BuildingNet
Building Detection from Satellite Imagery

Jasmine Hu
Department of
Electrical Engineering
Stanford University
jxhu@stanford.edu

Manan Rai
Department of
Computer Science
Stanford University
mananrai@stanford.edu

Vivian Wong
Department of Civil and
Environmental Engineering
Stanford University
vwwong3@stanford.edu

Abstract

A major challenge for humanitarian organizations is monitoring changes of refugee camp locations. A building detection model using satellite imagery can provide an automated mean of detecting updated footprint of buildings. This project aims to evaluate, improve and compare the performance of two state-of-the-art models commonly used for image segmentation: Mask R-CNN and U-Net. We experimented with different convolutional backbones for the Mask R-CNN and with deeper connectivity in the U-Net. We find that even with 4 epochs of training, our best model surpasses the performance of our baselines, and achieves comparable results as most existing work.

1 Introduction

In a world where wars and natural disasters render unassuming citizens as refugees, humanitarian support is called for at an alarming rate. In such a case, automated tracking of refugee camps is important for the dispersal of supplies and other forms of aid. This project is motivated by the hope to assist UNICEF in providing assistance to refugees all around the globe, by creating a model that can segment buildings from satellite images and thereby help them detect, segment, and optimize refugee camps, in turn helping them in their efforts to provide humanitarian support.

2 Related Work

The Mask R-CNN is well-known for semantic segmentation [2] and a source code using a ResNet as its backbone is available as the CrowdAI challenge baseline model [11].

The U-Net is another well-known network for efficient image segmentation [1,3] and a baseline code is available from the Kaggle DSTL Competition's 3rd place winner [12].

The idea of adding morphological processes stemmed from the Kaggle DSTL Competition's 1st place winner's interview [4], where the interviewee mentioned their strategy of dilating the training masks. The idea of adding morphological post-processing was inspired by Minerva ML Lab's entry to the 2018 Data Science Bowl Competition, where they added the watershed algorithm [10].

3 Dataset and Features

In order to build a comprehensive model, we used datasets from the CrowdAI Mapping Challenge (with zoomed-in satellite images) and the Kaggle DSTL Competition (with zoomed-out images)

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

[6][7]. The CrowdAI dataset includes RGB images with a resolution of 300 × 300 and MS-COCO format annotations. The training set includes 8366 images, while the validation and test sets include 1820 images each. The Kaggle DSTL includes RGB satellite images with a resolution of 3345×3358, annotated with MultipolygonWKT. The train set includes 25 images and the test set include 32 images. An example of an image from each dataset is shown in Figure 1 and 2.

Fig 1. Sample Image from the CrowdAI Dataset

Fig 2. Sample Image from the Kaggle Dataset

4 Methods

4.1 Mask R-CNN

The Mask R-CNN is a framework for effective semantic segmentation. It is an extension of the Faster R-CNN model which uses a Region Proposal Network with Non-max Suppression to identify bounding boxes for objects of interest, as well as predicts the class of the object. These are complemented by masks generated using an additional branch (composed of a Fully Convolutional Network) that conducts pixel-level segmentation.

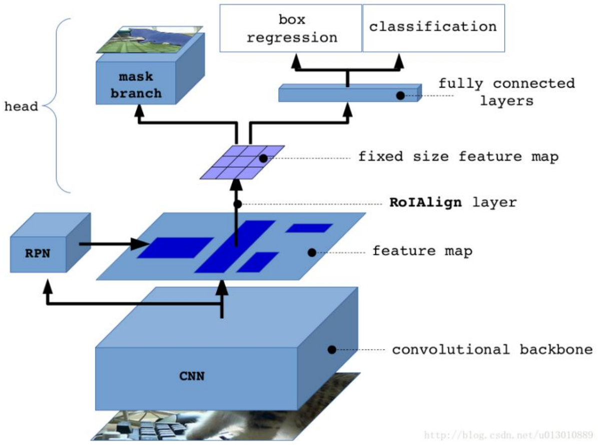


Figure 3. Mask R-CNN architecture.

It uses a combination of the class, mask, and bounding box losses as the total loss. A Smooth L1 Loss is used to regress the position of the bounding box; the mask prediction is based on a binary loss. This architecture typically uses a ResNet as its convolutional backbone. We also experimented with a DenseNet121 backbone.

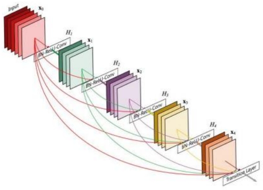


Figure 4. DenseNet architecture [14].

4.2 U-Net

The U-Net has 10 convolution layers. It has 5 contraction and 4 expansion blocks with 1 final convolution layer to map the output. Our loss function contains a cross-entropy of IoU probability in addition to the typical binary cross-entropy (eqn. (6) in [3]). For a typical U-Net, each contraction/expansion block has 2 repeated batch normalization, 3×3 convolutions, and exponential linear units (eLu). Each contraction block output is also concatenated to a corresponding expansion block.

We experimented with 2 variations: one inspired by the ResNet [13], we added local residual skip connections within each contraction/expansion block, and the other inspired by the DenseNet [14], we attempted concatenating all convolution outputs among the pairing contraction/expansion blocks. Due to memory limit, we only concatenated inside blocks, and from contraction blocks (before the 1 by 1 channel-reduction bottleneck layer) to the pairing expansion blocks.

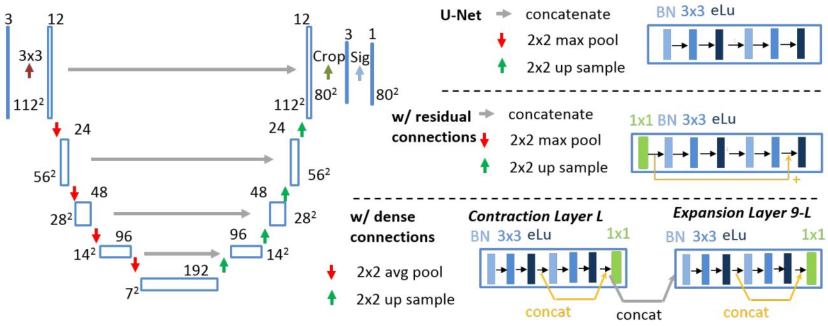


Figure 5. Illustration of the overall U-Net structure and layers within contraction/expansion blocks.

4.3 Morphological Operations

Furthermore, morphological operations were experimented in the pre-processing and post-processing parts of the model. Before training, buildings' ground truth masks were dilated, which is the enlargement of masks. This allowed the model to capture some buildings that it couldn't notice before. However, the immediate downside is that some building masks started to overlap, leading to an overall reduction of number of buildings detected[8].

We tried two post-processing methods to solve the problem. The first was with erosion, shrinking masks to reduce overlapping[9]. The second is the watershed algorithm, which separates overlapping objects by computing an image that is the distance to the background, then marking objects along the pixels with minimum distances. However, this algorithm has shredding effect and over-increased the detectable building numbers. Future work along the watershed algorithm could include tuning the watershed algorithm so that it can separate buildings with a higher tolerance.

5 Results and Evaluation

5.1 Hyper-parameters

Training was done on an NVDIA GeForce 1800 Ti GPU. Some tests were conducted on an AWS Ubuntu Deep Learning AMI EC2 instance. Figure 6 lists the hyper-parameters for the Mask-RCNN with a DenseNet backbone and shows the losses during 4 epochs of training.

Hyperparameter	Value(s)
Learning Rate	0.001
Weight Decay	0.0001
Learning Momentum	0.900
Mini-batch size	5
Steps per epoch	1000
Layers per Dense Block	[6, 12, 24, 16]

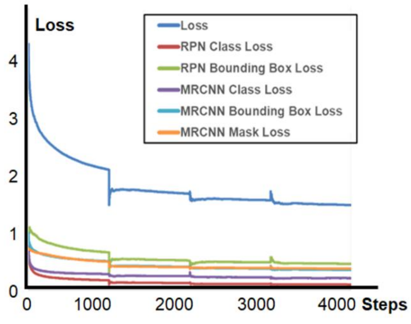


Figure 6. Hyper-parameters for the Mask-RCNN with DenseNet backbone and losses during training.

Figure 7 lists the hyper-parameters for the U-Net, and they are the same for the typical U-Net, model with residual connections, and model with dense connections. We tuned the hyper-parameter of the starting number of channels. 12 channels had a training IoU of 0.736 and took 166 ms to train per step. A typical starting channel number of 32 improved the IoU by 2.7% but took 2.25 times longer to run. Hence we stayed with 12 starting channels to test network structure additions.

Hyperparameter	Value(s)
Learning Rate	0.001
β_1	0.9
β_2	0.999
Mini-batch size	128
Steps per epoch	25
Contract / Expand Layers	5/4
Initial Number of Channels	12

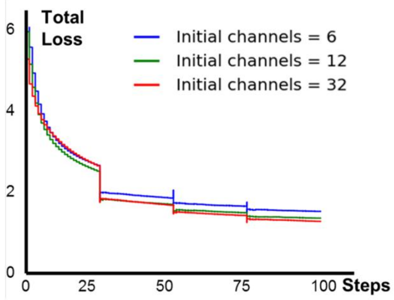


Figure 7. Hyper-parameters for the U-Net and training losses during hyper-parameter tuning.

5.2 Performance

The models' performances are evaluated quantitatively with the Intersection over Union (IoU) metric. Table 1 shows the IoU with various architectures trained over 4 epochs on the CrowdAI dataset. Qualitative performances can be observed from the predicted masks in Figures 8 and 9.

Model	Training IoU	Test IoU
Basic Mask R-CNN (trained over 40 epochs)	N/A	0.396
> w/ DenseNet	N/A	0.00023
Basic U-Net	0.792	0.765
> w/ res. con.	0.736	0.705
> w/ dense con.	0.803	0.788

Table 1. Model performances on the crowdAI dataset over 4 epochs of trainings.

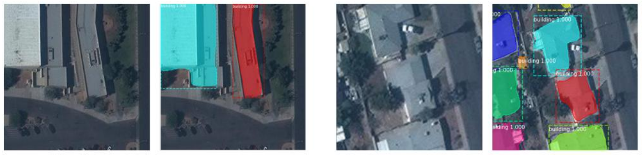


Figure 8. Sample CrowdAI images and predicted masks with the baseline Mask R-CNN model.

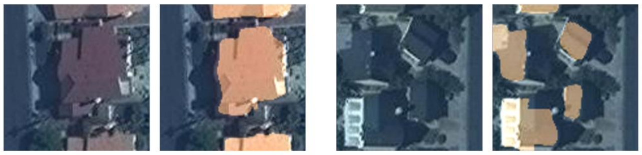


Figure 9. Sample CrowdAI images and predicted masks with U-Net with dense connections.

Tested on the Kaggle dataset, the IoU obtained by baseline U-Net, U-Net with residual connection, and U-Net with dense connections, are 0.6790, 0.7009, 0.7015. (This IoU is evaluated on the training set because the validation/test set data is no longer available to the public.) The IoU is reasonable compared to the training IoU of 0.7453 over 50 epochs with 16-band data.

5.3 Morphological Operations

Table 1 demonstrates the results on a sample image from the Kaggle dataset, with masks predicted by a U-Net with residual connections. The results showed the final predicted masks with each morphological operation. In the Watershed algorithm, the three masks shown respectively in the order of left to right are the originally predicted mask with overlapping buildings, the distances computed and the mask showing separated building masks.

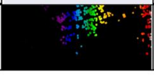
	Predicted Mask	# of Buildings Detected
No Morphology		121
Dilation in Pre-Processing		101
Dilation in Pre-Processing + Erosion in Post-Processing		103
Watershed (building separation)		625

Table 2. Example morphological operation results on U-Net predicted masks from Kaggle dataset.

The effects of dilation, erosion and watershed are more effective for satellite images where buildings are small and sparse. Therefore, these morphological operations were implemented on a model using the Kaggle dataset, where satellite images are more zoomed out.

5.4 Discussion

For the Mask R-CNN model, our Baseline with ResNet50 backbone, trained over 40 epochs, performs decently well. DenseNet121 as backbone gives comparable relative performance after 4 epochs.

All U-Net model performed well after 4 epochs of trainings, and the one with dense connection has a test IoU twice higher than that of basic R-CNN trained over 40 epochs. The U-Net performance varied based on the dataset: residual connection improved IoU on the Kaggle dataset, but worsened it on the CrowdAI dataset.

Dilation in pre-processing improves detection but overlaps buildings, whereas erosion in post-processing reduces overlap. Watershed can separate overlapped masks but requires additional tuning since it impairs the model's ability to classify buildings with uneven shapes as a single building, thereby resulting in a very high prediction of the number of buildings.

6 Conclusion

In conclusion, we experimented with the architectures of Mask R-CNN and U-Net to improve their performances on building detection from satellite imagery. We also experimented with morphological pre- and post-processing. The U-Net model with dense connection performed best, surpassing the baseline by a huge margin with just 4 epochs of training.

7 Future Work

We would like to explore using the U-Net as a backbone for the Mask R-CNN, combining the strengths of the two models. Other backbones that we considered as possible backbones but did not complete tests on include the VGG-16 and ResNeXt models, which may hold promise. The Mask R-CNN with a DenseNet121 backbone achieved comparable results over 4 epochs as the baseline, and training for longer may improve its performance. Training longer epochs on the U-Net model, and understand better how connections affect training can also be worth experimenting with. Finally we could improve the watershed algorithm to separate overlapping building masks with higher tolerance.

8 Code

The code is available at `github.com/mananrai/BuildingNet`.

9 Contributions

Manan Rai worked on preparing the CrowdAI dataset and dataset splits, and experimented with the Mask R-CNN, considering the DenseNet121, VGG-16 and ResNeXt backbones. Jasmine Hu worked on experimenting with various additions to the U-Net model including residual connections and dense connections, and extended the network functionality to take CrowdAI MS-COCO format data. Vivian Wong worked on experimenting with morphological operations on the U-Net model, and contributed to report writing and poster creation. All team members contributed to report writing, poster creation and poster presentation.

References

[1] Olaf Ronneberger, Philipp Fischer, Thomas Brox. U-Net: Convolutional NEtworks for Biomedical Image Segmentation. Retrieved from `arXiv:1505.04597`

[2] Kaiming He, Georgia Gkioxari, Piotr Dollar, Ross Girshick. Mask RCNN. Retrieved from `arXiv:1703.06870`

[3] Vladimir Iglovikov, Sergey Mushinskiy, Vladimir Osin. Satellite Imagery Feature Detection using Deep Convolutional Neural Network: A Kaggle Competition. Retrieved from `arXiv:1706.06169`

[4] Kaggle Team. (2017) Dstl Satellite Imagery Competition, 1st Place Winner's Interview: Kyle Lee. *The Official Blog of Kaggle.com.* Retrieved from `http://blog.kaggle.com/2017/04/26/dstl-satellite-imagery-competition-1st-place-winners-interview-kyle-lee/`

[5] Minerva-ML. (2018) Open Solution to the Data Science Bowl 2018. Retrived from `https://github.com/minerva-ml/open-solution-data-science-bowl-2018`

[6] Datasets from the CrowdAI Mapping Challenge. Retrieved from `https://www.crowdai.org/challenges/mapping-challenge/dataset_files`

[7] Dstl Satellite Imagery Feature Detection. Retrieved from `https://www.kaggle.com/c/dstl-satellite-imagery-feature-detection`

[8][9][10] scikit-image Morphology Module Documentation. Retrieved from `http://scikit-image.org/docs/dev/api/skimage.morphology.html`

[11] Crowdai-mapping-challenge-mask-rcnn. Retrieved from `https://github.com/crowdAI/crowdai-mapping-challenge-mask-rcnn`

[12] Vladimir Iglovikov, Sergey Mushinskiy, Vladimir Osin. Kaggle_dstl_submission. Retrieved from `https://github.com/ternaus/kaggle_dstl_submission.`

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun. Deep Residual Learning for Image Recognition. Retrieved from `arXiv:1512.03385`

[14] Gao Huang, Zhuang Liu, Laurens van der Maaten, Kilian Q. Weinberger. Densely Connected Neural Networks. Retrieved from `arXiv:1608.06993`