CS230

Building footprint extraction based on RGBD satellite imagery

Denis Ulanov* and Andriy Syrov†

Department of Computer Science, Stanford University

Abstract

Artificial intelligence is now used for maintaining accurate up-to-date maps to respond to humanitarian, ecological and disaster recovery challenges. These tasks require near real-time, accurate, automated mapping directly from aerial and satellite images. In this application project, we apply Mask-RCNN and Conditional Adversarial Network methods for building footprint segmentation. We approach this problem as supervised learning, apply data augmentation, experiment with learning parameters, transfer learning, leveraging multi-modal RGB-D data and achieve high accuracy results.

1 Introduction

Reliable, automated aerial and satellite image semantic segmentation and instance segmentation one of the most challenging in solving cartographic tasks. Significant manual work is still required to product accurate quality maps. In this project we explore state of the art Mask R-CNN [1] method for instance segmentation, and Conditional Adversarial Network image-to-image translation for semantic segmentation. We use well known USSOCOM Urgan 3D dataset [2] which includes not only RGB satellite images, but also surface and terrain models as input and both building instance segmentation and semantic segmentation as output. Our learning intent for this project is apply and customize for our task well known matterport [3] opensource implementation of Mask R-CNN. Another goal is to apply Conditional Adversarial Network [4]. This image-to-image method is more end-to-end approach, where instead on handcrafting loss functions (as in case of Mask R-CNN), the method is using adversarial loss, so that loss function is learned to improve generated building footprint segmentation and make the resulting map look real.

2 Related Work

Object detection requires localization of variable instances in the image. We can train single instance CNN classifier on cropped image and use sliding window to solve this task. This basic idea was gradually improved with R-CNN [5] Selective Search approach for region proposals where CNN classifier needs to be applied. Fast R-CNN [6] improves further by removing external Selective Search and getting regions of interest directly from image using CNN. This approach reused convolutional computation and was significant speed up. Output of this model softmax classifier and regressed bounding boxes. Faster R-CNN [7] further improves speed by adding dedicated RPN (region proposal

*dulanov@stanford.edu; Equal contribution Facebook, Menlo Park, California

†asyrov@stanford.edu; Equal contribution UHG (United Healthcare Group), Bellevue, Washington

CS230: Deep Learning, Winter 2020, Stanford University, CA. (LateX template borrowed from NIPS 2017.)

network). Mask R-CNN [1], which we apply in this project now extends Faster R-CNN by adding branch for prediction of object masks.

For semantic segmentation use of Conditional Adversarial Network [4] image-to-image translation can be used as good and simple alternative to Mask R-CNN with we also apply in this project.

Other approaches are Fully Convolutional Networks which can solve the problem, but the accuracy of this approach is quite low [8]. Fully Convolutional DenseNets is a further extension of Fully Convolutional Network approach [9]. Encoder-Decoder approach with Separable Convolutions is another extension of CNN which separates encoding/decoding into two deep CNNs [10].

3 Dataset and Features

USSOCOM Urban dataset consisting of satellite images of Tampa, FL and Jacksonville, FL [11]. Inputs to our models are RGB images, as well as two terrain height maps, one with the objects on surface and another without. Each image (tile) is a 2048 by 2048 with pixel ground sample distance of about 0.5 meters. Each tile may contain from zero up to hundreds building footprints.

The training set consists of 174 tiles, validation and test sets consist of 31 tiles each. Appendix A.3 includes examples of DSM surface images (height or depth map), DTM images (terrain model) and RGB photographs. We split each tile to 64 pieces, each of size 256 by 256 pixels. This significantly reduced GPU memory requirements, sped up the training process and allowed us to iterate faster on the model hyperparameter search, though reduced performance scores.

Images were already geometrically corrected. Depth images were rescaled to range [0,1] and missing data set to 0. To verify our preprocessing we developed image analysis and visualization utilities. Figure 9 shows result image histograms of RGB and Depth images and segmented with ground truth 3D result images. Human level performance results are included in Appendix A.2.

Augmentation applied at training stage: horizontal and vertical flip of images; scaling transform in range of 1.0 to 1.2 and up to $\pm 45^{\circ}$ rotation angle and 1.4 rage shear. Gaussian Noise layer is used for Conditional GAN.

4 Methods

In this project, we have explore use of Mask-RCNN architecture for instance segmentation and Conditional GAN approach for semantic segmentation.

4.1 Mask R-CNN

Mask R-CNN architecture is state-of-the art approach for instance segmentation. The model consists of:

- Backbone CNN for feature extraction (ResNet50) with Feature Pyramid Network on top
- Region Proposal Network to find anchor boxes
- ROI Classifier to propose regions-of-interest that may have classifying objects
- Segmentation Masks to identify objects masks.

As the model receives input image, it extracts features from it using multiple convolutional/BatchNorm/MaxPool layers from the backbone network. The next stage RPN uses the extracted features and anchors of different sizes and shapes to find proposals for foreground objects and for their approximate bounding boxes. The next stage ROI Classifier assigns proposed objects to one of the output classes (which is only one class in our case) and refines their bounding boxes. The last step Segmentation Mask applies multiple convolutional layers on top of proposals to generate object masks.

Each part of the model uses its own loss: L1 loss for bounding boxes and cross-entropy loss for instances and masks classifiers. The loss that training minimizes is a mean of all intermediate losses.

For baseline model, we used an open-source implementation of Mask-RCNN provided by Matterport [12]

4.2 Conditional Adversarial Network

Conditional Adversarial Network combines GAN objective with traditional L_1 loss. The discriminator is unchanged, except it classifies separately each of $N \times N$ (where N is a hyperparameter) grid patches from last convolutional layer. Then each output grid patch outputs if it is real building footprint patch or generated. Objective function of discriminator is then to minimize:

$$\mathcal{J}_D(x, y) = -0.5(\mathbb{E}_{x,y}[\log D(x, y)] + \mathbb{E}_x[\log(1 - D(x, G(x))]) \tag{1}$$

where x is RGB-D satellite image, y is ground truth building semantic map. In other words (1) finds mean over binary cross entropy losses of patches, first term for true footprint and second for generated by generator network.

Generator network task is to fool (maximize loss of) discriminator and also come close to ground truth image of building footprint. Therefore objective function of generator is to minimize:

$$\mathcal{J}_G(x, y) = -\mathbb{E}_x[\log D(x, G(x)] + \lambda \mathbb{E}_{x,y}[||y - G(x)||_1] \tag{2}$$

where first term is non-saturated generator loss for failing to "fool" discriminator and second term (factored by hyperparameter λ) is L_1 distance between generated and ground truth image (of building footprint), discriminator weights are not trained by this loss. L_2 distance is not recommended [4] as it causes to generate more blurry images.

4.2.1 Network Architecture Overview

Generator network is U-Net [13] type encoder-decoder network with skip connections between corresponding layers in encoder and decoder to help information flow between layers.

PatchGAN discriminator models image as Markov random field, which assumes patches which do not touch are independent. Therefore this approach can be viewed as texture style loss. Discriminator network consists of several conv-relu-batchnorm layers which gradually reduce initial image size to $N \times N \times 1$ patch layer output.

Discriminator with such architecture captures high-frequency image structures, and for low- frequency image structures correctness we then use L_1 distance term in generator loss (2).

5 Experiments & Discussion

5.1 Performance Metrics & Baseline

To identify positive detection, we calculate IoU score for each (prediction, label) mask pair and then determine which pairs have IoU score exceeding a threshold value of 0.5 [14]. After identifying positive/negative detections for each building footprint in validation set, we compute total precision and recall over all detections. With that, we compute F1 and use it as a single metric to evaluate model performance.

5.2 Mask R-CNN

5.2.1 Error analysis

To identify the direction of further improvement of the baseline Mask-RCNN model, we took 40 random image samples that contained misclassified buildings and categorized them as shown in Table 1. The analysis showed us the key categories of errors we need to focus on. We excluded the categories that are hard even for humans (building being on a border and buildings covered by trees) and focused on improving the rest of the top categories.

5.2.2 Experiments

The following improvements have been applied to the baseline model to increase the quality of the model for the building footprints extraction task:

Category	Error %	Category	Error %
Parking lot as building	27.5%	Road in trees as building	12.5%
Building on the border of image	22.5%	Large building as many buildings	5%
Non-rectangular building	22.5%	Many small buildings as one large	2.5%
Building covered by trees	22.5%	Diagonal building	2.5%
Roof of multiple colors	12.5%		

Table 1: Error analysis results

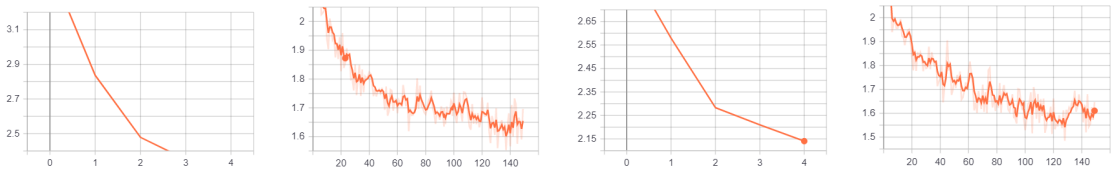

Figure 1: Left to right: Mask-RCNN losses on train set (top layers and full training) and on dev set (top layers and full training)

1. We have applied transfer learning from pre-trained MS COCO dataset, which provided us a reasonable set of weights to start training with. in training we run training of only top layers for a few epochs, and then run a longer training for the entire network.
2. We added depth filter from DSM surface model as 4th input channel. The signal carries an important information about height of the surface which is not captured in RBG channes. The signal should help the model to distinguish between buildings and parking lots/roads, which are the top misclassification examples
3. Similarly to depth filter, we also added DTM terrain model as 5th input channel. The improvement serves similar purpose as depth filter and give the model more input to distinguish between buildings and flat surfaces.
4. We applied data augmentation such as flipping fraction of existing training examples along vertical and horizontal axes, as well as rotation and scaling. Rotation in particular helps model to perform better on rare examples when a building is oriented diagonally.
5. Experimentation with mini-batch size, learning rate and the number of epochs leaded us to setting a mini-batch size to 128, to increase total number of epochs to 150 and also to set the learning rates to 0.001 for top layers training and to 0.0001 for all layers
6. We have run experiments on RPN network parameters and on detection thresholds, but they didn't yield any positive results

5.2.3 Training and results

Losses for top layers training and full training are displayed in figure 1. Model results on train/dev/test sets are finalized in Table 2. The results are consistent across datasets, the model doesn't suffer from overfitting. The results indicate that the final model performance on test set is only 3% away from human-level performance, which means that the quality of the model prediction is good and that the further improvements are expected to be difficult.

	Precision	Recall	F1	IoU
train	0.7734	0.7273	0.7496	0.6463
dev	0.7857	0.7127	0.7474	0.6506
test	0.7794	0.7278	0.7527	0.6411

Table 2: Results for Mask-RCNN

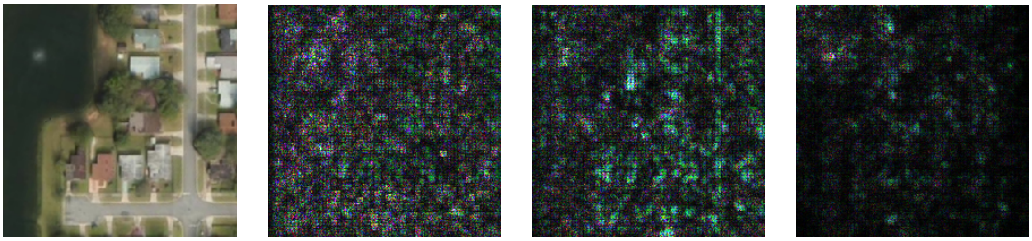

Figure 2: Left to right: original image, low-level convolutions, mid-level convolutions, RPN output

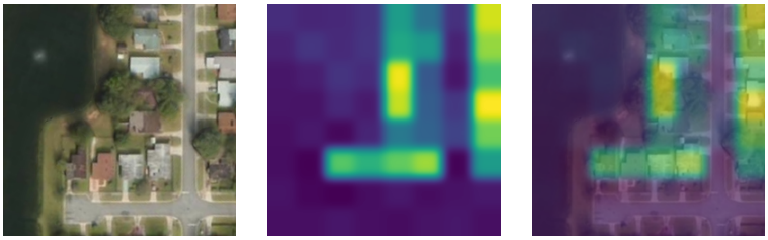

Figure 3: Heatmap generated using occlusion maps

5.2.4 Visualization of the convolution layers

In order to understand the sensitivity of intermediate layers of the model to the input and to gain insights on the types of transformations that our network is performing, we have used saliency maps analysis. We picked a few intermediate layers in backbone network and in RPN and gradient of the layer output with respect to every pixel of the input image [15]

Figure 2 shows the results of the visualization. Low-level convolutions capture low-level features like edges across the image, mid-level convolutions detect larger objects like roads, buildings and group of trees and the RPN output has high activations on areas that may contain a building.

5.2.5 Visualization of the results

In order to understand which areas of the image affect successful prediction of buidings, we applied occlusion maps analysis. We have sequentially covered input image with a black area 32x32 pixels and measured total IoU score across all true positive matches. Based on that we built heatmap that indicates which areas of the image affect total IoU score and which don't [16].

Figure 3 shows the resulting heatmap. The areas that don't affect predictions are over the roads and the water, whereas the areas over buildings do affect the scores, as expected.

5.3 Conditional Adversarial Network

Most of our decision for training Conditional GAN are due to reported to results for CGAN in [4] and results for U-Net [13]. Importantly, Conditional GAN works well on datasets smaller than ours, so we did not use any transfer learning from existing models and trained our networks from scratch. Also as with Mask R-CNN extensively used image augmentation: rotations, horizontal and vertical flips, zoom range $[0.7, 1]$ and shear range 1.2. Other important hyperparameters are $\lambda = 100$ (weight of L_1 distance in loss (2)) and patch grid size $N \times N = (16, 16)$. We use mini-batch (8 random images per mini-batch) SGD and Adam Optimizer training.

Metrics we used are per pixel precision, recall, F1 and IoU (intersect over union), which are also used in [4], [13]. Also, we separately calculated results for entire image (256×256) and inner area with 32 pixel border removed (192×192) to also evaluate quality of the model on partially cut off buildings.

5.3.1 Learning and Results

Initial values for parameters where set per [4]. Further we experimented with different patch size, weight of L_1 loss, learning rate and number of filters, image generation parameters, adding of Gaussian noise layer, and other. Most of parameters were not changed [4].

	Precision	Recall	F1	IoU
train	0.8662	0.7826	0.8127	0.6982
dev	0.8518	0.7676	0.7956	0.6781
test	0.8437	0.7670	0.7899	0.6703

Table 3: Results for Conditional Adversarial Network semantic segmentation

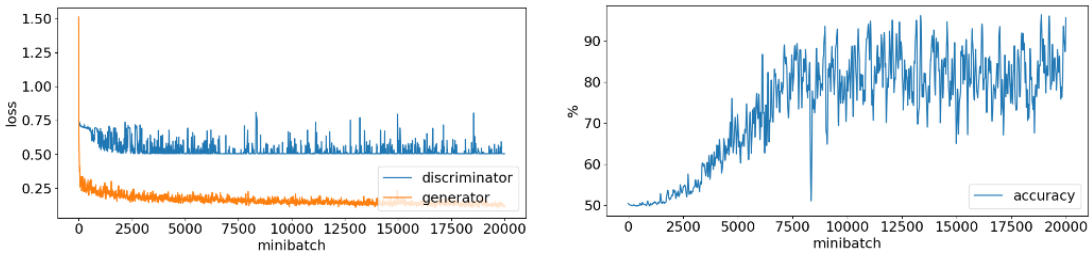

Figure 4: Left to right: Conditional GAN training losses, discriminator accuracy

Learning progress summarised in Figure 4[3]. Achieved result for Conditional Adversarial Network semantic segmentation are summarised in Table 3 are better than then human level results we described earlier, but still lest than other known results [4]. Examples of generated images and their corresponding ground truth images provided in Appendix A.1.

6 Conclusion & Future Work

As a result of our project, we implemented 2 models that recognize building footprints with the accuracy similar to human-level performance. The models are based on the state-of-the-art architectures Mask-RCNN and GAN. We performed error analysis and provided visualization on the internal and external model results.

Further improvements of the models are expected to be difficult since they already perform at human-level. If we had more time and more computational resources, we would try training the models on original uncut 2048x2048 images to avoid frequent misclassification of building partially appearing on image.

7 Contributions

Source code can be found at `https://github.com/asyrovprog/cs230project`

Our work for instance segmentation is based on open source Mask RCNN matterport library [12]. Provided in our github implementation Conditional Adversarial Network is derivative of [17]. The rest of the code is original. We closely worked together on all major modules and the paper: Andriy implemented data preprocessing; integration of dataset into Mask-RCNN model; depth filter to the model; data augmentation; 3D visualization of dataset; refactoring of Mask-RCNN code; implemented and trained baseline CGAN model; integrated dataset into CGAN model; implemented performance

[3] Generator loss scaled by factor $1/50$ since L_1 weight $\lambda = 100$

[4] Topcoder competition result F1 score 0.89 for semantic segmentation. There reasons for being behind top score are that we used smaller image size (and performance decreases when buildings are cut) and our model hyperparameters and architecture still have room for improvements.

metrics for CGAN model. Denis setup of baseline Mask-RCNN model on a toy example; configured AWS training/validation environment; implemented precision/recall/F1 metrics; performed error analysis; organized and performed human-level performance estimation; added terrain channel to the model; PRN hyperparameters experimentation; occlusion maps visualization saliency maps visualization.

Acknowledgements

We gratefully acknowledge Prof Andrew Ng, Kian Katanforoosh and their CS230 course assistants (CA's) at Stanford University for their devotion to optimizing our learning experience and their inspiring passion in deep learning research. Our project has also greatly benefited from our project mentor, Jo Chuang, who has been instrumental in guiding us to improve our work.

References

[1] He, K., Gkioxari, G., Dollár, P. & Girshick, R. Mask r-cnn (2017). URL `https://arxiv.org/abs/1703.06870`.

[2] Goldberg, H., Brown, M. & Wang, S. A benchmark for building footprint classification using orthorectified rgb imagery and digital surface models from commercial satellites. In *Proceedings of IEEE Applied Imagery Pattern Recognition Workshop 2017* (2017).

[3] matterport.com. Model zoo: Mask r-cnn for object detection and segmentation (2020). URL `https://modelzoo.co/model/mask-r-cnn-keras`.

[4] Isola, P., Zhu, J., Zhou, T. & Efros, A. A. Image-to-image translation with conditional adversarial networks (2016). URL `http://arxiv.org/abs/1611.07004`. `1611.07004`.

[5] Girshick, R. B., Donahue, J., Darrell, T. & Malik, J. Rich feature hierarchies for accurate object detection and semantic segmentation. *CoRR* **abs/1311.2524** (2013). URL `http://arxiv.org/abs/1311.2524`. `1311.2524`.

[6] Girshick, R. B. Fast R-CNN. *CoRR* **abs/1504.08083** (2015). URL `http://arxiv.org/abs/1504.08083`. `1504.08083`.

[7] Ren, S., He, K., Girshick, R. B. & Sun, J. Faster R-CNN: towards real-time object detection with region proposal networks. *CoRR* **abs/1506.01497** (2015). URL `http://arxiv.org/abs/1506.01497`. `1506.01497`.

[8] Shelhamer, E., Long, J. & Darrell, T. Fully convolutional networks for semantic segmentation (2016). URL `https://arxiv.org/abs/1605.06211`.

[9] Jégou, S., Drozdzal, M., Vazquez, D., Romero, A. & Bengio, Y. The one hundred layers tiramisu: Fully convolutional densenets for semantic segmentation (2016). URL `https://arxiv.org/abs/1611.09326`.

[10] Jégou, S., Drozdzal, M., Vazquez, D., Romero, A. & Bengio, Y. Liang-chieh chen and yukun zhu and george papandreou and florian schroff and hartwig adam (2018). URL `https://paperswithcode.com/paper/encoder-decoder-with-atrous-separable`.

[11] SpaceNet. Spacenet on amazon web services (aws) (2018). URL `https://spacenetchallenge.github.io/datasets/datasetHomePage.html`.

[12] Abdulla, W. Mask r-cnn for object detection and instance segmentation on keras and tensorflow. `https://github.com/matterport/Mask_RCNN` (2017).

[13] Ronneberger, O., Fischer, P. & Brox, T. U-net: Convolutional networks for biomedical image segmentation (2015). URL `http://arxiv.org/abs/1505.04597`. `1505.04597`.

[14] Jordan, J. Evaluating image segmentation models. (2018). URL `https://www.jeremyjordan.me/evaluating-image-segmentation-models/`.

[15] Kotikalapudi, R. Keras-vis documentation: Saliency maps. URL https://raghakot.github.io/keras-vis/visualizations/saliency/.

[16] Pal, S. A guide to understanding convolutional neural networks (cnns) using visualization (2019). URL https://www.analyticsvidhya.com/blog/2019/05/understanding-visualizing-neural-networks/.

[17] Isola, P., Zhu, J.-Y., Zhou, T. & Efros, A. A. Image-to-image translation with conditional adversarial networks. *CVPR* (2017).

A Appendix

Here we include supplemental figures and tables for better legibility.

A.1 Result Images for Mask-RCNN

Figure 5 shows how results are compared to ground truth. We can see that after first iteration performance of the model is not very good, a good number buildings are missing, there are overlaps in predictions, there are areas that are clearly misclassified like the island in the first sample. The last iteration results are much closer to ground truth, the model only misses complex cases like buildings covered by trees.

A.2 Result Images for Conditional Adversarial Network

Figure 6 shows how results are compared to ground truth. We can see that after first iteration performance of the generator is not very good (column a, row 4). But at last iteration the model predicts buildings reasonably well. We can notice that some building are missed because they are entirely (or almost entirely) covered by trees. We can also notice that when building are not covered by trees the model does, in some cases, more reasonable segmentation than specified by ground truth.

A.3 Human Level Performance

To determine human level performance we analysed 20 random images by 3 volunteers, who are not cartography experts. Instance segmentation was done on RGBD images where depth buffer was converted to gray scale image. The group of volunteers correctly recognized 202 buildings and had 41 false positives and 70 false negatives. Based on this results, we got recall as 0.743, precision as 0.831 and F1 score as 0.78. Figure 7 demonstrates challenges for the volunteers, where it is hard to determine why marked in red areas do not have buildings per ground truth. The majority of the errors are caused by buildings being on the border on the image and by buildings being covered almost entirely by trees.

A.4 Dataset and Features

Figure 8 shows example of one tile from resulting dataset. This tile consists of 256×256 images. Figure 9 shows result image histograms of RGB and Depth images and segmented with ground truth 3D result images.

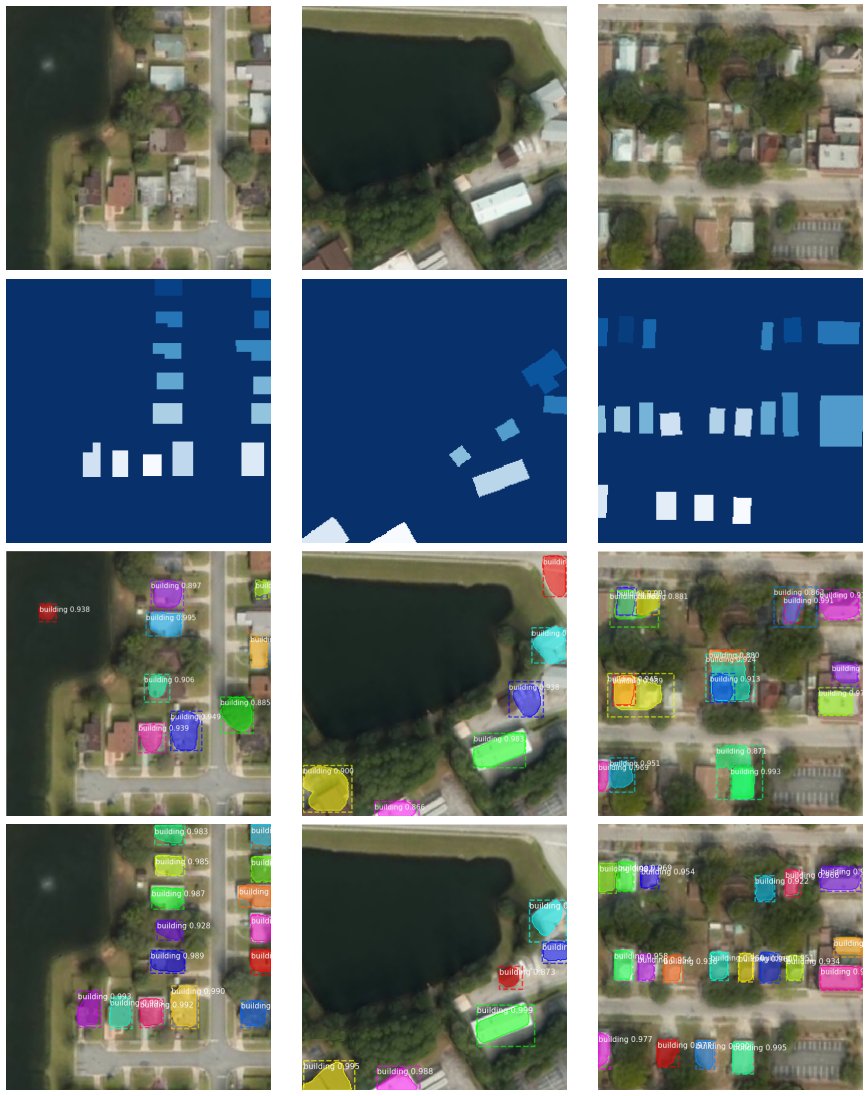

Figure 5: Mask-RCNN instance segmentation results. Results by rows: 1 - source, 2 - ground truth, 3 - prediction at the beginning of the training, 4 - prediction at the end of the training

Figure 6: Conditional Adversarial Network semantic segmentation. Results by rows: 1 - source, 2 - ground truth, 3 - ground truth highlighted on source, 4 -predicted image, 5 - predicted image highlighted on source

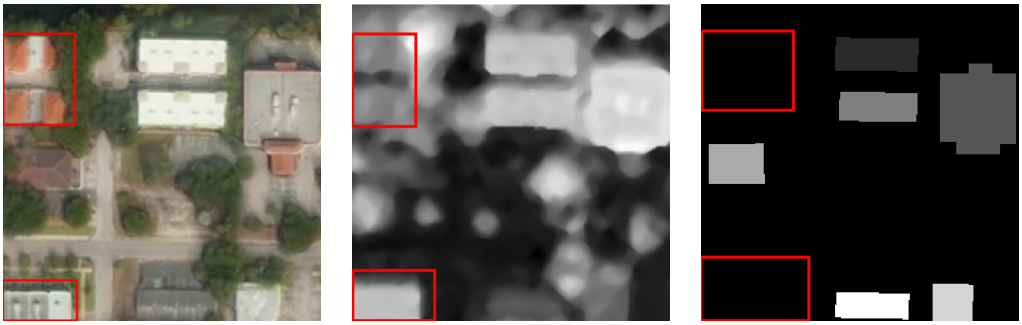

Figure 7: Left to right: RGB, DSM (depth), Ground Truth. Human error areas marked in red.

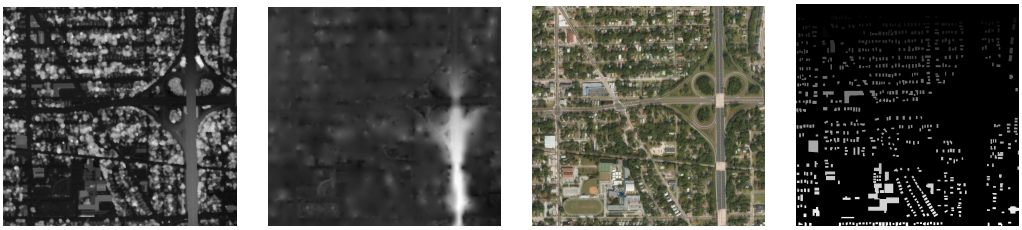

Figure 8: Left to right: DSM (depth), DTM (terrain), RGB, Ground Truth

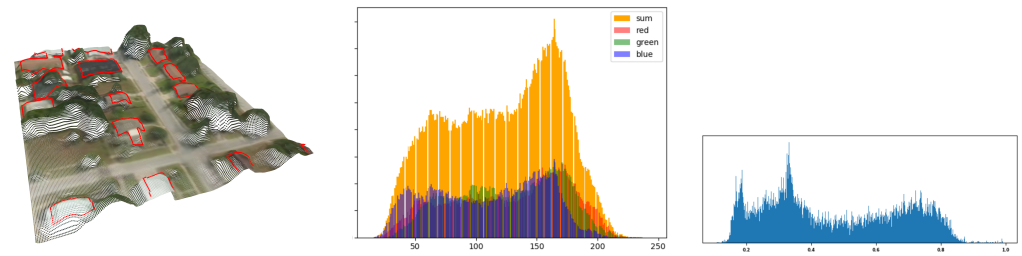

Figure 9: Left to right: 3D point cloud, RGB hist, DSM (depth) hist