SUPERVISED DETECTION OF BOMB CRATERS IN HISTORICAL AERIAL IMAGES USING CONVOLUTIONAL NEURAL NETWORKS

D. Clermont*, C. Kruse, F. Rottensteiner, C. Heipke

Institute of Photogrammetry and GeoInformation, Leibniz Universität Hannover, Germany
- (clermont, kruse, rottensteiner, heipke)@ipi.uni-hannover.de

ICWG II/III: Pattern Analysis in Remote Sensing

KEY WORDS: Object Detection, Convolutional Neural Networks, Aerial Wartime Images, Bomb Craters

ABSTRACT:

The aftermath of the air strikes during World War II is still present today. Numerous bombs dropped by planes did not explode, still exist in the ground and pose a considerable explosion hazard. Tracking down these duds can be tackled by detecting bomb craters. The existence of a dud can be inferred from the existence of a crater. This work proposes a method for the automatic detection of bomb craters in aerial wartime images. First of all, crater candidates are extracted from an image using a blob detector. Based on given crater references, for every candidate it is checked whether it, in fact, represents a crater or not. Candidates from various aerial images are used to train, validate and test Convolutional Neural Networks (CNNs) in the context of a two-class classification problem. A loss function (controlling what the CNNs are learning) is adapted to the given task. The trained CNNs are then used for the classification of crater candidates. Our work focuses on the classification of crater candidates and we investigate if combining data from related domains is beneficial for the classification. We achieve a F1-score of up to 65.4% when classifying crater candidates with a realistic class distribution.

1. INTRODUCTION

Although the last air raids of World War II happened more than 70 years ago, their aftermath still poses a threat to the people in Europe. Planes dropped numerous bombs, many of which did not explode (Merler et al., 2005) and, thus, still lie underground to this day. These duds sometimes explode without any external stimulation, other times they are dug out during construction works. In order to neutralise the duds, they have to be tracked down, which can be tackled by detecting bomb craters in historical aerial images. The underlying assumption is that bombs were dropped in groups, i.e. the existence of one or more craters increases the probability that a dud lies in the vicinity of those craters. This work proposes a method for the automatic detection of bomb craters in historical aerial wartime images.

Our approach consists of two steps. In the first step, we extract crater candidates from given images by means of a standard blob detector (Bradski, 2000), thus exploiting the blob-like appearance of craters in aerial images. The goal of this proposal extraction is to obtain all craters present in the image as crater candidates. These crater candidates are represented by square bounding boxes centred around the detected blobs, and their sizes correspond the detected blobs' sizes.

In the second step, we present these candidates to a Convolutional Neural Network (CNN) (LeCun et al., 1989), which classifies the candidates either as *crater* or as *background*. To this end, we utilize a pre-trained version of *Inception ResNet V2* (Szegedy et al., 2017) as a feature extractor. While carrying out the actual classification using two fully connected layers and a softmax layer, we also fine-tune the pre-trained network. For training, we use the softmax cross entropy loss and extend it by a parameter, which is used to compensate for an imbalanced distribution of training samples for the two classes. We carry out an experimental evaluation of our methods using historical aerial images. In this context we also investigate the transferability between images from Germany and Italy.

The remainder of this paper is structured as follows. In section 2 related work regarding image classification, object detection in general and crater detection in particular are outlined and discussed. Section 3 describes the developed methods for the detection of bomb craters. We present the used aerial images in section 4. In section 5 we describe the generation of training samples for the CNN and experimentally evaluate the developed methods based on the given data. The paper concludes with section 6, where we summarise the achieved results and give recommendations for future work.

2. RELATED WORK

We start this overview on related work with a discussion of methods for object detection using CNNs. Afterwards we have a look at work dealing with crater detection with and without CNNs. Because of the similarity to our task of detecting bomb craters, work dealing with the detection of planetary craters is also considered.

Although CNNs were first presented 30 years ago (LeCun et al., 1989), their use for image classification was revolutionised with the development of AlexNet (Krizhevsky et al., 2012), laying the foundation for deep learning. To make use of the strong potential of CNNs in the context of object detection, this task can be split up into two sub-tasks, namely localisation and classification (Sermanet et al., 2014); this separation has long be known to be beneficial in image analysis, see e.g. Schickler (1995). While the localisation is used to generate object proposals, i.e. to find out where in an image an object might be present, the classification is used to classify the

*Corresponding author

proposals as one of multiple classes (e.g. foreground/object or background). The proposals can for example be generated by a standard sliding window approach, as was done by Brenner et al. (2018) in the context of crater detection or by Sermanet et al. (2014) in the broader context of general object detection for the ImageNet Large Scale Visual Recognition Challenge 2013. With the advent of Faster R-CNN (Ren et al., 2015), the generation of proposals was tackled using neural networks, too. This allowed the CNN to be trained end-to-end for the object detection while learning where plausible proposals might reside in an image. Nevertheless, the generation of proposals by means of windows of different sizes and ratios ultimately relies on a sliding window approach, too. We consider this to be a drawback of the approach because of the large amount of proposals being generated, which in turn might lead to a larger amount of false positive classifications.

The two-step-approach of localisation and classification is also applied in the context of crater detection. Merler et al. (2005) deal with the detection of bomb craters in historical images. They extract proposals using a sliding window approach, while classifying those proposals using an accuracy-sensitive variant of AdaBoost. They reported a lot of false positive detections on initial experiments, which led them to weighting the classification errors caused by false positive and false negatives. One problem in this approach is the large amount of proposals created by the sliding window approach, especially because of the sparse distribution of bomb craters in aerial images.

Regarding the detection of planetary craters, Urbach & Stepinski (2009) make use of the observation that the appearance of craters is characterised by a pair of crescent-like highlight and shadow regions. Their approach for extracting proposals consists of detecting those highlight and shadow regions separately. Using a library of reference shapes, they omit regions having shapes that do not correspond to their model. The classification is carried out using a decision tree with hand-crafted features. Their reported results suggest that hand-crafted features may be a problem because these may not be suitable for differentiating between *crater* and *background*, resulting in a rather large number of false positives.

A model-based approach for the detection of bomb craters is presented by Kruse et al. (2019). Here, marked point processes in combination with RJMCMC methods are used to sample different object configurations. In particular, they compare ellipses and circles as models for the craters. The underlying assumption is that craters exhibit a large image gradient along their edge and that craters do not overlap. To limit the search space for their sampling technique, they make use of a simple blob detector, exploiting the blob-like appearance of craters. Although their approach avoids the need for training data, the chosen model is not sufficient for detecting non-elliptical variations of bomb craters while being depended of many parameters.

Regarding the detection of planetary craters using CNN, Cohen et al. (2016) present an approach using a shallow CNN for the classification of crater candidates, training the network with few training samples from scratch. They use the same method for retrieving crater candidates as Urbach & Stepinski (2009). Their results show that their approach outperforms other crater detection algorithms by a large margin, highlighting the potential of CNNs for that task. A similar approach for the detection of planetary craters was chosen by Emami et al. (2015). They make use of a multi-scale Canny edge detector and convex grouping to extract crater candidates, while classifying those candidates with a shallow CNN. Because of the small amount of training data, we believe that by making use of a pre-trained CNN, their classification can be improved. As was e.g. demonstrated by Sharif Razavian et al. (2014), using features from a pre-trained CNN is suitable even for a new recognition task, which is especially helpful when dealing with few training data.

To the best of our knowledge, Brenner et al. (2018) presented the only work dealing with the detection of bomb craters using CNNs. Thus, this work is particularly related to ours. They make use of a sliding window approach for the extraction of crater candidates while classifying those candidates with a CNN, employing the DenseNet architecture presented by Huang et al. (2017). They trained the CNN with an equal amount of samples of both classes *crater* and *background* using Nesterov Momentum. When testing their approach on samples with the same ratio, they achieved a precision of 90.7%, but in a case of a more realistic class distribution, where the ratio of *crater* and *background* samples is approximately 1:250, the precision drops to 4.0%. Their approach would probably benefit from a weighting of errors to reduce the number of false positives, as was proposed by Merler et al. (2005).

Based on this overview, we propose a new method for the detection of bomb craters. We use the insights of Urbach & Stepinski (2009) and Kruse et al. (2019), namely that the extraction of crater candidates can be limited based on the appearance of craters. As bomb craters are generally not characterised by a pair of highlight and shadow regions, like planetary craters often are, we resort to using a standard blob detector. Our main contribution is based on the combination of the approaches presented by Brenner et al. (2018) and Merler et al. (2005): We use a CNN for the classification of crater candidates, weighting the classification errors from false positives in order to compensate the imbalanced distribution of proposals in a realistic scenario.

3. METHODOLOGY

Our approach to detect bomb craters in historical aerial images consists of two steps, namely the localisation of blob-like objects and the classification of those objects as either *crater* or *background*. While the former step is carried out using a standard blob detector (Bradski, 2000), the latter makes use of a fine-tuned version of the *Inception ResNet V2* (Szegedy et al., 2017).

3.1 Extraction of proposals

As we have shown in section 2, the approaches for object detection in general and crater detection in particular make use of some kind of proposal technique, which is often based on a sliding window approach. We consider such an approach not to be optimal for our task because the dense extraction of proposals using a sliding window does not match the sparse distribution of bomb craters in historical aerial images. Therefore, we use the blob detector, which we assume to have two advantages compared to the sliding window: On the one hand, it lowers the number of extracted proposals, which should help to reduce the number of false positives reported in the related work. On the other hand, it limits the appearance of crater candidates to blob-like structures. This should help the

CNN to better differentiate between *crater* and *background* and thus improve the classification performance.

The blob detector presented in Bradski (2000) works as follows. In the first step, the input image is converted into several binary images by applying different thresholds to the grey values. The thresholding starts at a minimum B_{ThMin} and is increased by a stepsize of B_{ThStep} until the maximum B_{ThMax} is reached. Coherent blobs in the binary images are merged if the distance between them is smaller than $B_{DistMin}$. In the second step, these merged blobs are filtered according to their size (counting all pixels in one cluster), which has to be in the range of $B_{SizeMin}$ and $B_{SizeMax}$, as well as their circularity ($B_{CircMin}$), convexity ($B_{ConvMin}$) and inertia ratio ($B_{InertMin}$). The blob detector delivers the coordinates of the centre of each valid blob as well as its size.

From these blob coordinates we form proposals, whose bounding boxes' centres equal the centre of a blob. In order to include some context in the proposals, the size of the bounding box equals the size of the blob times a factor $B_{Context}$, which we chose to be $B_{Context} = 1.2$. We assume that this context will help to increase the classification performance.

3.2 Classification

The classification of the proposals as either *crater* or *background* is carried out using a CNN. The input of our CNN is an image with a size of 299x299 pixels. The first part of our network is a pre-trained version of *Inception ResNet V2*. We omit its last layer, formerly used for the classification in the ImageNet classification challenge (Russakovsky et al., 2015). To help the pre-trained network adapt its feature extraction to the classification problem at hand, we append two fully connected layers with 512 and 256 nodes, respectively. Both layers use the *Rectified Linear Unit (ReLU)* (Nair & Hinton, 2010) as an activation function. Additionally, we perform Batch Normalisation (Ioffe & Szegedy, 2015) after both layers. The final classification is carried out in the subsequent softmax layer, which consists of two nodes, one for the classes *crater* and *background*, respectively. The network architecture is illustrated in Figure 1.

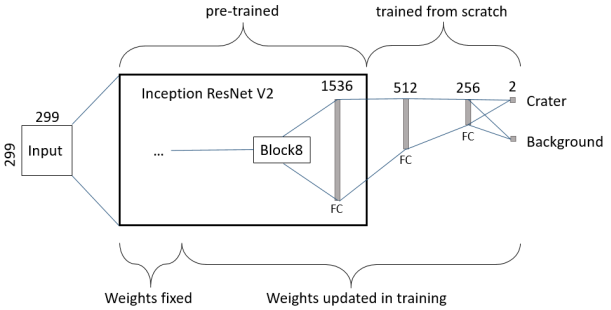

Figure 1. Architecture of our CNN. The input image is fed into a pre-trained version of *Inception ResNet V2*. The last two layers of the pre-trained network, i.e. 'Block8' and the FC-layer, are fine-tuned during the training. We append two fully connected layers to the pre-trained architecture, followed by a softmax layer.

3.3 Training

Training a CNN consists of minimising a loss function, which measures the classification error of the network. This classification error is based on the network's belief y_n^k that a sample $\mathbf{x}_n$ belongs to the class k, given the current network parameters $\mathbf{w}$. Using the softmax activation function (Bishop, 2006), the belief can be estimated as follows:

$$y_n^k(\mathbf{x}_n, \mathbf{w}) = \frac{e^{\bar{y}_n^k}}{\sum_\kappa^K e^{\bar{y}_n^\kappa}} \tag{1}$$

where $\bar{y}_n^k(\mathbf{x}_n, \mathbf{w})$ is the output of the last network layer. The result of the softmax activation can be interpreted probabilistically, such that

$$0 < y_n^k(\mathbf{x}_n, \mathbf{w}) < 1 \tag{2}$$

$$\sum_k y_n^k(\mathbf{x}_n, \mathbf{w}) = 1 \tag{3}$$

The network's total classification error E can now be calculated using the softmax cross entropy (Bishop, 2006):

$$E(\mathbf{w}) = \sum_n E_n(\mathbf{w}, \mathbf{x}_n) = -\sum_n \sum_k C_n^k \cdot ln(y_n^k) \tag{4}$$

where C_n^k is an indicator variable that equals 1 if the n-th sample belongs to the k-th class. In our special case of a two class classification problem we can rewrite equation 4 as

$$E(\mathbf{w}) = -\sum_n \left\{ C_n^1 \cdot ln(y_n^1) + C_n^2 \cdot ln(y_n^2) \right\} \tag{5}$$

Given that k=1 corresponds to the class *crater* and k=2 to the class *background*, equation 5 can be interpreted as the sum of errors caused by false negative (first term) and false positive (second term) classifications.

Based on the approach presented by Merler et al. (2005), we introduce the parameter γ_{FP} into Equation 5 to scale the error caused by false positive classifications. In order to avoid overfitting the network's weights to the training data, we use the regularisation in the form of *weight decay* (Bishop, 2006) so that our final loss function becomes

$$\begin{aligned} E(\mathbf{w}) = &-\frac{1}{N}\sum_{n=1}^{N} \left\{ C_n^1 \cdot ln(y_n^1) + \gamma_{FP} \cdot C_n^2 \cdot ln(y_n^2) \right\} \\ &+ \frac{1}{P}\sum_{p=1}^{P} w_p^2 \cdot \lambda_w \end{aligned} \tag{6}$$

where N is the number of samples in one training iteration, P is the number of parameters in the network ($\sim 6.2M$) and λ_w is a parameter to control the influence of the weight decay on the training. The goal of the training is to determine the network's parameters $\mathbf{w}$, which we achieve by minimising equation 6 using stochastic minibatch gradient descent (Bishop, 2006).

4. DATA

In order to test our proposed methodology, we make use of 45 8-bit greyscale images, stemming from two different sources: The Explosive Ordnance Disposal Service of Lower Saxony provided us with 18 images, as well as annotations for all 4187 craters in those images. The images were scanned with 1200 dpi, have different ground sampling distances between 13 cm and 57 cm and were taken in 1944/1945 over Northern Germany. The other 27 images as well as the respective 3250 annotations were provided by the 3D Optical Metrology Unit of the Bruno Kessler Foundation (FBK) in Trento, Italy. Unfortunately, no further information regarding the ground sampling distance or digitalisation is known for those images. For both sources, the references craters have been annotated manually in the images. The reference craters are given as square bounding boxes with known position and size. We note that the references for both sources include minor inaccuracies with respect to their size and centre, e.g. that the references are sometimes larger than the corresponding crater. Additionally, a coarse investigation of the Italian references indicated that not all craters in the images have been annotated. In Figure 2 and Figure 3, one example image for each source is shown. To make full use of the the radiometric resolution, we apply a Contrast Limited Adaptive Histogram Equalisation (Bradski, 2000) prior to experimental use of the images.

Figure 2. Example for an aerial image taken over Germany. The image has a ground sampling distance of 51 cm and was taken on 23.12.1944.

5. EXPERIMENTS

In this section we evaluate our proposed method using the aerial images presented in section 4. First, we introduce our experimental setup regarding the candidate extraction and the classification, as well as our evaluation strategy. After that we present and discuss the results of the experiments.

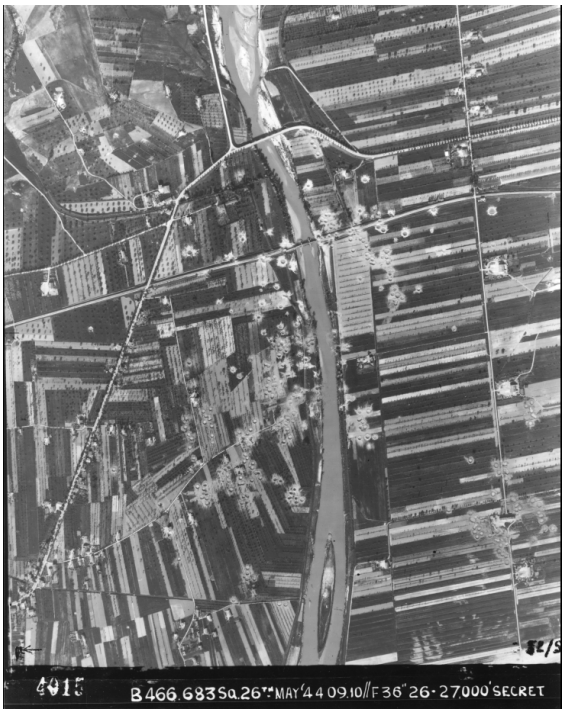

Figure 3. Example for an aerial image taken over Italy.

5.1 Setup

5.1.1 Candidate Extraction As mentioned in section 3.1 we use a simple blob detector for the extraction of candidates. We set the grey value thresholds to $B_{ThMin} = 0$ and $B_{ThMax} = 220$; we observed that a small ratio of reference craters has a bright appearance, which is why we also include larger grey values. We set the stepsize to $B_{ThStep} = 4$ and the merging distance to $B_{DistMin} = 10$ pixels. Based on an initial investigation of the sizes of the references, we set the minimal size to $B_{SizeMin} = 6^2 \cdot 2\pi$, corresponding to a blob radius of 6 pixels, and the maximal size to $B_{SizeMax} = 80^2 \cdot 2\pi$, corresponding to a blob radius of 80 pixels. We set the minimum circularity to $B_{CircMin} = 0$, the minimum inertia ratio to $B_{InertMin} = 0$ and the minimum convexity to $B_{ConvMin} = 0.3$.

5.1.2 Generation of training samples Based on the results of the crater candidate extraction (presented in section 5.2.1) we generate training samples to train and test our CNN. A blob is regarded as a positive sample, i.e. one of the class *crater*, if the intersection-over-union (IoU) between the blob and a reference crater is at least 35%. Candidates that do not fulfil this condition are regarded as negative samples, i.e. samples of the class *background*. We chose the IoU criterion because of the inaccuracies of the reference craters w.r.t. their size and position. If multiple candidates fulfil the condition towards one reference, all candidates will be regarded as positive samples. Using this approach, we generated 3,537 positive and 424,188 negative samples from the German images, which form the first dataset DS_G for the training of the CNN. From the Italian images, 2,169 positive and 235,764 negative samples were generated, which in turn form the second dataset DS_I. We also make use of data augmentation (Bishop, 2006): When accessing samples during training, validation or testing we additionally apply a random rotation of either $0°$, $90°$, $180°$ or $270°$, virtually increasing the amount of training data by a

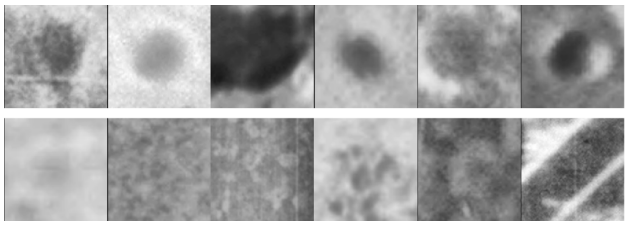

Figure 4. Positive (top row) and negative (bottom row) samples randomly drawn from dataset DS_G.

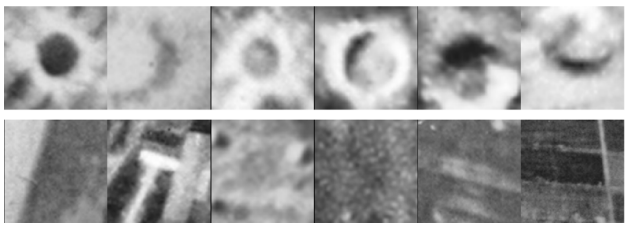

Figure 5. Positive (top row) and negative (bottom row) samples randomly drawn from dataset DS_I.

factor of 4. Figure 4 and Figure 5 show six examples from both datasets for both classes.

5.1.3 Classification For our training procedure, we scale all samples in both dataset to a size of 299 by 299 pixels, in order to be able to process them with a pre-trained *Inception ResNet V2*. We freeze the weights of all pre-trained layers except the last two. The weights of the three appended fully connected layers are initialised using Variance Scaling (He et al., 2015). Regarding the weight decay in Equation 6 we use a weight factor $\lambda_w = 10^{-2}$, as preliminary experiments showed that with smaller weight factors, i.e. $\lambda_w = 10^{-3}$ or $\lambda_w = 10^{-4}$, worse results were achieved. We train with a batchsize of $N = 350$ samples. The weight update is carried out using the Adadelta algorithm presented by Zeiler (2012), using a learning rate of 1. In preliminary experiments we also tested the Adam Optimizer (Kingma & Ba, 2014), resulting in worse F1-Scores compared to the Adadelta Optimizer. We train the CNN for 2500 iterations. We carry out specific experiments using a 9-fold cross validation, splitting the positive samples into 7/9th of the data for training, and 1/9th for training and validation, respectively. As described above, the datasets include more negative samples than positive samples. We assume that samples of the negative class have a larger variability. For the CNN to learn that variability, we use one third of the negative samples for training, validation and testing, respectively. The validation step in a cross validation iteration is carried out using 10 positive and 1200 negative samples, as this ratio of samples resembles a realistic distribution. We assume that using only a small fraction of the validation set (instead of all validation samples at once), the validation will contribute to a better generalisation, as validating on all samples at once will probably result in a network adapted to the validation data. Testing is carried out with all positive test samples and 120 times as many negative samples from the test dataset, as this ratio of samples resembles a realistic distribution (cf. section 5.1.2). In some experiments the samples for training and testing are stemming different datasets, so we do not carry out those experiments using cross validation. In those cases we instead use 80% of the positive samples of one dataset (e.g. DS_G) for training and the other 20% for validation, while the negative samples are again split equally. The testing is then carried out using all of the positive and negative samples of the other dataset (e.g. DG_I).

We carry out two sets of experiments. In the first set, we show the impact of the false positive factor γ_{FP} on the achieved test results. In the second set of experiments, we investigate the transferability between the two datasets DS_G and DS_I by comparing the results of training on one and testing on the other dataset or combining both datasets.

5.1.4 Evaluation For the evaluation of the crater candidate extraction we use the recall R:

$$R = \frac{TP}{TP + FN} \tag{7}$$

where TP are true positive detections, i.e. blobs that have an IoU of at least 35% with a reference crater, and FN are false negative detections, i.e. reference craters that do not have an IoU of at least 35% with a blob. We also report the total number of blobs N_B:

$$N_B = TP + FP \tag{8}$$

where FP are false positive detections, i.e. blobs that do not have an IoU of at least 35% with a reference crater.

For the evaluation of the classification we also use the recall R as well as the precision P and the harmonic mean of the two measures, i.e. the F1-score:

$$P = \frac{TP}{TP + FP} \tag{9}$$

$$F1 = 2 \cdot \frac{P \cdot R}{P + R} \tag{10}$$

where TP are true positive classifications, FP are false positive classifications and FN are false negative classifications. A sample is classified as a crater if the network's output y_n^1 is above 0.5.

5.2 Results and Discussion

5.2.1 Candidate Extraction As described in section 5.1.1 we use rather weak restrictions for the form filters, i.e. the circularity, convexity and inertia ratio. Preliminary experiments showed that using stricter restrictions (e.g. $B_{CircMin} = B_{InertMin} = B_{ConvMin} = 0.7$), which we assumed to be suitable given the appearance of craters in the images, in fact resulted in small number of blobs as well as a low recall of about 12%. We therefore omitted our initial assumption and resorted to weak form filters.
We applied the blob detector to the German as well as the Italian images. With the chosen set of parameters, a total of $N_B = 427,733$ candidates were extracted from the 18 German images. From all 4187 reference craters, a total of 3314 had an IoU of at least 35% with a blob, which corresponds to a recall of $R = 79.1\%$.
A total of $N_B = 237,948$ candidates were extracted from the 27 Italian images. From all 3250 references, a total of 2040 had an IoU of at least 35% with a blob, which corresponds to a recall

of $R = 62.7\%$. For comparison, if we were to apply a sliding window approach comparable to the one presented in Merler et al. (2005), we would end up with about $2,448,000$ candidates for the German and $602,000$ candidates for the Italian images. Depending on the size of the images, the number of candidates would increase by a factor of 10. On the other hand, the sliding window approach would obtain all crater references as candidates, which is beneficial for this kind of approach.

The results show that, although we used weak restrictions regarding the form filter, we still end up with rather low values for the recall, especially regarding the comparison with the sliding window approach. We assume that the low recall is caused by the fact that not all craters have a clear blob-like appearance, as can be seen in figures 2 and 3. Some craters exhibit a shadowed and a highlighted region inside their cone, others seem to have no clear edge. Additionally, the inaccuracies of the reference craters might still contribute to the low recall, as the blobs may match the craters but not the reference.

5.2.2 Classification In the first set of classification experiments we show the impact of the false positive weight on the classification. To this end, we train one CNN with $\gamma_{FP} = 1$ and another CNN with $\gamma_{FP} = 120$, where the latter choice corresponds to the ratio of the positive and negative samples generated from the German images. We train and test both CNNs on the German dataset DS_G using cross validation as described above. We report the averaged F1-Score over all cross validation iterations in Table 1.

	Prec. [%]	Rec. [%]	F1 [%]
$\gamma_{FP} = 1$	38.6	**77.6**	51.6
$\gamma_{FP} = 120$	**77.7**	56.5	**65.4**

Table 1. First set of experiments: Precision, recall and F1-score for differing false positive weights γ_{FP}.

The results from the first set of experiments show that by using a false positive weight corresponding to the use-case ratio of positive and negative samples, the F1-score can be improved by a large margin of 14%. As was to be expected, the recall drops for $\gamma_{FP} = 120$. Nevertheless, the improvement in the precision makes for a suitable trade-off, resulting in an increased F1-score.

Based on this insight, we set $\gamma_{FP} = 120$ for the following, second set of experiments. With this set we want to investigate if the datasets DS_G and DS_I can be combined to improve the classification of crater candidates. To this end, we carry out five experiments, combining the datasets in different ways. In the first two experiments we want to assess our network's ability to correctly classify data from one dataset only. We therefore train and test a CNN only on DS_G and another one only on DS_I. In the next two experiments, we want to assess the transferability between the two datasets. We train a CNN only on DS_G (this includes validation) and test it on DS_I, and vice versa. Note that we do not use cross validation in those two experiments. The last experiment investigates if the combination of the two datasets is beneficial for the classification. We combine the samples of DS_G and DS_I to one larger dataset DS_C and carry out the training as described for the first two experiments, i.e. the positive samples of both sets are combined, shuffled and chosen for training, validation or testing; the negative samples are handled similarly, according to the protocol presented in section 5.1.3. The results of the second set of experiments are shown in Table 2.

#	Train	Test	Prec. [%]	Rec. [%]	F1 [%]
1	DS_G	DS_G	77.7	56.5	65.4
2	DS_I	DS_I	66.0	58.5	62.0
3	DS_G	DS_I	54.2	39.0	45.3
4	DS_I	DS_G	85.0	18.2	29.9
5	DS_C	DS_C	67.2	48.0	56.0

Table 2. Second set of experiments: Results for various dataset combinations.

The first two experiments of the second set show that our proposed method achieves better results on DS_G than on DS_I, albeit by a small margin of 3% in the F1-score. This might indicate that our network architecture is better suited for the former dataset, or that distinguishing the samples from the latter dataset is more difficult in general. The large difference of 12% for the precision of experiment 1 and 2 can be explained by the observation that not all craters in the Italian images have been annotated as references. This might lead to blobs in the Italian images being extracted as negative samples, although the blobs, and hence the samples, do in fact resemble a crater. Therefore, some negative test samples might actually show a crater and would hence be classified as positive, resulting in more false positive classifications and, thus, in a lower precision.

Comparing the results of experiments 2 and 3, we see that the precision drops by about 12% when training on DS_G instead of DS_I, carrying out the test on DS_I. We assume that this drop on the one hand indicates that the two datasets have some substantial differences, and that the drop may on the other hand be again explained by the inaccuracies of the references. The drop of the recall of about 20% can be explained by the ratio of positive and negative samples in the Italian dataset DS_I. The number of samples reported in section 5.1.2 correspond to a ratio of about 1:109, whereas the training was carried out under the assumption that this ratio is 1:120. We assume that this caused the CNN to classify more positive samples as negative, thus resulting in more false negative classifications and consequently in a smaller recall.

Comparing the results of experiments 1 and 4 shows that the precision increases by 7% and the recall decreases by 38% when training on DS_I instead of DS_G, carrying out the test on DS_G. We assume that the CNN in experiment 4 had the tendency to generally classify more test samples as negative, as this explains the increase of the precision (more negative classifications can explain less false positives) as well as the decrease of the recall (more negative classifications can explain more false negatives). This might again be caused by the inaccuracies in DS_I; we assume that the CNN learned to classify some samples, which in fact resemble a crater but have a negative label, as negative. This can lead the CNN to classify some positive samples as negative, thus increasing the number of false negatives.

The last experiment 5 shows that training and testing on DS_C achieves considerably worse results compared to the single datasets in experiments 1 and 2. Although the results seems to indicate that our CNN cannot transfer its knowledge from one dataset to the other (experiment 3 and 4) and that the combination of both datasets is not beneficial (experiment 5), it is unclear whether this is caused by a low transferability between the two sets of images (i.e. craters and background in German images have a substantially different appearance than those in Italian images) or whether the inaccuracies of the reference craters are causing the smaller F1-score.

6. CONCLUSION

In this paper, we present an approach for the automatic detection of bomb craters in historical aerial images. Our approach consists of a crater candidate extraction using a simple blob detector and the classification of those candidates using a CNN.

The results of the candidate extraction showed that our approach was not able to achieve the required recall, i.e. too few reference craters were extracted as candidates. The comparison with the sliding window approach showed that we are able to reduce the total number of candidates, although the decrease was nearly negligible for the Italian images, especially with respect to the low recall. This indicates that our chosen set of parameters is not well suited for the given task. We assume that this can on the one hand be explained by the inaccuracies of the labels, but on the other hand also by the observation that not all craters have a circular, convex and compact appearance. As the examples for the samples showed, some craters have shadows in their cones, which we did not incorporate in our parameter choice for the blob detector. In future work, further experiments w.r.t. to the parameter choice are needed. We assume that using multiple sets of parameters might be beneficial, as different appearance variations could be considered in that way. Alternatively, the candidate extraction could be included into the CNN, e.g. using the Faster R-CNN architecture presented by Ren et al. (2015).

The experiments regarding the impact of the false positive weight showed that using a weight corresponding to the ratio of positive and negative samples is beneficial for the classification of samples with a realistic distribution, compared to using a unit weight. Nevertheless, further experiments are needed to test the sensitivity of the parameter and whether it was in fact chosen optimally with γ_{FP} corresponding to the ratio of generated samples. The experiments regarding the transferability also indicated a drawback of the false positive weight, namely that it is generally not known when applying a trained CNN to new data. Because historical aerial images can exhibit between very few and more than one thousand craters, a CNN trained with a specific false positive weight may not be suitable for the classification of crater candidates from one image. Instead, the CNN would have to be applied to candidates from a whole set of images. Assuming that the images we used represent a good generalisation, the ratio of positive and negative candidates might then match the chosen false positive weight of 1:120. Applying the classification to candidates from previously unseen aerial images in order to obtain test results for the detection remains to be done in future work, but we expect those results to be on par with the classification results.

The experiments regarding the transferability between the two sets of images, i.e. the two datasets DS_G and DS_I, showed that our CNN is not capable of transferring knowledge from one dataset to the other. We assume that this is on the one hand caused by the inaccuracies of the references and on the other hand by the differences between the samples of the two datasets. The impact of the apparent differences could be alleviated by the use of Domain Adaption (Wang & Deng, 2018), as the CNN could then learn a common representation for samples from both datasets, which could thus improve the classification ability of the CNN.

ACKNOWLEDGEMENTS

This project was financially supported by the EU project ”innoVation in geOspatiaL and 3D daTA - VOLTA” funded under the Marie-Curie RISE scheme as no. 734687. The authors would like to thank the State Office for Geoinformation and Surveying of Lower Saxony and its Explosive Ordnance Disposal Service as a department of the Regional Directorate Hamelin-Hanover as well as the 3D Optical Metrology unit of the Bruno Kessler Foundation (FBK) in Trento, Italy for providing the data to this project.

REFERENCES

Bishop, Christopher M., 2006. *Pattern Recognition and Machine Learning*. 1^{st} edn, Springer, New York (NY), USA.

Bradski, G., 2000. The OpenCV Library. *Dr. Dobb's Journal of Software Tools*.

Brenner, S., Zambanini, S., Sablatnig, R., 2018. Detection of bomb craters in wwii aerial images. *Proceedings of the OAGM Workshop*, 94–97.

Cohen, J. P., Lo, H. Z., Lu, T., Ding, W., 2016. Crater detection via convolutional neural networks. Proceedings of the 47th Lunar and Planetary Science Conference, Abstract 1143.

Emami, E., Bebis, G., Nefian, A., Fong, T., 2015. Automatic crater detection using convex grouping and convolutional neural networks. *International Symposium on Visual Computing*, 213–224.

He, K., Zhang, X., Ren, S., Sun, J., 2015. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. *Proceedings of the IEEE International Conference on Computer Vision*, 1026–1034.

Huang, G., Liu, Z., Van Der Maaten, L., Weinberger, K. Q., 2017. Densely connected convolutional networks. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 4700–4708.

Ioffe, S., Szegedy, C., 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *Proceedings of The 32nd International Conference on Machine Learning*, 448–456.

Kingma, D. P., Ba, J., 2015. Adam: A method for stochastic optimization. *3rd International Conference on Learning Representations (ICLR 2015)*.

Krizhevsky, A., Sutskever, I., Hinton, G.I., 2012. ImageNet classification with deep convolutional neural networks. *Advances in Neural Information Processing Systems 25 (NIPS'12)*, 1, 1097–1105.

Kruse, C., Rottensteiner, F., Heipke, C., 2019. Marked Point Processes for the automatic detection of bomb craters in aerial wartime images. *Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci.*, XLII-2/W13, 51-60.

LeCun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W., Jackel, L. D., 1989. Backpropagation applied to handwritten ZIP code recognition. *Neural Computation*, 1 (4), 541–551.

Merler, S., Furlanello, C., Jurman, G., 2005. Machine learning on historic air photographs for mapping risk of unexploded bombs. *International Conference on Image Analysis and Processing*, 735–742.

Nair, V., Hinton, G.E., 2010. Rectified linear units improve restricted boltzmann machines. *Proceedings of the 27th international conference on machine learning (ICML-10)*, 807–814.

Ren, S., He, K., Girshick, R., Sun, J., 2015. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in Neural Information Processing Systems (NIPS)*, 91–99.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A. C., Fei-Fei, Li., 2015. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115 (3), 211–252.

Schickler, W., 1995. Ein operationelles Verfahren zur automatischen inneren orientierung von Luftbildern. *Zeitschrift für Photogrammetrie und Fernerkundung (3/1995)*, 115–122.

Sermanet, P., Eigen, D., Zhang, X., Mathieu, M., Fergus, R., LeCun, Y., 2014. OverFeat: integrated recognition, localization and detection using convolutional networks. ICLR 2014.

Sharif Razavian, A., Azizpour, H., Sullivan, J., Carlsson, S., 2014. Cnn features off-the-shelf: an astounding baseline for recognition. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 806–813.

Szegedy, C., Ioffe, S., Vanhoucke, V., Alemi, A. A., 2017. Inception-v4, inception-resnet and the impact of residual connections on learning. Thirty-First AAAI Conference on Artificial Intelligence.

Urbach, E. R., Stepinski, T. F., 2009. Automatic detection of sub-km craters in high resolution planetary images. *Planetary and Space Science*, 57 (7), 880–887.

Wang, M., Deng, W., 2018. Deep visual domain adaptation: A survey. *Neurocomputing, 312*, 135–153.

Zeiler, M. D., 2012. Adadelta: an adaptive learning rate method. arXiv preprint arXiv:1212.5701.