*Sensors & Transducers, Vol. 249, Issue 2, February 2021, pp. 45-53*

www.sensorsportal.com

**Sensors & Transducers**

Published by IFSA Publishing, S. L., 2021
http://www.sensorsportal.com

# Lessons Learned on Conducting Dwelling Detection on VHR Satellite Imagery for the Management of Humanitarian Operations

**[1, *] Lorenz WICKERT, [1] Manfred BOGEN and [1] Marvin RICHTER**

[1] Fraunhofer IAIS, Schloss Birlinghoven 1, 53757 Sankt Augustin, Germany
[1] Tel.: +49 2241 14-3415, Fax: +49 2241 14-2342
E-mail: lorenz.wickert@iais.fraunhofer.de

*Received: 27 November 2020 /Accepted: 29 December 2020 /Published: 28 February 2021*

**Abstract:** The management of humanitarian operations in highly intense situations like migration movements happening at borders often lack current and sufficient information. Satellites do provide large-scale information fast. When dealing with a migration situation, satellite images now can give information about where refugees are before they arrive at a border, giving first responders urgently needed lead time for contingency and capacity planning. Dwelling Detection, a method conducted on satellite images of refugee camps, is able to count the dwellings in a camp. From that, the number of inhabitants in a camp can be derived for forecasting purposes. To count the dwellings, object detection machine learning methods can be used. In Wickert et al. [ASPAI' 2020, **1**, 2020], a dwelling detection workflow using a Faster R-CNN is described. The workflow contains a newly developed annotation method, an inhabitant estimate for analyzed camps and a global confidence factor indicating the quality of the analysis of the image and the estimate of the inhabitants. In this actual extension of Wickert et. al. [ASPAI' 2020, **1**, 2020], lessons learned from multiple training and testing runs are documented, following a detailed analysis of those tests and validations in Wickert et. al. [ISPRS 2020, **2**, 2020]. In this extended article we conclude that the workflow produces results that can be used in humanitarian operations. We further document our lessons learned in developing a dwelling detection workflow and we provide recommendations for training a dwelling detection classifier. We advise humanitarian operators to build a dwelling detection classifier following our recommendations and use satellite images in actual humanitarian operations. This approach can reduce stress for all people involved in a humanitarian (crisis) situation and lead to better decisions in intense migration situations.

**Keywords:** Artificial Intelligence (AI), Machine Learning (ML), Deep Learning, Convolutional Neural Network (CNN), Remote Sensing (RS), Dwelling Detection, Object Detection, Lessons Learned.

## 1. Introduction

Humanitarian operations, with migration management in particular can be complicated and time-critical tasks. Often migration movements allow only a short reaction time due to their high intensity. To ease the work of humanitarian operators, a dwelling detection workflow was developed in [1] to automate parts of the information gathering process. In the paper, the development of the dwelling detection workflow was motivated as following:

"Forced Migration is one of the most pressing issues of society today. The United Nations High Commissioner for Refugees reported that in 2018 'the worlds forcibly displaced population remained yet again at a record high' [3]. This leads to highly intense humanitarian operations like the migration movements in Europe in 2015/16, where humanitarian

operations were challenging to all people involved because of a short reaction time and a lack of sufficient information. Satellite data allows the monitoring of large areas of the earth in a short time. Together with machine learning technology, satellite data can be used to yield large-scale information almost immediately, supporting decision making in humanitarian operations. In this paper, advances in satellite sensors [4, p. 187ff] and object detection using machine learning [5] are used to speed up the task of 'dwelling detection' [6]. This is a fast method to get information on the number of inhabitants of camps when this information is hard to get otherwise, either because those numbers are not available or there is no communication among the different parties, authorities and organization engaged in a humanitarian situation."

After developing the initial dwelling detection workflow, extensive tests were conducted on the workflow's core component, the object detection model. More precisely, different configurations of the annotation workflow, different representations of the input images into the object detection model and the generalization of the workflow were tested. From the results of these tests, recommendations for training a dwelling detection classifier were derived. A detailed analysis of the results is reported in [2]. In this article, we extend the description of the initial dwelling detection workflow by reporting what was tested, the results of the tests as well as the recommendations derived from those tests.

The research described in this paper was conducted in the context of the HUMAN+[1] project. In the HUMAN+ project, a real-time situational awareness system for efficient management of migration movements was developed. Besides the dwelling detection module, camera streams from cameras located at borders were analyzed, social media platforms were monitored and evaluated concerning migration movement and reports from operators in the field were collected and included in the HUMAN+ situational awareness system.

## 2. Methodology

In the following chapter, the methodology for building the dwelling detection classifier is documented in Sections 2.1 – 2.5. These sections are cited from [1]. In Section 2.6, the methodology used for testing the annotation workflow and the dwelling detection classifier is described, following [2].

### 2.1. Dwelling Detection

Due to the fact that individual humans cannot be seen on commercially available satellite images, a method for extracting valuable information about humans from satellite imagery had to be defined. dwelling detection offers a fitting method. It is conducted on very high resolution (VHR) satellite images of refugee camps. Those satellite images are available as result of regular satellite operations and recordings. Dwellings are counted and multiplied with an average, size-based factor of how many people live in one dwelling, giving an estimate over how many people may live in a camp and may eventually arrive at a border. This information can be used in medium-term planning of humanitarian operations during migration situations. Reasons why these information are not available for humanitarian operators: camps are run by private companies [7], camps do not develop as planned because of a rapid growth of inhabitants [8] or because camps are makeshift camps which are created arbitrarily, sometimes called "jungles" [7], [9].

### 2.2. Deep Learning Object Detection

Dwelling detection is traditionally done by experts by hand, taking a lot of time which is not available in crisis situations. Recent advantages in image classification and object detection on images using Convolutional Neural Networks (CNN) allow first approaches to automate this task [10], [11]. A Faster R-CNN [12] offers a state-of-the-art CNN architecture for object detection to eventually develop a dwelling detection workflow. To do so, a complete machine learning workflow (see Fig. 1) was created, following a framework of preparing input data, defining the expected output data and building a core network which constructs the intrinsic and natural relationship of the input-output pair [13].

The training workflow consists of three main steps. In the first step, satellite images annotated by hand are used to prepare ground truth training data consisting of a train- and a validation-set.

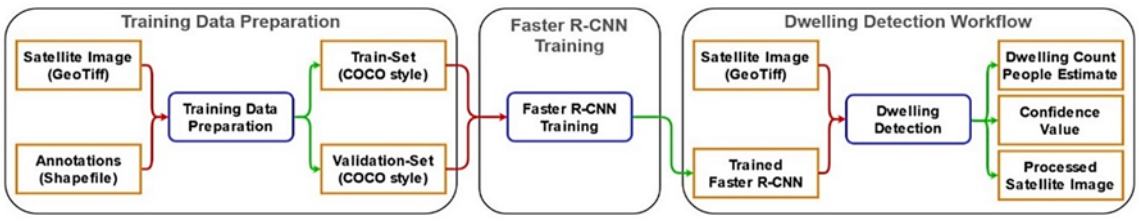


**Fig. 1.** The machine learning workflow of dwelling detection on remote sensing satellite images. Figure taken from [1].

[1] https://giscience.zgis.at/human/ (13N14716-13N14723)

Those sets are used to train a Faster R-CNN in the second step. In the last step, the trained Faster R-CNN is used to analyze a new satellite image of a camp. The classifier outputs a dwelling count and a people estimate, a confidence value concerning the Faster R-CNN analysis and the processed input satellite image with found dwellings marked.

## 2.3. Training Data Preparation

The input data in a dwelling detection task is VHR satellite imagery of refugee camps identified at this point in time by a human being.

In this work, images were taken from Google Earth (Google Inc.)[1] as examples and due to the fact that free satellite images were not available. On those, the dwellings forming a camp have to be found by an object detection algorithm like Faster R-CNN and marked with a bounding box.

A Faster R-CNN is trained using supervised learning. The ground truth data required to train the network consists of bounding boxes around each dwelling. Further, the size of the input images needs to be small enough to be efficiently handled by a Faster R-CNN. Therefore, the input satellite image is split up into 300 pixel × 300 pixel tiles. For speeding up the annotation process, a workflow building up on a seeded region growing algorithm [14], was developed (see Fig. 2). Each dwelling needs to be point-annotated by a human; then each annotation is used by the region growing algorithm as a seed to estimate the extent of one dwelling. The results of the region growing algorithm are then filtered to eliminate bad regions. Around each region, a bounding box is defined and stored in the MS COCO[2]-annotation format [15].

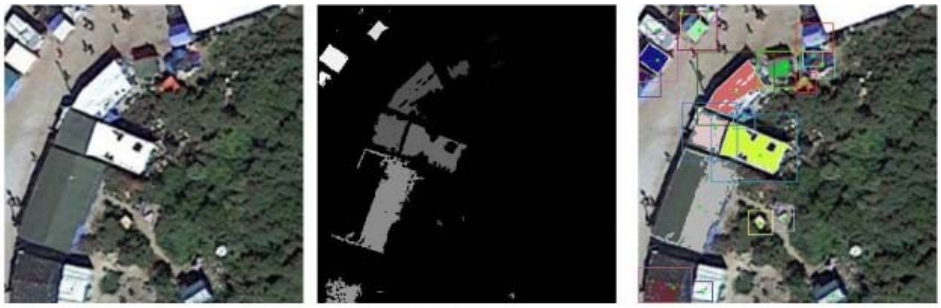


**Fig. 2.** To create training data, the initial tile containing point annotations (a) is analyzed using a region fill algorithm (b) resulting in ground truth boxes (c) around dwellings. Figure taken from [1].

## 2.4. Faster R-CNN Training

Refugee settlements inhabit a huge degree of variety due to their characteristics discussed in Section 2.1. To achieve a network generalizing well, this variety has to be represented in the training data [10]. Therefore, annotations for the training set were made on satellite images of nine different refugee camps. The images have a huge variation in the landscape surrounding the camps, the organization form of the camps, the size of the camps and the quality of the images as a result of the recording distance from the satellite and the satellites optical sensors. On each image, annotations on the upper 50 % of the input image were added to the training set, annotations on the lower 50 % of the input image were added to the validation set.

For implementation and training of the Faster R-CNN an open source implementation was used[3]. A Faster R-CNN consists of a pre-trained backbone network for feature extraction, a Regional Proposal Network (RPN) for generating object proposals, and a classification network outputting bounding boxes around found objects and a class vector for each bounding box. As a backbone, ResNet50 [16] was used. The network was trained on one GPU for 36,000 iterations.

## 2.5. Dwelling Detection Workflow

To analyze a satellite image using the trained Faster R-CNN, a copy of the input image, which is cut in various 300 pixel × 300 pixel tiles to handle its size, is made. Each tile is analyzed separately. Found objects are accepted as a dwelling if their class security for being a dwelling is higher than a defined threshold.

The dwelling detection workflow has two outputs: The first output is a copy of the input satellite image with all found dwellings marked with a bounding box (see Fig. 3). The second output consists of the number of found dwellings and the estimated number of inhabitants in a camp. The number of inhabitants is estimated with regard to the size of the found bounding

[1] https://www.google.com/intl/de_de/earth/
[2] https://cocodataset.org/#home
[3] https://github.com/facebookresearch/Detectron

boxes. Problematic is that the spatial resolution of one pixel can change from image to image and is not always known. Therefore, the estimation algorithm has to be independent of this information. This is achieved by assigning a fixed number of inhabitants to the smallest and the largest dwelling found by the Faster R-CNN. For the dwellings with sizes between those anchor points, the number of inhabitants in a tent is interpolated linearly.

Further, a confidence metric indicating the certainty of the network about the results of the analysis is calculated. The metric is the ratio of the number of found objects $O_{cls \geq 0.85}$ with a class security higher or equal than 0.85 divided by the number of objects $O_{cls \geq 0.5}$ with a class security higher or equal than 0.5:

$$\text{Confidence} = \frac{O_{cls \geq 0.85}}{O_{cls \geq 0.5}} \quad (1)$$

**Fig. 3.** The output of the Faster R-CNN. The input satellite image of a refugee camp located in Idomeni (Greece) was analyzed. The red bounding boxes mark the dwellings found by the Faster R-CNN. Figure taken from [1].

## 2.6. Test and Validation Workflow

The main idea of the test and validation workflow is to train various instances of the Faster R-CNN using different configurations. The resulting Faster R-CNNs are evaluate on uniquely configured evaluation sets. The performance of the resulting Faster R-CNNs on the different evaluation sets is used to derive an assertion about the performance of the training configurations. The main information derived is how well the individual networks learn the training distribution as well as how well they generalize their detection capabilities to unseen satellite imagery.

While training the Faster R-CNNs, three parameters were varied:

1. The color-representation of the images in the training sets with possible values “color”, “grayscale” and “color & gray”;

2. The color-representation of the images which were given to the annotation workflow as input with the possible values “color” and “grayscale”;

3. And the number of iterations the Faster R-CNNs were trained.

Table 1 shows the resulting eight trained networks, where the networks with id 1-5 were discussed in [2] and the networks with id 6-8 have not been discussed before.

**Table 1.** Training configurations of the networks tested. Adapted table taken from [2].

| Id | Conf | Training on | Annotations on | Train Time |
|---|---|---|---|---|
| 1 | n_cc_36k | Color | Color | 36.000 (Iter) |
| 2 | n_gc_27k | Grayscale | Color | 27.000 (Iter) |
| 3 | n_gc_36k | Grayscale | Color | 36.000 (Iter) |
| 4 | n_gg_27k | Grayscale | Grayscale | 27.000 (Iter) |
| 5 | n_gg_36k | Grayscale | Grayscale | 36.000 (Iter) |
| 6 | n_cg_27k | Color | Grayscale | 27.000 (Iter) |
| 7 | n_mix_27k | Color & Gray | Color | 27.000 (Iter) |
| 8 | n_mixg_27k | Color & Gray | Grayscale | 27.000 (Iter) |

The different networks were tested on two sets of validation sets with four validation sets each. In the first set of validation sets, the images contained are those used to train the networks. In the second set of validation sets, new images not seen by the networks before, but of the same camps the networks were trained on, are contained. The four individual validation sets inside one set differentiate in the color-representation of the images inside each set and the color-representation of the input-images in the annotation algorithm. The complete configuration of each validation set is documented in Table 2. Validation sets with id 1-3 & 5-7 were discussed in [2], validation sets with id 4 & 8 have not been discussed before.

**Table 2.** Configurations for the Validation Sets the different networks were trained on. Adapted table taken from [2].

| Id | Conf | Type | Images | Annotations on |
|---|---|---|---|---|
| 1 | v_orig_cc | Trained on | Color | Color |
| 2 | v_orig_gc | | Grayscale | Color |
| 3 | v_orig_gg | | Grayscale | Grayscale |
| 4 | v_orig_cg | | Color | Grayscale |
| 5 | v_new_cc | Not Seen | Color | Color |
| 6 | v_new_gc | | Grayscale | Color |
| 7 | v_new_gg | | Grayscale | Grayscale |
| 8 | v_new_cg | | Color | Grayscale |

As performance metrics, Average Precision (AP) and Average Recall (AR) were calculated using the MS COCO Detection evaluation metrics [17]. Further the F1-score [18] was calculated for each network on all eight validation sets.

## 3. Results

In [1] it was outlined, that "a complete machine learning workflow was implemented. For training data creation an annotation method using point annotations which are processed using a seeded region growing algorithm to generate ground-truth data was developed. This method allows creating a sufficient amount of ground-truth bounding boxes in a short amount of time".

The results for the initial network used in the conference paper, corresponding to network *n_cc_36k*, are reported in Section 3.1 and the results regarding the dwelling detection performance are discussed in Section 3.2, both citing [1]. Section 3.3 discloses the results of the tests and validations described in Section 2.6, following [2].

### 3.1. Machine Learning Results

The Faster R-CNN *n_cc_36k* was trained and tested on nine satellite images of refugee camps with different sizes from different parts of the world on different landforms and with varying image quality. On the validation set, comprising of ground truth data from all of the nine used satellite images, a mean Average Precision (mAP) of 68.2 % and a mean Average Recall (mAR) of 72.0 %, determined using the MS COCO detection evaluation metrics [17], was achieved. It can be concluded that the network was able to construct the relationship between the input-satellite image and the dwellings appearing on the image. This holds true for the huge variety of camps the network was trained on.

### 3.2. Dwelling Detection Results

The trained Faster R-CNN finds objects on the analyzed image. To estimate the number of inhabitants in a camp the workflow described in Section 2.5, which is independent of the spatial resolution of the input satellite image and therefore works without image-specific configurations, is used. For the estimation, the number of inhabitants in the smallest dwelling was set to three and the number of inhabitants in the largest dwelling was set to twelve. These limits were set based on an information from the HUMAN+ project partner Johanniter Austria, saying that around ten people live in a standard tent in a transit camp and around six people live in a standard tent in a permanent camp. The threshold for a found object being accepted as a dwelling was set to a class security of 0.85, following [11].

To evaluate the estimate numbers, real-world numbers of inhabitants for each camp, measured at around the time the satellite image was shot, were researched in newspaper articles. It has to be noted that these numbers are not official numbers and are therefore fuzzy. Further, the estimations are based on no concrete knowledge of an individual camp. Therefore, the absolute numbers cannot be assumed as the real numbers but function as an early warning system, stating an order of magnitude of the number of inhabitants.

Consequences from the result of the workflow have to be taken in accordance with the computed confidence metric and the visual output of the network. A look on the annotated satellite image produced by the workflow can give a first impression on how successful the analysis was. Further, the confidence metric makes a statement about how well the Faster R-CNN was able to work with the input image. A low confidence means, that there were a lot of objects the network did not discard in the first place but is also not sure about the objects being a dwelling. In that case, we recommend to humanitarian operators to gather more information from different sources concerning the camp and the area around it. A higher confidence indicates that humanitarian operators can include the order of magnitude of displaced peoples in their medium-term planning.

### 3.3. Test and Validations Results

Table 3 shows the results for the tests on images used for training the networks, Table 4 the results for the tests on new images.

Looking at Table 3, the most expectable result is, that the best-performing network for a validation set is the network that was trained on training data constructed using the same configurations as the validation set. This shows, that the networks learn the distribution of the input data they are trained with as expected [19].

The second parameter influencing the performance of a network is the training time. When comparing the networks which have the same configuration concerning the input images and the annotation-mode but were trained for a different number of iterations (**_gc_** and **_gg_**), it can be seen in Table 3 that the longer trained networks only perform better on the validation sets having the same distribution as the training set the networks were trained on. On the other validation sets, the performance is about the same between the longer (36.000 iterations) and the shorter (27.000 iterations) trained networks. In Table 4, which shows the generalization capabilities of the networks, it can even be seen that the shorter-trained networks outperform the longer trained networks throughout all validation sets. This is a clear indicator that the longer trained networks struggle with overfitting [20].

**Table 3.** Validation Results of the individually trained networks (*n_**) on validation sets built from images the networks were trained on (*v_orig_**). Adapted table taken from [2].

| Val.-Set | v_orig_cc | | | v_orig_gc | | | v_orig_gg | | | v_orig_cg | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Metric** | **AP** | **AR** | **F1** | **AP** | **AR** | **F1** | **AP** | **AR** | **F1** | **AP** | **AR** | **F1** |
| **Network** | | | | | | | | | | | | |
| **n_cc_36k** | 68.2 % | 72.0 % | 70.0 % | 55.9 % | 61.1 % | 58.4 % | 27.6 % | 35.0 % | 30.9 % | 22.1 % | 29.5 % | 25.3 % |
| **n_gc_27k** | 57.9 % | 63.4 % | 60.5 % | 60.9 % | 65.8 % | 63.3 % | 23.7 % | 30.4 % | 26.6 % | 19.9 % | 25.2 % | 25.2 % |
| **n_gc_36k** | 61.6 % | 66.2 % | 63.8 % | 65.2 % | 69.1 % | 67.1 % | 23.6 % | 30.9 % | 26.8 % | 19.7 % | 25.5 % | 22.2 % |
| **n_gg_27k** | 36.0 % | 48.2 % | 41.2 % | 35.4 % | 47.7 % | 40.6 % | 42.6 % | 48.1 % | 45.2 % | 14.1 % | 21.3 % | 17.0 % |
| **n_gg_36k** | 35.5 % | 47.8 % | 40.7 % | 35.5 % | 47.9 % | 40.8 % | 46.2 % | 50.6 % | 48.3 % | 14.6 % | 21.6 % | 17.4 % |
| **n_cg_27k** | 43.6 % | 54.7 % | 48.5 % | 42.6 % | 52.2 % | 46.9 % | 19.5 % | 29.2 % | 23.4 % | 29.6 % | 34.6 % | 31.9 % |
| **n_mix_27k** | 59.3 % | 64.9 % | 62.0 % | 58.0 % | 63.6 % | 60.7 % | 23.4 % | 30.0 % | 26.3 % | 19.9 % | 25.4 % | 22.3 % |
| **n_mixg_27k** | 42.2 % | 53.4 % | 47.1 % | 42.1 % | 53.4 % | 47.1 % | 18.7 % | 28.3 % | 22.5 % | 29.0 % | 34.1 % | 31.3 % |

**Table 4.** Validation Results of the individually trained networks (*n_**) on validation sets built from images not seen by the networks while training (*v_new_**). Adapted table taken from [2].

| Val.-Set | v_new_cc | | | v_new_gc | | | v_new_gg | | | v_new_cg | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Metric** | **AP** | **AR** | **F1** | **AP** | **AR** | **F1** | **AP** | **AR** | **F1** | **AP** | **AR** | **F1** |
| **Network** | | | | | | | | | | | | |
| **n_cc_36k** | 6.6 % | 13.0 % | 8.8 % | 5.3 % | 11.7 % | 7.3 % | 9.2 % | 14.9 % | 11.4 % | 10.4 % | 16.5 % | 12.8 % |
| **n_gc_27k** | 8.4 % | 12.8 % | 10.1 % | 8.8 % | 13.3 % | 10.6 % | 10.1 % | 14.4 % | 11.9 % | 9.7 % | 13.9 % | 11.4 % |
| **n_gc_36k** | 8.2 % | 12.5 % | 9.9 % | 8.3 % | 12.8 % | 10.1 % | 9.8 % | 13.7 % | 11.4 % | 9.6 % | 13.5 % | 11.2 % |
| **n_gg_27k** | 7.9 % | 13.9 % | 10.1 % | 8.4 % | 14.6 % | 10.7 % | 11.1 % | 17.2 % | 13.5 % | 10.4 % | 16.2 % | 12.7 % |
| **n_gg_36k** | 7.3 % | 12.7 % | 9.3 % | 7.7 % | 13.4 % | 9.8 % | 9.7 % | 15.4 % | 11.9 % | 9.0 % | 14.5 % | 11.1 % |
| **n_cg_27k** | 9.6 % | 17.5 % | 12.4 % | 6.5 % | 14.2 % | 8.9 % | 9.4 % | 17.1 % | 12.1 % | 12.4 % | 20.0 % | 15.3 % |
| **n_mix_27k** | 9.3 % | 13.8 % | 11.1 % | 8.9 % | 13.2 % | 10.6 % | 10.2 % | 14.4 % | 11.9 % | 10.7 % | 15.0 % | 12.5 % |
| **n_mixg_27k** | 10.7 % | 18.0 % | 13.4 % | 9.9 % | 17.4 % | 12.6 % | 12.9 % | 20.3 % | 15.8 % | 13.4 % | 20.5 % | 16.2 % |

A highly interesting behavior of the networks becomes evident when looking at the color representation of the images in the training set used to learn the networks models. Table 3 shows that the models learned on color-images achieve the highest overall performance over all conducted tests. This indicates that color images enable more detailed feature learning. However, the networks trained on grayscale images generalize better on unseen data compared to those networks trained on color images.

A possible explanation for that can be found in [21]: The authors show that CNNs tend to learn object textures instead of object shapes when building a model of the input data. Inputting grayscale images in a CNN, and thus making textures more abstract, could force the CNN to take the object shapes more into consideration when learning the input data.

From that observation, the hypothesis was made that for best generalization a network should be trained with a set of training images that contains color- and grayscale-images. The hypothesis was tested by training the networks *n_mixg_27k* and *n_mix_27k*. To evaluate and compare the generalization capabilities, we examined the performance on unknown images in Table 4. Therefore, the average F1-score over all validation sets containing unknown images (*v_new_**) was aggregated in Table 5.

It is observed that network *n_mixg_27k* has a more than 2 % higher average F1 score on the *v_new_** validation sets than the other trained networks. The median of all average F1-scores is 11.5 % and the standard deviation of all networks average F1 scores is 1.3 %. Network *n_mixg_27k* deviation from the mean is 2.3 times higher than the standard deviation of all averaged F1 scores over all networks. It can therefore be argued that *n_mixg_27k* generalizes better than the other networks.

**Table 5.** Average F1 over all validation sets containing images not seen by the networks before.

| Network | Average F1 on v_new* |
|---|---|
| n_cc_36k | 10.0 % |
| n_gc_27k | 11.0 % |
| n_gc_36k | 10.7 % |
| n_mix_27k | 11.5 % |
| n_gg_27k | 11.7 % |
| n_gg_36k | 10.5 % |
| n_cg_27k | 12.2 % |
| n_mixg_27k | 14.5 % |

Regarding the best annotation method, only observations and no precise conclusions concerning the best mode are possible. Table 3 shows that networks trained on annotations made on grayscale images generally achieve a lower maximum AP and AR but are more stable on validation sets with annotations made on color images than networks trained on annotations made on color images are on validation sets with annotations made on grayscale images. This higher generalization capability is further indicated by the tests run on the *v_new_** validation

sets, where the networks trained on annotations made on grayscale images (*n_gg_27k*, *n_cg_27k*) perform better than network *n_mix_27k* trained on mixed-representation input data with annotations made on color images.

## 4. Discussion

The created workflow can be discussed from two perspectives: On the one hand, Section 4.1 take a broad look on the whole workflow, discussing possible improvements of the workflow in terms of updating technology and methodology citing [1]. Further, the overall real-world applicability is discussed in this section. On the other hand, Section 4.2 zooms into the building and training of the dwelling detection classifier, the Faster R-CNN model itself, following [2]. There, recommendations for training an object detection classifier are derived from the observations made in Section 3.3.

### 4.1. Dwelling Detection Workflow

The workflow was developed with the goal of a fast applicability. This goal was reached successfully by using open software and data, speeding up the slow and cumbersome process of ground-truth generation and annotation and building convenient workarounds when information about the data was missing. Nevertheless, trade-offs between accuracy and applicability had to be made:

The seeded region growing algorithm used for speeding up annotating ground truth data is rather simple. Plenty of improved image segmentation algorithms that are existing today [22] could improve the ground truth generation. A more accurate ground truth segmentation would also allow switching the deep learning architecture from a Faster R-CNN to a Mask R-CNN [23]. Mask R-CNN builds up on the Faster R-CNN architecture and allows image segmentation on top of object detection in images. Using segmentations of dwellings can yield more accurate estimates for the inhabitant estimation. Further accuracy can be achieved by using georeferenced satellite imagery where the spatial resolution of a pixel is known. Combining the extend or the footprint of a dwelling with the spatial resolution of a pixel allows to accurately determine the size of a dwelling which can be used for inhabitant estimation based on a parameter taking the actual size of a tent into account.

If accuracy or fast applicability is more important is nevertheless context dependent. For operators managing one specific camp, more accurate results are essential. For humanitarian operators working on broader migration situations, fast information giving insights about the magnitude of the situation are important. In this context, the presented workflow is a good enhancement: We presented actual workflow results in September 2019 to professionals in migration management like the German Red Cross[1] as part of an international exercise carried out in the HUMAN+ project. The additional information about people in the refugee camps generated through dwelling detection was appreciated unanimously. Though those professionals wanted to have absolute numbers, this is not doable due to the uncertainties described in this paper. The number of people in a refugee camp has to be seen in context with the confidence value generated. Those numbers must be combined and merged with other information gathered in the HUMAN+ project. Only then a meaningful picture of the situation can be achieved to help first responders to make the right decisions in crisis management and refugees to be taken care of.

### 4.2. Training Recommendations

The tests and validations described in Section 3.3 reveal some interesting properties indicating measures to increase the performance of a (dwelling detection) object detection classifier. From those observed properties, recommendations become available for future projects developing dwelling detection classifiers.

First, we recommend having a high variance in the training data. Variance should be present concerning technical parameters, meaning that the training set should contain (satellite) imagery of different image quality, different spatial resolutions as well as different environment conditions. Further variance should be present concerning semantic parameters of the images. In the case of dwelling detection that means having different types of camps and landscape-types in the training set.

Highly important for generalization purposes is the color-representation of the images in the training set. As shown in Section 3.3, color-images enable a higher maximum performance in a trained network and grayscale images enable better generalization capabilities in networks. In this article, it is shown that having both, color- and grayscale-images, in the training set improves the generalization capabilities of a trained dwelling detection network. We therefore highly recommend converting some training images to grayscale when working with a training set consisting of color-images.

Another problem that we found during the tests and validations was overfitting. Two measures can be taken to prevent overfitting. On the one hand, it is important to have a high variance in the training set. We propose as a rule of thumb that it is better to annotate few dwellings on many satellite images than to annotate many dwellings on only a few satellite images. On the other hand we suggest to use well

[1] https://www.drk.de/en/

established deep learning approaches to tackle overfitting like data augmentation techniques [24] and early stopping [25].

Concerning the annotation algorithm, no direct recommendations can be given from a technical point of view. For a general discussion on the fast annotation approach see Section 4.1.

## 5. Conclusions

Concerning the dwelling detection workflow we concluded in [1]: "A machine learning based dwelling detection classifier was built using modern object detection techniques, a classifier which yields results almost immediately helpful in humanitarian operations. An important step for doing so is the developed training data preparation workflow, which is fast and convenient for annotations.

The workflow outputs a result in the magnitude of the real numbers of inhabitants in a migrant camp. It works on the image without the need of adjustments by the user, thus generating rapid and useful information. The generated information is best combined with other information about the migration situation from different sources. The visual representation of the results with bounding boxes drawn on the input image as well as the calculated confidence factor increase the interpretability of the results. The results were deemed to be useful by experts in the field."

Additionally, we carried out extensive tests and validations. An exhaustive analysis of these tests can be found in [2]. In this article, the hypothesis created in [2] that a mixture of color- and grayscale-images in the training data set improves the generalization capabilities of an object detection classifier, was backed up by training a new network which shows a higher generalization capability compared to the networks trained in [2]. Finally, as lessons learned, recommendations for training a dwelling detection classifier, derived from the tests and validations conducted were specified.

## Acknowledgements

The authors acknowledge the financial support by the German Federal Ministry of Education and Research (BMBF) and the Federal Ministry Republic of Austria Climate Action, Environment, Energy, Mobility, Innovation and Technology (BMVIT/BMK) in the framework of the HUMAN+ project (funding code 13N14716 to 13N14723)

## References

[1]. L. Wickert, M. Bogen, M. Richter, Dwelling Detection on VHR satellite imagery of Refugee camps using a Faster R-CNN, in *Proceedings of the 2nd International Conference on Advances in Signal Processing and Artificial Intelligence (ASPAI'2020)*, Berlin, November 2020, pp. 7–11

[2]. L. Wickert, M. Bogen, M. Richter, Supporting the Management of Humanitarian Operations Concerning Migration Movements with Remote Sensing, in *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, Vol. XLIII-B3-2020, August 2020, pp. 233–240.

[3]. Global Trends - Forced Displacement in 2018, UNHCR, Switzerland, 2018. Accessed: July 30, 2019. [Online]. Available: https://www.unhcr.org/statistics/unhcrstats/5d08d7ee7/unhcr-global-trends-2018.html.

[4]. J. B. Campbell, R. H. Wynne, Introduction to Remote Sensing, Fifth Edition, *Guilford Press*, 2011.

[5]. Z.-Q. Zhao, P. Zheng, S.-T. Xu, X. Wu, Object Detection with Deep Learning: A Review, *IEEE Transactions on Neural Networks and Learning Systems*, 2019, pp. 1–21.

[6]. K. Spröhnle, D. Tiede, E. Schoepfer, P. Füreder, A. Svanberg, T. Rost, Earth Observation-Based Dwelling Detection Approaches in a Highly Complex Refugee Camp Environment — A Comparative Study, *Remote Sensing*, Vol. 6, Issue 10, September 2014, pp. 9277–9297.

[7]. I. Katz, A network of camps on the way to Europe, *Forced Migration Review*, Issue 51, 2016, pp. 17-19.

[8]. A. Dalal, A. Darweesh, P. Misselwitz, A. Steigemann, Planning the Ideal Refugee Camp? A Critical Interrogation of Recent Planning Innovations in Jordan and Germany, *UP*, Vol. 3, Issue 4, December 2018, pp. 64-78.

[9]. B. Beznec, M. Speer, M. S. Mitrović, Governing the Balkan route: Macedonia, Serbia and the European border regime, *Rosa Luxemburg Stiftung Southeast Europe*, 2016.

[10]. J. A. Quinn, M. M. Nyhan, C. Navarro, D. Coluccia, L. Bromley, M. Luengo-Oroz, Humanitarian applications of machine learning with remote-sensing data: review and case study in refugee settlement mapping, *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, Vol. 376, Issue 2128, September 2018, p. 20170363.

[11]. O. Ghorbanzadeh, D. Tiede, Z. Dabiri, M. Sudmanns, S. Lang, Dwelling Extraction in Refugee Camps Using CNN – First Experiences and Lessons Learnt, *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, Vol. XLII–1, September 2018, pp. 161–166.

[12]. S. Ren, K. He, R. Girshick, J. Sun, Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 39, Issue 6, June 2017, pp. 1137–1149.

[13]. L. Zhang, L. Zhang, B. Du, Deep Learning for Remote Sensing Data: A Technical Tutorial on the State of the Art, *IEEE Geoscience and Remote Sensing Magazine*, Vol. 4, Issue 2, June 2016, pp. 22–40.

[14]. R. Adams, L. Bischof, Seeded region growing, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 16, Issue 6, June 1994, pp. 641–647.

[15]. T.-Y. Lin, *et al.*, Microsoft COCO: Common Objects in Context, in *Proceedings of the Computer Vision – ECCV 2014*, Vol. 8693, (D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, Eds.), *Springer International Publishing*, 2014, pp. 740–755.

[16]. K. He, X. Zhang, S. Ren, J. Sun, Deep Residual Learning for Image Recognition, in *Proceedings of the*

*IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016, pp. 770 - 778.
[17]. The COCO Consortium, Common Objects in Context, *Detection Evaluation*, 2019. http://cocodataset.org/#detection-eval
[18]. N. Chinchor, MUC-4 evaluation metrics, in *Proceedings of the 4th Conference on Message Understanding (MUC4'92)*, McLean, Virginia, 1992, pp. 22-29.
[19]. Y. LeCun, Y. Bengio, G. Hinton, Deep learning, *Nature*, Vol. 521, Issue 7553, May 2015, pp. 436–444.
[20]. N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, R. Salakhutdinov, Dropout: a simple way to prevent neural networks from overfitting, *The Journal of Machine Learning Research*, Vol. 15, Issue 1, 2014, pp. 1929–1958.
[21]. R. Geirhos, P. Rubisch, C. Michaelis, M. Bethge, F. A. Wichmann, W. Brendel, ImageNet-trained CNNs are biased towards texture; increasing shape bias improves accuracy and robustness, *in Proceedings of the Seventh International Conference on Learning Representations (ICLR' 2019)*, New Orleans, Louisiana, USA, 2019.
[22]. H. Zhu, F. Meng, J. Cai, S. Lu, Beyond pixels: A comprehensive survey from bottom-up to semantic image segmentation and cosegmentation, *Journal of Visual Communication and Image Representation*, Vol. 34, January 2016, pp. 12–27.
[23]. K. He, G. Gkioxari, P. Dollar, R. Girshick, Mask R-CNN, 2017, Accessed: 31 Jul. 2019, pp. 2961–2969. [Online]. Available: http://openaccess.thecvf.com/content_iccv_2017/html/He_Mask_R-CNN_ICCV_2017_paper.html
[24]. A. Mikołajczyk, M. Grochowski, Data augmentation for improving deep learning in image classification problem, in *International Interdisciplinary PhD Workshop (IIPhDW)*, 2018, pp. 117–122.
[25]. Y. Yao, L. Rosasco, A. Caponnetto, On Early Stopping in Gradient Descent Learning, *Constructive Approximation*, Vol. 26, Issue 2, August 2007, pp. 289–315.