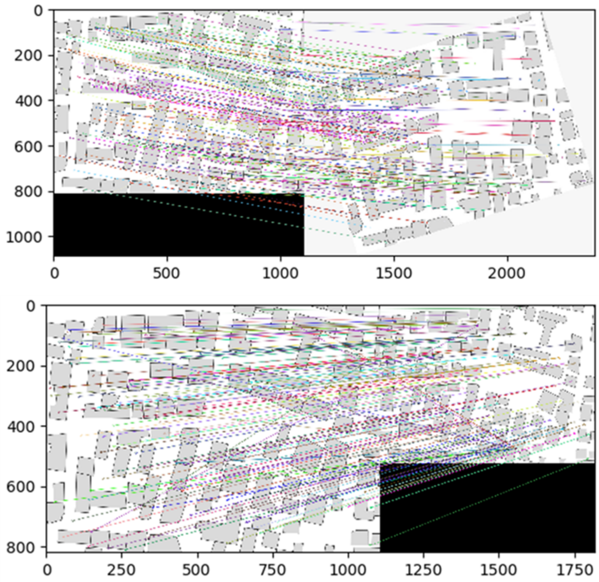

Fig. 4. SIFT matches between pairs of masks having different transformations.

allows to predict the segmentation probabilities for every pixel in an image. These segmentation probabilities indicate how likely each pixel is a background or a building. These probabilities are then turned into discrete segmentation masks. The same segmentation model is used for extracting buildings footprints from old imagery as well as the input aerial image.

B. Image Correspondence

At this point, after extracting buildings footprints from the original input aerial image, we need to find its corresponding mask from the already prepared dataset. The pair of masks will not be perfectly aligned. Many types of transformations may be applied to one of the images with respect to the other. Different scales, different views, overlapping regions between images are examples of such transformations.

Here, and because we do not know the exact location of the images in the global coordinates system, some similarity measure is needed to find the mask from the dataset that best matches with our input image. This similarity measure will help us in deciding whether the two images are for the same scene or not. For this purpose, SIFT image matching algorithm is used.

The objective here is to find a similarity measure that helps us to know that the masks are extracted from the same geographic region despite of the applied transformation.

First, we use SIFT image matching algorithm to detect the interest points in both masks (having different transformations). Then, we compute the descriptors for each one of the images in order to use them in the matching process. SIFT algorithm provides us with the coordinates of the detected keypoints, the set of matched keypoints between the pair of images and many other useful information.

Fig. 4 represents the matching points between pairs of images having different transformations. For visualization, the original image is put on the left side and the other image is put on the right side and the matches are drawn as lines between both images.

Let n and m be the number of keypoints in the first mask and second mask respectively. And let $S = \{P_i / i \in 1, 2, \ldots, n\}$ be the set of detected keypoints in the first mask and $S' = \{P'_i / i \in 1, 2, \ldots, m\}$ be the set of detected keypoints in the second mask. Let M be the set containing the pair of keypoints indices that match with each other. Then $M = \{(i, j) / P_i \in SandP'_j \in S'\}$ are found as matched keypoints. This naming will be used in all next sections.

Both images are of the same scene then there must be proportionality between the relative distances of the keypoints. Thus in all cases this condition must be satisfied:

$$\frac{d(P_a, P_b)}{d(P_c, P_d)} \approx \frac{d(P'_e, P'_f)}{d(P'_g, P'_h)} \tag{1}$$

such that $a, b, c, d \in \{1, 2, \ldots, n\}$ and $e, f, g, g \in \{1, 2, \ldots, m\}$ and $\{a, e\}, \{b, f\}, \{c, g\}, \{d, h\} \in M$.

We compute this factor for the matched keypoints found for the pair of images. In some cases, there might be false matches which lead to some disparity in the values of the factor between the matching pairs. To remove this inconsistency, we remove all the matching pairs that give a factor which is far from the most frequent factor. Then we compute the ratio of the number of the remaining matching pairs over the total number of good matches. We rely on this ratio as a similarity measure between the two images.

In other cases, this similarity factor can vary. In order to have a threshold that can be used in any other case, we compute the similarity factor for 408 pair of masks with different sizes and different applied transformations. We computed the average of the proportionality factor of the 408 pairs of masks and we got 0.88685 as an average factor. But since we are assuming that the pair of masks that we need to compare have differences in buildings, we accept 0.7 as a threshold.

C. Change Detection

After finding the corresponding masks, SIFT matching algorithm is very efficient in detecting the type of the transformation applied to one of the images with respect to the other. We differentiate between three main types of transformations: masks that have overlapping regions, masks that are different in scale and masks that are different in rotation angle. We will explain in details how to detect each type of these transformations by applying simple mathematics on the information provided by SIFT algorithm.

a) Overlapping Regions: For this type of transformation, we use template matching algorithms. This algorithm is available in OpenCV library for computer vision [20]. This algorithm proved to be very efficient in detecting the overlapping regions between two images.

After computing the similarity measure between the pair of masks and checking that the views correspond to the same scene, we try to find the overlapping regions between the pair of masks. The bigger image is then cropped to be aligned with its overlapping region. We apply template matching algorithm to search for the small mask in the bigger one. Although there are some differences in the buildings, the template matching algorithm gives us an accurate result. Now, we have two aligned pair of masks that are ready to detect changes between them.

b) Scale Transformation: In this type of transformation, we aim to find the scale factor between the pair of masks. When we get the scale ratio , we can transform both masks to be in the same scale. The process now is very similar to the one performed in computing the similarity measure since the ratio of distances computed there was in fact the scale factor. So

$$\sigma = \frac{d(P_a, P_b)}{d(P_c, P_d)} \approx \frac{d(P'_e, P'_f)}{d(P'_g, P'_h)} \tag{2}$$

for all $a, b, c, d \in \{1, 2, \ldots, n\}$ and $e, f, g, g \in \{1, 2, \ldots, m\}$ and $\{a, e\}, \{b, f\}, \{c, g\}, \{d, h\} \in M$.

We also remove inconsistencies because of the presence of false matches. Now, we have two aligned pair of masks that are ready to detect changes between them.

c) View Point Transformation (Orientation: In this type of transformation, we aim to find the rotation angle between the pair of masks. When we get the rotation angle, we can transform both masks to be in the same orientation. To calculate the angle of rotation between the two masks, we have to find the angle between the lines that are formed by respective matched points. So

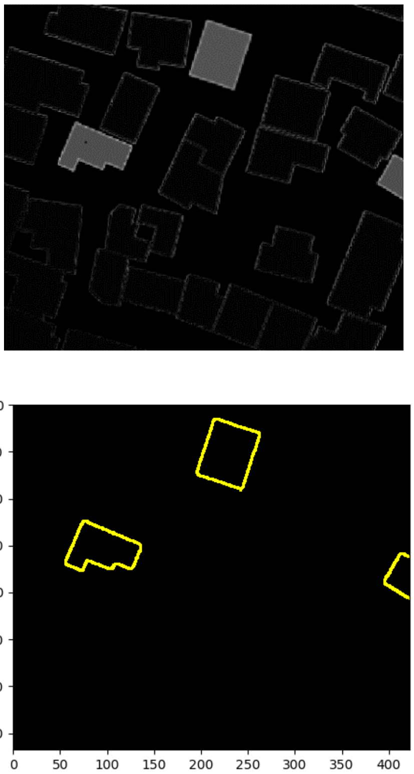

Fig. 5. Difference image before and after filtering.

$$\theta = \frac{d(P_a, P_b)}{d(P_c, P_d)} \approx \frac{d(P'_e, P'_f)}{d(P'_g, P'_h)} \tag{3}$$

We also remove inconsistencies because of the presence of false matches. Now, we have two aligned pair of masks that are ready to detect changes between them.

D. Difference Image

Whatever was the transformation applied to one of the images, at this point we have two aligned images. All what we have to do is to find the difference image. Of course, the difference image will contain some noise because of the differences in the resolution of the pair of masks, the fact that drives us to filter the difference image.

Filtering the noise in the difference image consists of finding the contours in it. Contours are curves joining all the continuous points (along the boundary) having same color or intensity. Contours are very helpful for shape detection or recognition. Since we are using binary images, we have more chance to get a better accuracy. Finding such contours rely on detecting Canny Edges [38].

Fig. 5 represents the noisy difference image and the filtered one.

The contours are projected finally onto one of the original images to show the differences clearly.

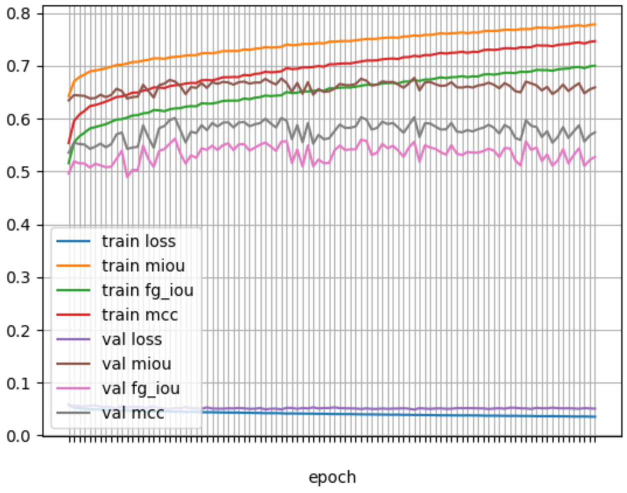

Fig. 6. Evaluation metrics for each epoch of the training process.

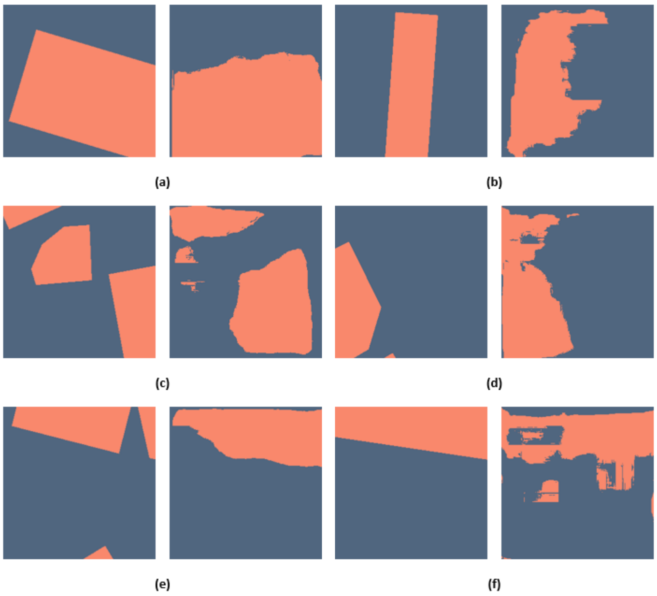

Fig. 7. Comparison between ground truth buildings masks and the predicted ones.

V. Results and Discussions

Fig. 6 shows the evaluation metrics for both training and evaluation sets during the training process. We pick up the checkpoint of the epoch 66 since it has the best values for the validation set. This epoch has the least loss value which is 0.0491. At the same time this epoch has the highest mean intersection over union value which is 0.757.

Fig. 7 shows a comparison between ground truth masks (on the left) an the predicted masks (on the right).

TABLE I
Description of the transformed masks

	Description
A	Scale factor = 1.56, 4 changes
B	Rotation angle = 73°, 2 changes
C	Have overlapping region, 3 changes

TABLE II
Results of SIFT algorithm applied on different pairs of masks

	No. of keypoints in the original mask	No. of keypoints in the transformed mask	No. of matches
A	261	216	169
B	261	779	171
C	261	121	91

For change detection, we used the accurate buildings extracted from OSM to evaluate the change detection procedure, in order to guarantee that the results of the image segmentation do not affect our evaluation.

In order to show the results of the whole workflow, refer to Fig. 2 that shows an aerial image and its corresponding mask. We suppose that this image is acquired now from an aircraft. We also suppose that we have a database containing old masks (extracted from old aerial images). The goal is to find the mask in the database that corresponds to the mask of this aerial image by computing the similarity measure between each pair of masks.

After extracting the buildings' footprints from the image, we manually apply different transformations to the mask in order to evaluate our procedure. Table 1 shows the description of the applied transformations. We also apply manually some changes between the masks.

First, we apply SIFT algorithm to the original mask with each one of the applied mask. Table 2 represents the results of SIFT algorithm, and Fig. 8 shows the resulting matching points found by SIFT for each pair of masks.

Now, we compute the similarity measure and the geometric parameters of the pair of masks to compare them with the ground truth shown in Table 1. The results are shown in Table 3.

As shown in the table, all the similarity measure for the transformed mask with respect to the original mask are greater or equal to the threshold. As for scale factor, the difference between the computed scale factor and the real one for the four pairs of masks does not exceed 0.1. As well, the difference between the computed rotated angle and the real one does not exceed 0.1°. Now, the difference image is computed for each

TABLE III
Similarity measures and geometric parameters computed for the pairs of masks

	Similarity measure	Scale factor	Rotation angle
A	0.692308	1.53581	0.12881
B	0.88636	1.01337	73.0351
C	0.875	0.94348	-0.00921

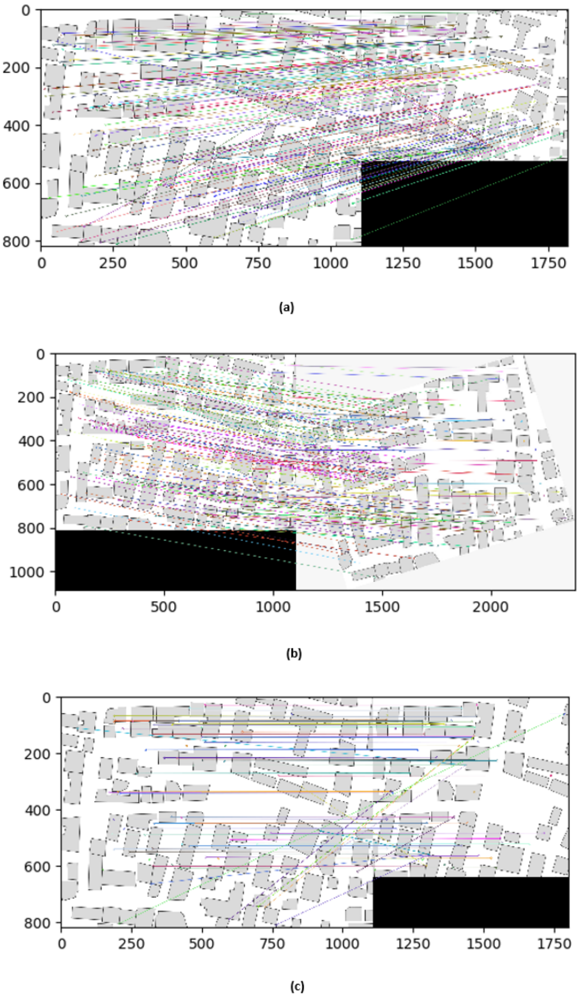

Fig. 8. SIFT matches between each pair of masks.

pair of masks after aligning them. Fig. 8 shows the difference image of each of the four pair of masks.

The procedure was applied on a test set of 80 pairs of aerial images with different characteristics and different applied transformations in order to evaluate our procedure. The following histograms show the accuracy rate of the results of the change detection as well as for the geometric parameters for each type of transformation.

It is clear from the obtained results that our procedure works the best with the scale transformation as well as overlapping regions. However, some errors were encountered with rotation and mixed transformations. The results are expectable since SIFT algorithm is designed to be robust with scale transformation.

As an overall rate, our procedure gives 92.7% of true change detection for different types of transformations.

The strengths of our procedure can be summarized by the following points: (1) this procedure works with simple PNG aerial images without any additional metadata, (2) if the shape

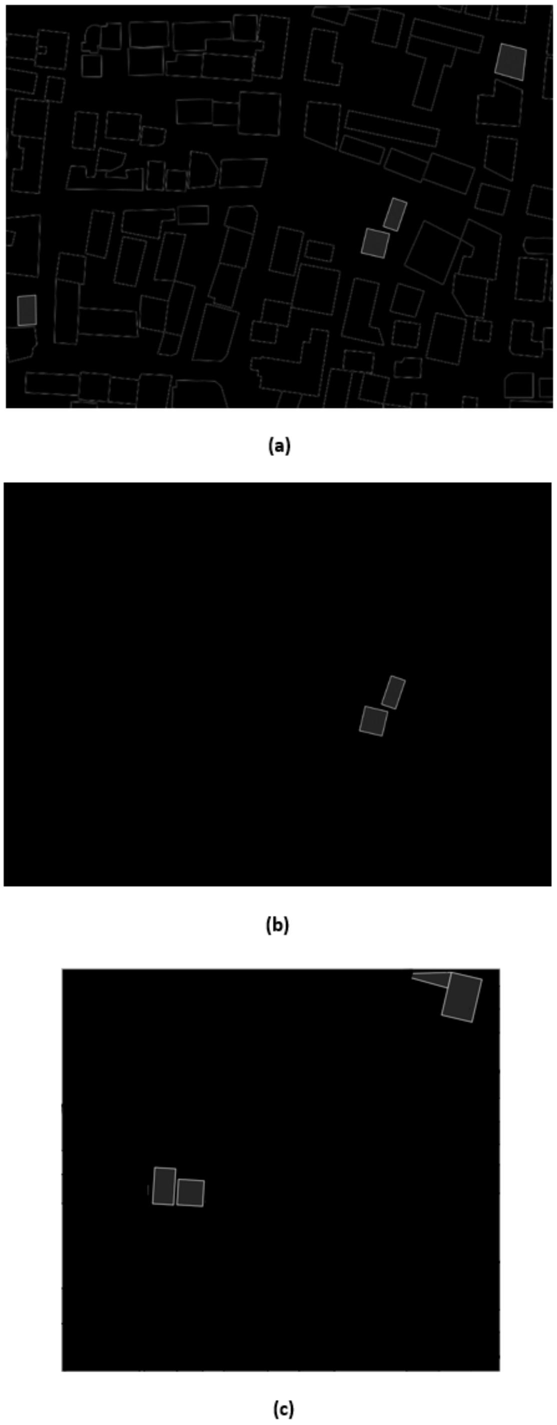

Fig. 9. Difference image between each pair of masks.

TABLE IV
ACCURACY RATE OF THE RESULTS OF THE CHANGE DETECTION WITH DIFFERENT TYPES OF TRANSFORMATIONS

	Transformations			
	Scale	**Orientation**	**Overlapping**	**Mixed**
Accuracy (%)	100	85.5	99	86.3

of buildings in another region are different from the buildings in the training set, simply anyone can train his own dataset and then use the same procedure to detect changes, (3) this procedure can be extended to point of interests other than buildings, finally (4) this procedure is robust against different types of transformations.

However two main limitations encounter this procedure which are (1) this procedure has expensive computation time so it cannot act as real time application and (2) the final results are always dependent to the accuracy of the segmentation phase.

VI. Conclusion

Building change detection in aerial images that differ in many geometric aspects such as scale and view point is a challenging research topic nowadays. A complete solution for this problem is not yet found and developed. This work has presented a complete procedure to detect new and demolished buildings in two aerial images taken at different times. Our procedure worked in three steps. The first step which is extracting building footprints from original aerial images was accomplished using a segmentation model. Using machine learning, specifically convolutional neural network, this model was built by training on a large number of aerial images coupled with their buildings masks. The second step which is image correspondence was done by calculating a similarity factor between each pair of images. At this point, the pair of images that represent the same geographic area is found. The last step, which is change detection, benefits from image matching algorithms in particular SIFT algorithm. This algorithm is applied to align the pair of the images and then compute their difference in order to detect the changed buildings. This procedure showed a change detection rate of 92.7% for different types of transformations.

VII. Challenges and Future Work

A big challenge is faced in our approach which is inability to be implemented as a real time system. The image segmentation phase as well as searching in a database for the mask that corresponds with the input image are computationally expensive, although building the model and preparing the dataset are carried out only once. Further studies must be accomplished in order to find suitable solutions for this critical issue.

Moreover, future work can have a more specific design for the experiment. The overall findings that emerged from our experiments gave us some promising directions to follow for building an optimal, operative, complete and automatic system in the future.

Furthermore, points of interest other than buildings can be taken into consideration in the process of change detection. It can also include roads, vegetation and any other class of objects that can be present in aerial images.

Additionally, enhancing the segmentation model with a larger and more suitable dataset is essential in a further research due to its considerable and significant effect in the improvement of the overall results of the approach, since detecting changes rely truly on the extracted buildings' footprints.

References

[1] J. D. Kiser and D. P. Paine, Aerial Photography And Image Interpretation, Canada: John Wiley Sons, Inc., 2012.

[2] M. N. Favorskaya and L. C. Jain, Computer Vision in Control Systems, Aerial and Satellite Image Processing, vol. 135, M. N. Favorskaya and J. C. Lakhmi, Eds., Canberra: Springer International Publishing, 2018.

[3] N. Paparoditis, M. Jordan and J. P. Cocquerez, ”Building Detection and Reconstruction from Mid- and High-Resolution Aerial Imagery,” Computer Vision And Image Understanding, vol. 72, pp. 122-142, 1997.

[4] G. Wilhauck, ”Comparison of Object Oriented Classification Techniques and Standard Image Analysis For the Use of Change Detection Between SPOT multispectral Satellite Images and Aerial Photos,” International Archives of Photogtammetry and Remote Sensing, vol. XXXIII, 2000.

[5] S. Nebiker, N. Lack and M. Deuber, ”Building change detection from historical aerial photographs using dense image matching and object-based image analysis,” Remote Sensing, vol. 6, pp. 8310-8336, September 2014.

[6] L.-C. Chen and L.-J. Lin, ”Detection of building changes from aerial images and light detection and ranging (LIDAR) data,” Journal of Applied Remote Sensing, vol. 4, no. 1, 2010.

[7] M. C. Alonso, J. A. Malpica, F. Papi, A. Arozarena and A. Martinez-Agirre, ”Change detection of buildings from satellite imagery and lidar data,” International Journal of Remote Sensing, vol. 34, no. 5, p. 1652, March 2013.

[8] I. Tomljenovic, D. Tiede and T. Blaschke, ”A building extraction approach for airborne laser scanner data utilizaing the object based image analysis paradigm,” International Journal of Applied Earth Observation and Geoinformation, vol. 52, pp. 137-148, October 2016.

[9] R. B. Irvin and D. M. McKeown, ”Methods for exploiting the relationship between buildings and their shadows in aerial imagery,” IEEE Transactions on Systems, Man and Cybernetics, vol. 19, no. 6, 1989.

[10] Y. Wang, ”Automatic extraction of building outline from high resolution aerial imagery,” The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Vols. XLI-B3, 2016.

[11] M. Leena, J. Hyyppa and H. Kaartinen, ”Automatic detection of changes from laser scanner and aerial image data for updating buildings map,” Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci., vol. 35, pp. 434-439, July 2004.

[12] M. C. A. Turker and B. Cetinkaya, ”Automatic detection of earthquake-damaged buildings using DEMs created from pre- and post-earthquake stereo aerial photographs,” International Journal of Remote Sensing, vol. 26, no. 4, pp. 823-832, 16 August 2006.

[13] F. Jung, ”Detecting building changes from multitemporal aerial stereopairs,” ISPRS Journal of Photogrammetry and Remote Sensing, vol. 58, no. 3-4, pp. 187-201, January 2004.

[14] F. Rottensteiner, S. Clode, J. C. Trinder and K. Kubik, ”Fusing airborne laser scanner data and aerial imagery for the automatic extraction of buildings in densely built-up areas,” International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, vol. 35, 2006.

[15] L. Y. Chen, T.-A. Teo, Y.-C. Shao and Y.-C. Lai, ”Fusion of LIDAR data and optical imagery for building modeling,” International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences, vol. 35, 2004.

[16] S. Saito and Y. Aoki, ”Building and road detection from large aerial imagery,” in Image Processing: Machine Vision Applications VIII, San Francisco, 2015.

[17] N. Bourdis, M. Denis and H. Sahbi, ”Constrained optical flow for aerial image change detection,” in 2011 IEEE International Geoscience and Remote Sensing Symposium (IGARSS), Vancouver, Canada, 2011.

[18] ”Arcgis.om,” 2019. [Online]. Available: https://pro.arcgis.com/en/pro-app/tool-reference/data-management/detect-feature-changes.htm.

[19] S. Yuheng and Y. Hao, ”Image Segmentation Algorithms Overview,” 2017.

[20] G. Bradski and A. Kaehler, Learning OpenCV, United States: O'Reilly Media, Inc., December 2016.

[21] N. R. Pal and S. K. Pal, ”A Review on Image Segmentation Techniques,” Pattern Recognition, vol. 26, no. 9, pp. 1277-1294, September 1993.

[22] K. S. Fu and J. K. Mui, ”A Survey on Image Segmentation,” Pattern Recoginition, vol. 13, no. 1, pp. 3-16, 1981.

[23] P.-g. Ho, Ed., Image Segmentation, 2011.

[24] A. P. Dhawan, ”Image Segmentation,” in Medical Image Analysis, Wiley-IEEE Press, 2011, pp. 229-264.

[25] ”fizyr/keras-retinanet,” 2019. [Online]. Available: https://github.com/fizyr/keras-retinanet.

[26] W. Abdullah, ”Mask R-CNN for object detection and instance segmentation on Keras and TensorFlow,” GitHub Repository, 2017.

[27] ”ENVI - The Leading Geospatial Analytics Software,” 2019. [Online]. Available: https://www.harrisgeospatial.com/Software-Technology/ENVI.

[28] M. J. Canty, Image Analysis, Classification and Change Detection in Remote Sensing, New York: Taylor Francis Group, 2014.

[29] V. Srivastava, ”Evaluation of various segmentation tools for extraction of urban features using high resolution remote sensing data,” The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, vol. 34, no. XXX.

[30] J. Bauer, N. Sunderhauf and P. Protzel, ”Comapring Several Implementations of Two Recently Published Feature Detectors,” in International Conference on Intelligent and Autonomous Systems (ICAS), Toulouse, France, 2007.

[31] P. M. Panchal, S. R. Panchal and S. K. Shah, ”A Comparison of SIFT and SURF,” International Journal of Innovative Research in Computer and Communication Engineering, vol. 1, no. 2, April 2013.

[32] U. M. Babri, M. Tnavir and K. Khurshid, ”Feature Based Correpondence: A Comparative Study on Image Matching Algorithms,” International Journal of Advamced Computer Science and Applications (IJACSA), vol. 7, no. 3, 2016.

[33] E. Karami, S. Prasad and M. Shehata, ”Image Matching Using SIFT, SURF, BRIEF and ORB: Performance Comparison for Distorted Images,” in Newfoundland Electrical and Computer Engineering Conference, Canada, 2015.

[34] ”Github - RoboSat,” Mapbox, 2018. [Online]. Available: https://github.com/mapbox/robosat.

[35] ”OpenStreetMap OSM,” OpenStreetMap Foundation OSMF, 2010. [Online]. Available: www/openstreetmap.org.

[36] ”GeoFabrik,” OpenStreetMap, 2018. [Online]. Available: geofabrik.de.

[37] ”Mapbox,” Mapbox, 2010. [Online]. Available: www.mapbox.com.

[38] J. Canny, ”A Computational Approach to Edge Detection,” in Readings in Computer Vision, M. A. Fischler and O. Firschein, Eds., Elsevier, 1987, pp. 184-203.