CS230

Building Detection in Satellite Images - Improving Resource Distribution in Rohingya Muslim Refugee Camps

Aprotim C. Bhowmik
Electrical Engineering
Stanford University
abhowmik@stanford.edu

Nichelle A. Hall
Computer Science
Stanford University
nhall2@stanford.edu

Minh-An Quinn
Computer Science
Stanford University
minhan@stanford.edu

Abstract

With the United States government promulgating xenophobic ideologies and refusing to assist refugees, it is of great importance for us to acknowledge and stand up against the persecution that several populations face. Rohingya Muslims, an ethnic group in the Myanmar-Bangladesh region, is one such group that has been forced to flee to refugee camps. In an effort to assist these camps, we propose the implementation of a building detection algorithm based on the Mask R-CNN architecture with a ResNet backbone. Herein, we show that a model based on crowdAI's work can perform robust building detection in satellite images of Rohingya refugee camps.

1 Introduction

Across the world, people have been forced to flee their country because of persecution, war, violence, or natural disaster. These refugees are often without homes and resources to care for themselves. In an attempt to diminish the struggles of refugees, many governing bodies have orchestrated the formation of refugee camps, which are meant to provide temporary safety, shelter, and resources to those in need. The maintenance of these camps, however, can be a challenge. With large populations of refugees moving in/out of camps, allocating shelter and resources can be difficult. With some governing bodies (e.g., the United States) eschewing humanitarian efforts and reducing funds, the financing of and workforce for refugee camps must be carefully managed.

In an effort to support the maintenance of these camps, we are working with UNICEF (specifically with representative Eric Boucher) to create an open-source algorithm to estimate the size and population of refugee camps. The first step of this task is building detection, which will help with mapping refugee camps. Given a satellite image, our model will annotate all the buildings in the image.

We build a general model of building detection and then fine tune the hyperparameters to satellite images of Rohingya refugee camps. Rohingya Muslims, an ethnic group with origins in Myanmar and Bangladesh, is widely known as the most persecuted minority group in the world. Their allegiance to Islam and their ethnic status in their countries are only part of the reason for the deep-seated oppression that they face, which is exacerbated by the strikingly low news coverage on the subject. We hope not only that our building detection model helps with resource distribution but also that our "air-time" regarding the Rohingya people brings more attention and action to this issue.

CS230: Deep Learning, Spring 2018, Stanford University, CA. (LateX template borrowed from NIPS 2017.)

2 Related Work

We researched 4 Convolutional Neural Network (CNN) models during our design process: R-CNN, Fast R-CNN, Faster R-CNN and Mask R-CNN. Since Mask R-CNN is the model that we implemented, we elaborate on that model in our "Methods" section.

R-CNN performs basic object detection by drawing bounding boxes around objects in an image. R-CNN uses Selective Search, an algorithm that loops through the image with windows of different sizes, to create bounding boxes. Selective Search groups together adjacent pixels by texture, color, or intensity to identify objects. Once the region proposals are made by Selective Search, R-CNN passes all bounding box dimensions through AlexNet and a Support Vector Machine (SVM) to classify the object inside the bound box. Finally, R-CNN passes each box through a linear regression model to output tighter coordinates for the box once the object has been classified.[2,4]

Fast R-CNN makes a few changes to the R-CNN model to improve its performance. The major downfall of R-CNN is its slow speed; training three models – the CNN (AlexNet), the classifier (SVM), and the linear regression model – makes the pipeline hard to train. Additionally, R-CNN requires each proposed region to be run through AlexNet independently. In an effort to optimize the R-CNN algorithm, Fast R-CNN uses RoIPool (Region of Interest Pooling) in conjunction with max pooling to process the entire image (and all proposed regions) in one pass of the CNN. All these regions can be done in one pass because many of the regions have overlaps and can therefore share learned features. Fast R-CNN also combines all three models into a single model by adding a softmax layer after the CNN to predict the class (instead of the SVM model) and a linear regression layer in parallel with the softmax layer to output the tight bounding box coordinates.[1,4]

Faster R-CNN is an even faster update to Fast R-CNN, which gets rid of the slow Selective Search process used by Fast R-CNN. Faster R-CNN makes use of the features learned during the the forward pass of the CNN to propose regions. This is done by creating a Region Proposal Network, which adds a Fully Convolutional Network on top of the CNN. This Region Proposal Network takes in the CNN feature map and outputs a bounding box for each anchor box and a score for each bounding box representing the probability that it contains an object. This is done by passing a sliding window over the CNN feature map and outputting bounding boxes and scores at each window. Additionally, Faster R-CNN uses a ResNet as its convolutional backbone architecture.[3,4,5]

Lastly, we researched Mask R-CNN, which is explained in more detail in the "Methods" section.

3 Dataset and Features

We trained several iterations of our model to see which one worked best with images of Rohingya refugee camps. We first trained an initial building detection model based on on a dataset acquired from crowdAI, "an organization that connects data science experts and enthusiasts with open data to solve specific problems". Our training dataset consists of 280,741 tiles (as 300 x 300 pixel RGB images) of satellite images, along with their corresponding annotations in MS-COCO format. Our validation set contains 60,317 tiles, and our test set contains 60,697 tiles. Each tile is labeled with annotations such as bounding box dimensions, area, image category, and a binary value describing if the image contains a crowd. An annotation corresponds to each building found in the tile, and as a result, there are several annotations per tile. All data preprocessing was performed by crowdAI.

In later stages of our model, in an attempt to fine tune our hyperparameters and reduce our loss, we used segments of the validation set based on image category to better cater to satellite images of Rohingya refugee camps. We made several smaller validation sets, each of which focused on one qualitative aspect of building detection (e.g., more roads, roof tiles similar to those in refugee camps, building shape). We intentionally focused on aspects of the images that might be important in Rohingya refugee camps, such as building shape and roof tiles.

As explained in more detail below, we found quite similar results for all validation sets, implying (1) that the initial building detection model is quite robust, and/or (2) our images were too similar to result in significant changes to our hyperparameters.

In addition to testing our model on the annotated images from crowdAI, we want to ensure that our model works well on Rohingya refugee camps, as that was the goal of our project. We obtained updated satellite images of these camps from OpenAerialMap.org, a website that provides public

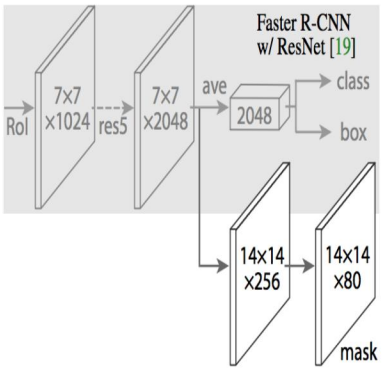
access to such data. However, a challenge of our project was that these images were not annotated. We had to process these images ourselves by resizing them to 300 x 300 pixels and annotating parts of these images ourselves. Given our very limited time constraints, we were able to hand annotate 800 of these camp images. The lack of availability of annotated camp images are such a big problem that in fact, a large number of satellite image websites for refugee camps are attempts to get the public to annotate the images.

Below is an example from our crowdAI dataset is provided below, with the satellite image and its corresponding annotation:


```
{
    "id":409190,
    "image_id":48034,
    "segmentation":[
        [
            94,
            78,
            32,
            78,
            32,
            25,
            94,
            25,
            94,
            78
        ]
    ],
    "area":3286.0,
    "bbox":[
        32,
        32,
        62,
        62
    ],
    "category_id":100,
    "iscrowd":0
}
```


4 Methods

Our model uses a vanilla Mask R-CNN architecture. Mask R-CNN is an improvement on Faster R-CNN because it enables instance segmentation. While Faster R-CNN is a framework for object detection (which classifies and localizes objects in an image), instance segmentation combines object detection and semantic segmentation (i.e., semantic segmentation classifies each pixel of an image into a category). In order to do this, Mask R-CNN makes two major changes to Faster R-CNN. The first change is that in addition to predicting the class and bounding coordinates of the box, Mask R-CNN outputs a binary mask for each region of interest (RoI). The network generates a binary mask for every class using a Fully Convolutional Network, which determines whether each pixel in the RoI belongs to an object. The second major change in Mask R-CNN is replacing RoIPool with a new layer called RoIAlign. RoIPool extracts a small feature map for each RoI. To do this, RoIPool rounds floats to discrete numbers, resulting in a misalignment when doing semantic segmentation. Instead of rounding, RoIAlign fixes the misalignment issue by using bilinear interpolation to compute more exact input feature values.[3,4]



Similar to the original Faster-CNN, our starter Mask R-CNN implementation uses a vanilla ResNet backbone. When training our baseline model, we start by training the network head; this part is used for bounding box classification and regression. Our baseline then trains the ResNet backbone and fine tunes all layers together. By using Mask R-CNN, our baseline is able to get pixel-level accuracy for building detection.[3,4]

The loss functions that we used for this architecture are nontrivial, as each R-CNN architecture builds on its ancestors. Let us start with the multi-task loss of an image:

$$L(\{p_i\},\{t_i\}) = \frac{1}{N_{cls}}\sum_i L_{cls}(p_i, p_i^*) + \lambda\frac{1}{N_{reg}}\sum_i p_i^* L_{reg}(t_i, t_i^*) \tag{1}$$

Here, i represents the anchor index of an image, p represents the probability of an anchor being an object, t represents the parametrized bounding box, N_{cls} is the mini-batch size, N_{reg} is the number of anchor locations, and the asterisk denotes the ground-truth label. Let us delve a bit deeper:

$$L_{cls}(p, u) = -\log p_u \tag{2}$$

$$L_{reg}(t_u, v) = \sum_i smooth_{L_1}(t_i^u - v_i) \tag{3}$$

$$smooth_{L_1}(x) = \begin{cases} 0.5x^2 & |x| < 1 \\ |x| - 0.5 & otherwise \end{cases} \tag{4}$$

Here, u is the ground-truth class for the RoI, v is the regression target for bounding boxes for the RoI, and $smooth_{L_1}$ is an operator for L1 loss that is less sensitive to outliers than normal L1 loss.

5 Results and Discussion

Our first goal was to use crowdAI's model to build a general network for building detection using the given train, validation, and test sets. Using pre-trained weights in this network allowed us to start with an already-robust method of building detection. Below are two images from CrowdAI that have been labeled with building locations:

As aforementioned, we then took segments of the validation set based on image characteristics to better tune our model to Rohingya refugee camps. We experimented with various combinations of learning rate, momentum, mini-batch size, and weight decay. Each combination was assessed based on the above loss functions, and despite settling on the following values, our models performed similarly with most small modifications:

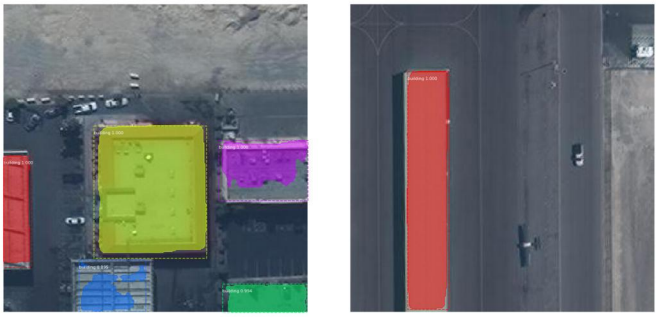
Learning Rate	0.001
Momentum	0.9
Mini-Batch Size	5
Weight Decay	0.0001
IoU Cutoff	0.50

We also tried different numbers of iterations and epochs, which again yielded similar results each time. Our loss values varied only slightly from one model to another, with our final losses after training being 0.6121 (general loss, eq. 1), 0.0112 (class loss, eq. 2), 0.0198 (bounding box loss, eq.

3). With such a large data set, running several models–even on Google Colaboratory and even with subsets of our data–was rather onerous. This is something we take into consideration in the section below. We ended up with a model achieving 69.7% precision and 47.9% accuracy on the crowdAI test set.

Our final step was to apply our model to images of Rohingya refugee camps. Below are two such images obtained from OpenAerialMap. It is immediately visible that the images–or at least, the buildings in the images–resemble closely the buildings in images from crowdAI.

Finally, below are two examples of labeled images of Rohingya refugee camps. We were able to get a precision of 93.2% and a recall of 87.9%. We can see that the accuracy is quite good, and this is likely because (1) the original building detection model is quite robust, and (2) many of the images of Rohingya refugee camps look rather similar to the train/validation/test images provided by crowdAI. In fact, we believe our model actually performs better on the Rohingya camp images because the majority of the camp builings were more boxy and uniform in shape. However, despite its high success rate, our model struggled to identify buildings that were (1) similar in color to the surrounding grass/environment, (2) seemed to have less distinguishable boundaries just with the naked eye, and (3) were either small and between more distinguishable buildings or were at the edges of the image. The images below show our model's output on real Rohingya camp photos, both on more densely packed buildings, as shown on the left image, as well as more sparse images of with buildings on the periphery, as shown on the right image.



6 Future Work

We have noted above that crowdAI's general building detection model has worked robustly for images of Rohingya refugee camps. We plan to annotate more images of Rohingya camps to allow for more fine tuning, as the availability of such images is low. After improving our model, we hope to work with UNICEF to make this algorithm open-source and accessible to those who are running these refugee camps. It is quite exciting how relatively simple neural networks can provide so much information, especially when that information can affect the well-being of refugees.

Eric Boucher, our contact at UNICEF, already has multiple projects that relate to ours (e.g., mapping roads, estimating population sizes). We hope that our project contributes to UNICEF's larger goals of helping refugees, and to that end, perhaps some of this baseline neural network structure can be applied to similar problems such as mapping roads.

7 Code

All updated code has been pushed to the following repository: https://github.com/NichelleBot/CS230-Final-Project.

8 Contributions

Our project has been divided as equally as possible among all of us. Aprotim reached out to our UNICEF contact (Eric Boucher) to develop our specific project topic. Nichelle followed up by securing our dataset and getting examples of code to for training/testing. Minh-An coordinated the logistics of our meetings and ensured that we followed a specific timetable for our project work.

We worked together to research our models, process our dataset, develop code for training, and run initial model parameters to ensure that our code runs. We also read background papers (referenced below) about Mask R-CNN and its lineage. And we worked together to use OpenAerialMap to find and analyze images of Rohingya refugee camps.

References

[1] Girshick, R. B. (2015). Fast R-CNN. *Computing Research Repository*.

[2] Girshick, R. B., Donahue, J., Darrell, T., & Malik, J. (2013). Rich feature hierarchies for accurate object detection and semantic segmentation. *Computing Research Repository*.

[3] He, K., Gkioxari, G., Doll, P., & Girshick, R. B. (2017). Mask R-CNN. *Computing Research Repository*.

[4] Parthasarathy, D. (2017). A brief history of CNNs in image segmentation: From R-CNN to Mask R-CNN. *Computing Research Repository*.

[5] Ren, S., He, K., Girshick, R. B. & Sun, J. (2015). Faster R-CNN: Towards real-time object detection with region proposal networks. *Medium - Athelas*.