Robust CNNs for detecting collapsed buildings with crowd-sourced data

Matthew Gibson[1] Dhruv Kaushik[1] Arcot Sowmya[1]

[1] University of New South Wales, Australia
{matthew.gibson1,dhruv.kaushik}@student.unsw.edu.au
sowmya@cse.unsw.edu.au

Technical Report
UNSW-CSE-TR-201806
October 2018

UNSW
SYDNEY

School of Computer Science and Engineering
The University of New South Wales
Sydney 2052, Australia

Abstract

Wildfires are increasingly common and responsible for widespread property damage and loss of life. Rapid and accurate identification of damage to buildings and other infrastructure can heavily affect the efficacy of disaster response during and after a wildfire. We have developed a dataset and a convolutional neural network-based object detection model for rapid identification of collapsed buildings from aerial imagery. We show that a baseline model built with crowdsourced data can achieve better-than-chance mean average precision of 0.642, which can be further improved to 0.733 by constructing a new, more robust loss function.

1 Introduction

Remote sensing is an important and widely used tool for disaster response in urban areas. There have been many studies to automate the assessment of natural disasters such as earthquakes [5], floods and wild fires using machine learning and computer vision techniques. Automatic assessment of building damage following natural disasters has been attempted across a variety of platforms from UAV to satellite, and with different modalities such as optical and SAR imagery. This can supplement traditional techniques such as ground-based field surveys which can be slow and dangerous for personnel, and provide a rapid response that is an essential part of disaster management.

Wildfires (termed bushfires in Australia) can cause large loss of human life and substantial property damage. The worst natural disaster in recent Australian history was the Black Saturday wildfires in the state of Victoria in 2009 which destroyed over 3500 homes and killed 173 people with an estimated cost of 3.3 billion USD, with more recent events being the 2017 Northern Californian wildfires and the 2018 wildfires in Attica in Greece. The likelihood of an increasing number of wildfires and severity is high due to expansion of the wildland-urban interface [9] and the potential for climate change to prolong the wildfire season.

Building collapse is a severe form of structural damage with no common scale for assessment. Vertical imagery from aerial and satellite platforms can prove useful, especially for wildfires in urban areas that often incinerate residential buildings. Remote sensing imagery can detect roof damage, burn mark outline, a debris curtain and sometimes severe facade destruction.

The leading methods for object detection are based on convolutional neural networks (CNNs). In this paper we outline an approach that allows quick training of CNNs to detect and approximately localise collapsed and infact buildings in urban areas. We use crowd-sourcing to annotate a large number of images quickly, transfer learning from standard CNN architectures, and substitution of the standard CNN loss function with a more robust alternative that allows effective use of noisy, crowd-sourced data. This has the potential for deployment in near real-time and to allow improvement of first response times, thereby improving outcomes for disaster relief.

We first review the state-of-art and discuss contributions in subsections IA and B. In section II, we provide an in-depth description of the Single Shot Multibox Detector (SSD) model for object detection along with a precise description of the changes to make it more robust to label noise. The experimental methodology is in section III, results and discussion in section IV and concluding remarks in section V .

1.1 Related Work

Application of CNNs is an active area of research in both computer vision and remote sensing [18]. Object detection has been used in remote sensing for several purposes including: counting wildlife on the African savannah [14], oil tanks [17], and vehicles [16]. The domain differences between remote sensing and computer vision have led to the development of novel CNN-based techniques such as rotation-invariant CNNs [3]. The work most closely related to ours [12] performs object detection with crowd-sourced data for disaster response

applications, although they focus on animals rather than buildings. A review of recent work on object detection in very high resolution optical imagery is available [4].

Object detection is becoming increasingly common in remote sensing alongside more traditional tasks such as scene or pixel-level classification [18]. Many enumeration and tracking tasks do not require complete pixel-level classification, and object detection models can be easier to develop because only localisation information is required for training. Contemporary object detectors are almost exclusively CNN based and typically come from two families: the first are so-called region proposal methods such as region-proposal CNN [6] and second are the single shot object detectors. Prominent examples of the latter are SSD [11], and RetinaNet [10]. In this paper we use the highly competitive SSD method.

Crowdsourced data often suffers from noise although this can be ameliorated through careful data collection [8]. Early work using crowd-sourcing for object localisation includes the LabelMe semantic segmentation dataset which proved essential prototyping the process of obtaining labels for the large object detection datasets that underpin deep learning research, such as ImageNet and MSCOCO. Research in remote sensing using voluntary workers such as in disaster response particularly earthquakes [1], has been discussed [12].

1.2 Contributions

We show that annotations for disaster imagery can be quickly crowd-sourced in a manner suitable for training a CNN-based object detector; an object detector can be effectively trained for detecting collapsed buildings using transfer learning with a small amount of data; and the object detector loss function can be modified for robustness, to take better advantage of crowdsourced annotations and boost performance.

2 Methodology

The SSD detector is a fully convolutional network [11] consisting of a standard image classification CNN used primarily as a feature extractor (usually VGG-16 trained on ImageNet), then a succession of 7 new multiscale convolutional layers which output a single tensor. The input image is split into parts and each part allows multiple potential objects. The potential objects in each part are called anchor boxes that are initialised with default values. The output tensor $(c+4) \cdot k \cdot m \cdot n$ encodes possible locations of objects of each class, where c is the number of classes, k the number of anchor boxes, and m and n the number of horizontal and vertical partitions. Finally non-maxima suppression is applied to the predictions. SSD is typically very fast [11].

There are several types of noise common in crowdsourcing, including problems with registration of the bounding box, omission of a bounding box, or an incorrect label on the bounding box. There are two common strategies to deal with these types of label noise: either design a more complicated model that can account for the noise generating process, or design a more robust model. There are several examples of both e.g. [15] and [13]. We focus on the latter strategy of increasing model robustness.

Bounding box prediction is a structured prediction task. Given an object category p, the jth predicted bounding box l_i, and the ith ground truth bounding box g_j, let x_{ij}^p be a $0-1$ variable which indicates whether l_i matches with g_j. The loss function for SSD is a weighted sum of two loss functions [11]:

$$L_{total}(x, p, l, g) = \frac{1}{N}\left[(L_{conf}(x, p) + \alpha L_{loc}(x, l, g)\right].$$

where $L_{conf}(x)$ is the confidence loss, $L_{loc}(x)$ the localisation loss, N the number of anchor boxes, and α a tuneable parameter set to 1. The confidence loss $L_{conf}(x)$ is a softmax loss given by:

$$L_{conf}(x, c) = -\sum_{i \in Pos} x_{ij}^p \log(\hat{c}_i^p) - \sum_{i \in Neg} \log(\hat{c}_i^0)$$

where $\hat{c}_i^p = softmax(c_i^p) = \frac{exp(c_i^P)}{\sum_p c_i^p}$, Pos is the set of matched boxes, Neg is the unmatched ground-truth boxes and c_i^p is the probability estimate of class p.

The localisation loss $L_{loc}(x))$ is the Huber or smoothed L_1 regression loss over the residuals between the predicted bounding box l and the ground truth bounding box g:

$$L(x, l, g) = \sum_{i \in Pos}^{N} x_{ij}^k smooth_{L_1}(l - g)$$

where

$$smooth_{L_1}(x) = \begin{cases} 0.5x^2 & \text{if } |x| < 1 \\ |x| - 0.5 & \text{otherwise.} \end{cases}$$

The Huber loss is the L_2 loss for small values of x and smoothly transitions to the L_1 loss when $x > 1$. This means that Huber loss is less sensitive to outliers in the data.

We modify the standard SSD loss function to make it more robust to the types of noise encountered in crowdsourcing, specifically omission noise on labels and measurement noise on the bounding box sizes. We change the regression loss from the smoothed L1 loss to the Tukey biweight loss [2]. We replace the function softmax with the "softmax with self-training" or "softmax with bootstrapping" [13], given by:

$$L_{softconf} = \sum_{i \in Pos} \left(\beta x_{ij}^p + (1 - \beta) c_p^i\right) log(c_p^i)$$

where β is yet another tuning parameter which we set to $\beta = 0.9$. This term replaces the term $\sum_{i \in Pos} x_{ij}^p \log(c_i^p)$ in the definition of L_{conf}. This modified loss function attempts to enforce consistency in predictions by evaluating samples as a convex combination of the predictions and the true labels. The model is therefore penalised less by omitted labels in the ground-truth data.

We modify the localisation loss by replacing $smooth_{L_1}$ with a loss function even more robust to outliers, the Tukey biweight loss, given by

$$L_{tukey}(x) = \begin{cases} k^2/6(1 - (1 - \frac{x}{k})^2)^3 & \text{when } |x| \leq k \\ k^2/6, & \text{otherwise.} \end{cases}$$

where k is a constant typically set to $k = 4.685$ where it produces 95% statistical efficiency when the errors are normally distributed. The Tukey loss assigns zero weight to errors beyond $k = 4.685$ when backpropagating errors through the network.

3 Experiments and Results

3.1 Dataset

The region of interest is a $1.2km^2$ region in the Western Australian regional town of Yarloop. The bushfire began on January 6, 2016 and lasted for 17 days. Photogrammetric RGB data was captured on January 13, 2016 at +/- 15cm ground resolution from a camera array on an aerial platform by the Nearmap company. The data was processed into mosaics of orthorectified images by Nearmap and a smaller subsection of the mosaics were selected as the region of interest, as depicted in Figure 3.1. The mosaics were then subdivided into 200 tiles of size 900×800. Of these, 124 contain objects of the relevant type, with 192 instances of destroyed buildings and 144 instances of intact buildings. Approximately 80% of this was used for training and 20% for validation. A further 120 tiles of the same area were collected for use as holdout test data. These tiles contained 84 objects, of which 40 were destroyed buildings and 44 were intact buildings.

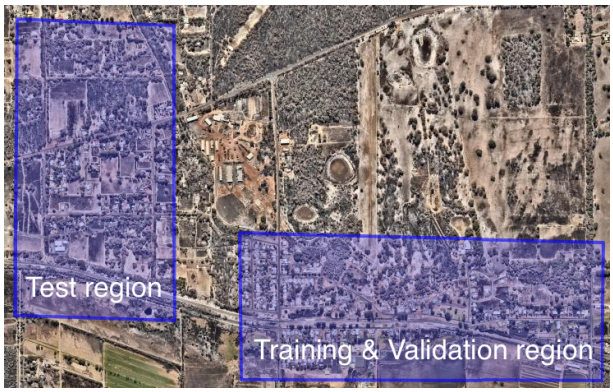

Figure 3.1: The region of interest in the larger photomosaic of the town of Yarloop used for object detection task.

The data was annotated with labelled bounding boxes by two groups: workers from the AWS Mechanical Turk crowd-sourcing platform, and two students with some imagery analysis experience. The crowd workers were given brief instructions, but had no prior experience on remotely sensed images. Each image in the training data was annotated by 3 crowd workers. In total 162 unique workers were involved, with each worker annotating on average 9 labels. All 600 image annotations were collected in approximately 1.5 hours. Examples of the worker annotations are shown in Figure 3.2 The annotations were screened for low quality but none were found and all worker annotations ended up being used.

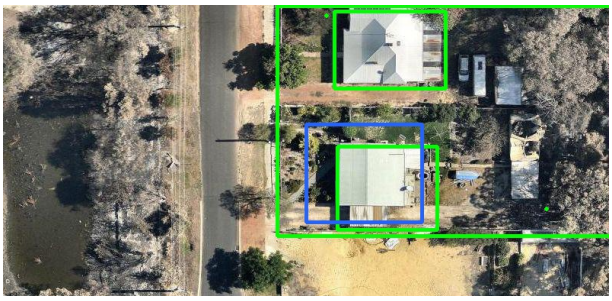

Figure 3.2: An example of annotations produced by 3 crowd workers for a single image, blue annotations for collapsed buildings, green for intact buildings.

3.2 Model training

The model was trained using ADAM [7] with an initial learning rate of 10^{-3}, which was learning rate slowly lowered to 10^{-5} for at most 100 epochs. We stopped training early after 8 epochs if there was no change to loss function at the lowest learning rate. Because of the small data size we performed numerous data augmentations including random adjustment of hue, saturation, contrast and brightness, geometric transformations of the images such as rotations and reflections, as well as random crops. Training took place on a server with a GeForce GTX Titan X 12GB graphics card, with Intel Xeon E5-2620 processor and 32GB of RAM.

Figure 3.3: An example of model predictions for the object detection task.

We took a robust approach to aggregating the data from noisy sources by making three copies of each image and associating one set of annotations from each of the annotators with a copy. We compared this against a gold standard of annotations performed by the second group, and also against sampling one annotation per image to simulate the effect of a single annotator.

Transfer learning, or domain adaptation, has proven useful in tackling the problem of insufficient data for training. This involves building a model on domain A where data is plentiful and subsequently training it on a small amount of data on domain B. We can achieve good model performance in situations where it is not possible to train a CNN from scratch. This approach is used in this work.

Experimental metrics in object detection differ from those used in image level classification or semantic segmentation. Commonly used metrics include

average precision (area under the precision recall curve), and mean average precision. An object is labelled as true positive if the intersection over union (IOU) score of the two bounding boxes is greater than some threshold (typically 0.5). Due to class imbalance in image detection datasets, average precision and recall are calculated for each image and averaged over all the images in the dataset. Finally, the mean average precision (mAP) is calculated as the average over all object classes. These metrics are preferred because the notion of a true negative is algorithm dependent.

We consider the following variants in our experiments: "Gold" refers to the vanilla SSD detector trained on the gold standard data, "Crowd" refers to an SSD detector trained with all the crowd annotations and redundant images, and "Random Crowd" refers to sampling random annotations for each image from the crowd. "Geom" refers to whether the geometric transformations were included in addition to the photometric augmentations, "Boot" refers to whether the bootstrap cross-entropy replaces the usual cross-entropy, and "Tukey" refers to whether the Tukey biweight loss replaces the smoothed Huber loss.

4 Results and discussion

Ablation test results on the holdout Yarloop test images are in Table 4. Despite

Model	***mAP***	***AP damaged***	***AP intact***
Gold + Geom	0.686	0.674	0.698
Gold	0.582	0.476	0.687
Crowd + Tukey + Boot + Geom	**0.733**	**0.696**	**0.77**
Crowd + Boot + Geom	0.705	0.646	0.764
Crowd + Geom	0.648	0.59	0.706
Crowd	0.642	0.596	0.642
Random Crowd	0.475	0.435	0.515

Table 4.1: Overview of ablative testing results on Yarloop dataset.

the very small data size, transfer learning proved to be quite effective even with a base model with as many parameters as VGG-16. We were not able to directly train even a shallow 7 layer CNN model on our dataset to an acceptable level of performance. Of the changes that we tested, strongest gains resulted from using the bootstrap cross-entropy instead of vanilla cross-entropy. We suspect that this performance boost would also extend to the gold standard data, although we have not tested this. It was surprising that crowd-sourced data was able to beat the performance of gold standard data. Occasionally when using the Tukey loss, the model rapidly diverged within 20 iterations in the first epoch, which has been observed before [2].

The geometric transformations were helpful on the gold standard data and provided an important boost. Interestingly they do not appear as helpful on the crowd-sourced data. In future work it might be suitable to either drop geometric transformations or render them redundant by building rotation invariance into the model as outlined elsewhere [3]. This would be helpful as the geometric augmentations slow down model convergence.

5 Conclusion

In this work, we have studied the utility of a CNN-based object detector trained on a small dataset with noisy, crowd-sourced labels. We show that by modifying the SSD loss function to include some well-known robust alternatives, we can construct a model that is more tolerant to omission and registration of bounding boxes derived from crowd annotations. Our experiments validate this approach and suggest that crowd-sourced imagery may be fruitfully used for other applications in remote sensing that use high resolution optical imagery. While crowd-sourcing allows rapid annotation of large quantities of data, there are non-trivial complexities in applying it effectively. Crowd-sourcing instructions need to be very clear and verified with either gold standard annotations or machine learning based screening techniques. Nevertheless, CNN-based object detectors can be made remarkably robust to label noise from crowd-sourced data.

6 Acknowledgements

The first author has been supported by an Australian Government Research Training Program (RTP) Scholarship. We gratefully acknowledge the use of Nearmap imagery in this work, and thank crowd workers for their contributions.

Bibliography

[1] Luke Barrington, Shubharoop Ghosh, Marjorie Greene, Shay Har-Noy, Jay Berger, Stuart Gill, Albert Yu-Min Lin, and Charles Huyck. Crowdsourcing earthquake damage assessment using remote sensing imagery. *Annals of Geophysics*, 54(6), January 2012.

[2] Vasileios Belagiannis, Christian Rupprecht, Gustavo Carneiro, and Nassir Navab. Robust Optimization for Deep Regression. pages 2830–2838, 2015.

[3] G. Cheng, P. Zhou, and J. Han. Learning Rotation-Invariant Convolutional Neural Networks for Object Detection in VHR Optical Remote Sensing Images. *IEEE Transactions on Geoscience and Remote Sensing*, 54(12):7405–7415, December 2016.

[4] Gong Cheng and Junwei Han. A survey on object detection in optical remote sensing images. *ISPRS Journal of Photogrammetry and Remote Sensing*, 117:11–28, July 2016.

[5] Laigen Dong and Jie Shan. A comprehensive review of earthquake-induced building damage detection with remote sensing techniques. *ISPRS Journal of Photogrammetry and Remote Sensing*, 84:85–99, October 2013.

[6] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. pages 580–587, 2014.

[7] Diederik P. Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. *arXiv:1412.6980 [cs]*, December 2014. arXiv: 1412.6980.

[8] Adriana Kovashka, Olga Russakovsky, Li Fei-Fei, and Kristen Grauman. Crowdsourcing in Computer Vision. *Foundations and Trends in Computer Graphics and Vision*, 10(3):177–243, 2016.

[9] H. Anu Kramer, Miranda H. Mockrin, Patricia M. Alexandre, Susan I. Stewart, and Volker C. Radeloff. Where wildfires destroy buildings in the US relative to the wildlandurban interface and national fire outreach programs. *International Journal of Wildland Fire*, 27(5):329–341, June 2018.

[10] T. Lin, P. Goyal, R. Girshick, K. He, and P. Dollar. Focal Loss for Dense Object Detection. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2999–3007, October 2018.

[11] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C. Berg. SSD: Single Shot MultiBox Detector. In Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling, editors, *Computer Vision ECCV 2016*, Lecture Notes in Computer Science, pages 21–37. Springer International Publishing, 2016.

[12] Ferda Ofli, Patrick Meier, Muhammad Imran, Carlos Castillo, Devis Tuia, Nicolas Rey, Julien Briant, Pauline Millet, Friedrich Reinhard, Matthew Parkan, and Stphane Joost. Combining Human Computing and Machine Learning to Make Sense of Big (Aerial) Data for Disaster Response. *Big Data*, 4(1):47–59, February 2016.

[13] Scott Reed, Honglak Lee, Dragomir Anguelov, Christian Szegedy, Dumitru Erhan, and Andrew Rabinovich. Training Deep Neural Networks on Noisy Labels with Bootstrapping. In *CoRR, abs/1412.6596,*, 2014.

[14] Nicolas Rey, Michele Volpi, Stphane Joost, and Devis Tuia. Detecting animals in African Savanna with UAVs and the crowds. *Remote Sensing of Environment*, 200(Supplement C):341–351, October 2017.

[15] Filipe Rodrigues and Francisco Pereira. Deep learning from crowds. 2018. arXiv: 1709.01779.

[16] Tianyu Tang, Shilin Zhou, Zhipeng Deng, Huanxin Zou, and Lin Lei. Vehicle Detection in Aerial Images Based on Region Convolutional Neural Networks and Hard Negative Example Mining. *Sensors*, 17(2):336, February 2017.

[17] L. Zhang, Z. Shi, and J. Wu. A Hierarchical Oil Tank Detector With Deep Surrounding Features for High-Resolution Optical Satellite Imagery. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 8(10):4895–4909, October 2015.

[18] X. X. Zhu, D. Tuia, L. Mou, G. Xia, L. Zhang, F. Xu, and F. Fraundorfer. Deep Learning in Remote Sensing: A Comprehensive Review and List of Resources. *IEEE Geoscience and Remote Sensing Magazine*, 5(4):8–36, December 2017.