Towards Real-time House Detection in Aerial Imagery Using Faster Region-based Convolutional Neural Network

Ahmed, Khandaker Mamun; Ghareh Mohammadi, Farid; Matus, Manuel; Shenavarmasouleh, Farzan; Manella Pereira, Luiz; Ioannis, Zisis; Amini, M. Hadi

Abstract: ***In the past few years, automatic building detection in aerial images has become an emerging field in computer vision. Detecting the specific types of houses will provide information in urbanization, change detection, and urban monitoring that play increasingly important roles in modern city planning and natural hazard preparedness. In this paper, we demonstrate the effectiveness of detecting various types of houses in aerial imagery using Faster Region-based Convolutional Neural Network (Faster-RCNN). After formulating the dataset and extracting bounding-box information, pre-trained ResNet50 is used to get the feature maps. The fully convolutional Region Proposal Network (RPN) first predicts the bounds and objectness score of objects (in this case house) from the feature maps. Then, the Region of Interest (RoI) pooling layer extracts interested regions to detect objects that are present in the images. To the best of our knowledge, this is the first attempt at detecting houses using Faster R-CNN that has achieved satisfactory results. This experiment opens a new path to conduct and extent the works not only for civil and environmental domain but also other applied science disciplines.***

Index Terms: ***RCNN, Neural Network, Deep Learning, Convolution, Mini batch***

1. INTRODUCTION

In this section, we present the motivation for the development of an application to detect houses in aerial images. Subsequently, we discuss the prior works that have recently been published and explain how our proposed framework can be beneficial in the modern urbanized world. We also show the novelty of this paper, which is followed by a brief description of the paper's organization.

Manuscript received February 13, 2023.

Corresponding Author: M. Hadi Amini, moamini@fiu.edu
Khandaker Mamun Ahmed, M. Hadi Amini and Luiz Manella Pereira are with the Knight Foundation School of Computing and Information Sciences, Florida International University (FIU), Miami, FL, USA; and the Sustainability, Optimization, and Learning for InterDependent networks laboratory (solid lab), FIU, Miami, FL, USA
Farid Ghareh Mohammadi is with the Department of Radiology, Center for Augmented Intelligence (CAI), Mayo Clinic, Jacksonville, FL, USA
Manuel Matus and Ioannis Zisis is with the Dept. of Civil & Environ. Engineering Florida International University, Miami, FL, USA
Farzan Shenavarmasouleh is with the R&D Department, MediaLab Inc., GA, USA

1.1 Motivation

House detection is an important problem in computer vision and pattern recognition which has gained considerable attention in the past few decades [1]–[3]. Due to rapid urbanization, detecting houses plays a salient role in modern city planning, urban monitoring, change detection, and population estimation. Moreover, building shape related information can provide valuable input in engineering and risk applications related to natural hazards (e.g. extreme wind events, flooding, etc.). Aerial imagery is one of the prominent data sources for urban monitoring because it extracts various information such as roads, trees, buildings, etc. Although aerial imagery provides valuable insights, extracting appropriate features from them is a challenging task.

On the other hand, in recent years, deep learning models, especially Convolutional Neural Network (CNN) based models, have become a popular choice among the researchers for its state-of-the-art success in image classification, object detection, and localization tasks [4]–[7]. Faster-RCNN is a recently proposed object detection algorithm that has achieved state of-the-art results in ImageNet Large Scale Visual Recognition Challenge (ILSVRC) [8], [9]. In this work, we have utilized a faster-RCNN algorithm to detect buildings in aerial images.

1.2 Literature Review

In this section, we first talk about the history the algorithm applied in this work followed by a brief review of the prior works.

1.2.1 CNN and RCNN family of Algorithms:

Due to the rapid developments of science and technology (e.g., advancements in automated vehicles, robotic navigation, and object tracking), object detection has become a prominent field of study. The goal of object detection is to find the location of an object from a given image and mark the object in an appropriate category. However, object detection is a challenging task. The object's orientation, location, size, and

altitude can vary greatly in an image, making the task more difficult to solve. In the human visual system, we not only see and identify an object, we can identify multiple overlapping objects in diverse backgrounds. Moreover, we can classify these different objects and identify their boundaries, differences, and relationship to one another. However, in the field of computer vision, CNN-based architectures are applied successfully to solve various detection related tasks such as face detection, pedestrian detection and vehicle detection [10]–[14].

The first successful CNN architecture was developed by Yann Lecun in 1998 to recognize handwritten digits on checks [15]. In 2012, more than 12 years later, Alex Krizhevsky et al. followed his path and built the famous AlexNet algorithm that won the ImageNet challenge [16]. Since then, CNN architectures have become the gold standard for solving computer vision tasks and are now outperforming humans in some scenarios.

In 2014 Girshick et al. proposed the Regions with CNN features (R-CNN) algorithm for object detection, which is the first algorithm of the R-CNN family of algorithms [17]. RCNN achieved the mean average precision (mAP) result of 53.3% in PASCAL VOC dataset. To capture all possible objects' locations from a given image, authors applied the selective search algorithm [18]. The selective search algorithm proposes 2k regions for an image. In Figure 1, two examples of selective search are given where different sized scales are used to capture all possible objects. Each proposed region is warped to a compatible form of 227×227 pixels and forward propagated through the CNN architecture to compute feature maps. Next, the Support Vector Machine (SVM) algorithm is utilized to compute the classification score. In the RCNN architecture the workflow is like: an input image is given to detect possible objects; the selective search algorithm proposes ∼2k regions which are forwarded to the CNN layers, and the CNN architecture generates feature maps to detect which objects are present in the image. To compute the region proposal and features for images, R-CNN requires 13 s/image on a GPU integrated environment and 53 s/image on a CPU based environment, which is a significantly high computation time. Therefore, to minimize the computation time required by RCNN, an improved version of RCNN named Fast-RCNN was proposed by the same author Ross Girshick [19] in 2015.

The Fast-RCNN model requires an input image and a set of object proposals for its computation. Initially, it processes the whole image with several convolutional (conv) layers and max-pooling layers to produce the feature maps. Then, a fixed□length feature vector from the feature map is extracted by the RoI pooling layer to classify objects. Fast-RCNN is 25 times faster than R-CNN with the test time of 2 seconds per image. Even though Fast-RCNN significantly improved the processing time and model's performance, the selective search was still the bottleneck that slowed down the overall process. Region proposals are dependent on the feature maps and reusing the feature maps to generate region proposals will be cost-free. Taking this idea into consideration, Ren et al. developed the faster R-CNN that exceptionally improved the overall model performance [8]. In Figure 2, we show a faster R-CNN algorithm where conv layers compute the feature maps and RPN layer extracts region proposals from the feature maps for classification. The faster R-CNN algorithm can detect objects in real time with the computational time of 0.2 seconds per image.

Figure 3 demonstrates the performance comparison of the R-CNN architectures where we can see that faster R-CNN reduced processing time by 250x, whereas Fast-RCNN had a reduction of 25x against the base case processing time of x for R-CNN. Both faster and Fast-RCNN maintained the same mean average precision (mAP) score of 66.9%, where R-CNN architecture's mAP score was 66.0%. [1]

1.2.2 Recent Works on House Detection:

Buildings are the primary source of information for urban planners and, many governmental and non-governmental agencies as they provide the holistic overview of a geographical area. However, building detection is a challenging task because of its complex appearance, variant shapes, and surroundings. In the past few years, researchers have proposed several building extraction methods and followed various approaches [20]–[22]. Although building detection methods with good performance have evolved significantly over the years, there are still many aspects that have not been considered and need improvements.

Stankov et al. [23], [24] exploited the multispectral information and applied a grayscale hit-or-miss transform (HMT) method for building detection. In the paper, authors transformed the multispectral images to grayscale images in order to apply grayscale HMT. Sirmacek et al. [25] extracted shadow information and areas of interest using invariant color features and utilized edge information building detection. In [26], Ziaei et al. presented a comparison between three object-based models for urban feature classification from WorldView-2 images, where they have shown that rule-based classification outperformed support vector machines (SVM), and nearest neighbour (NN) algorithms. Building extraction from Quickbird images is presented by Lefevre et al. [27] by using an adaptive ` binary HMT method. Authors also proposed a

[1]Stanford lecture notes on CNN by Fei Fei Li and Andrej Karpathy

clustering-based approach to convert grayscale image to binary image and to determine operators parameters automatically. In [28], Grinias et al. presented a novel segmentation algorithm based on a Markov random field model for building and road detection. To detect changes of buildings from VHR imagery, Guo et al. [29] presented a parameter mining approach by introducing GIS data. For automatically extracting and recognizing 2- D building shape information, Sahar et al. [30] used vector parcel geometries and their attributes to simplify the building extraction task. Huang et al. [31] introduced a framework for building extraction from high-resolution imagery aiming to alleviate Morphological Building Index (MBI) algorithm's limitations. Benarchid et al. [32] used shadow information and object-based approach to extract buildings where they first used object-based classification to detect building and then the invariant color features to extract shadow information of the buildings. Based on shadow detection, Chen et al. [33] proposed a superpixel segmentation algorithm for splitting input image into patches, and the Level Set segmentation algorithms is leveraged to extract buildings for detection.

In this paper, we present a Faster RCNN based deep learning model that can detect different houses in aerial images.

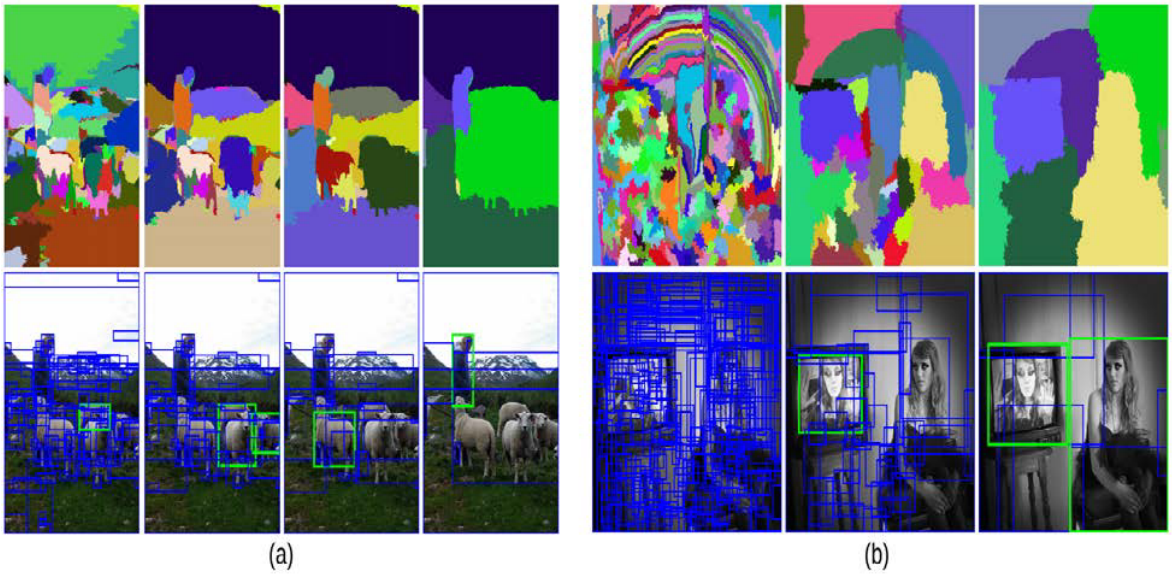


Figure 1: Two examples of selective search showing the necessity of different scales. On the left we find many objects at different scales. On the right we necessarily find the objects at different scales as the girl is contained by the tv [18].

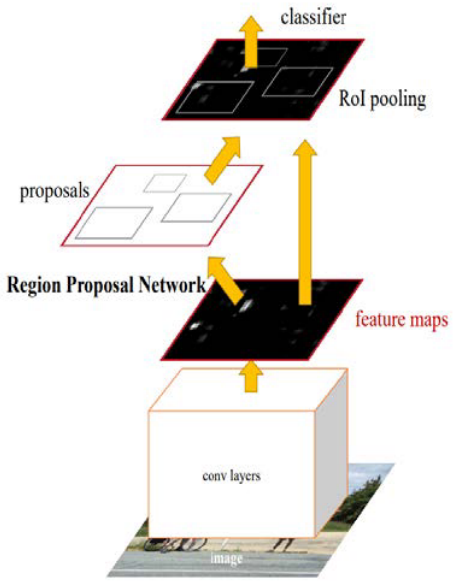
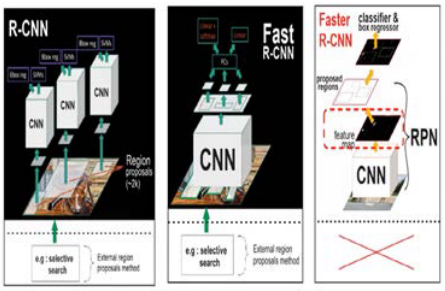


Figure 2: Faster-RCNN architecture.



	R-CNN	Fast R-CNN	Faster R-CNN
Test time per image	50 seconds	2 seconds	0.2 seconds
Speed-up	1x	25x	250x
mAP (VOC 2007)	66.0%	66.9%	66.9%

Figure 3: Performance comparison of R-CNN architectures: R CNN, Fast-RCNN, Faster R-CNN. [1]

1.3 Contribution

Faster-RCNN is one of the promising algorithms for object detection that has also opened up the area of real time object detection. In some situations, we need to extract the building's information in real time and our proposed method can be a good fit for such scenarios. It is our understanding that faster-RCNN based house detection technique, which paves the way for real time detection, has not been considered in previous works. The main contributions of this paper are listed as follows:

- House detection in aerial images leveraging faster R-CNN algorithm that paves the way for real time detection.
- Bounding-box information extraction and preprocessing of the dataset to remove inconsistent data that may hamper the overall performance of the model.
- Demonstrate the effectiveness of data augmentation such as random rotation, horizontal flip and shearing to im□prove performance and generalizability, and avoid over-fitting.
- Demonstrate our model's performance by considering average precision, loss function, prediction scores and image precision.

1.4 Organization

The paper is organized as follows: Section II presents the methodology of the work including data pre-processing, data augmentation and the house detection technique. Section III represents experimental setup. Section IV is dedicated for result analysis. Finally, Section V concludes the paper.

2. Proposed Method

This section discusses data pre-processing, and data augmentation techniques, and the methodology used to detect houses. In Figure 4, we show the overall architecture of our proposed model that includes dataset generation, data preprocessing, data augmentation, object detection, and results afterwards.

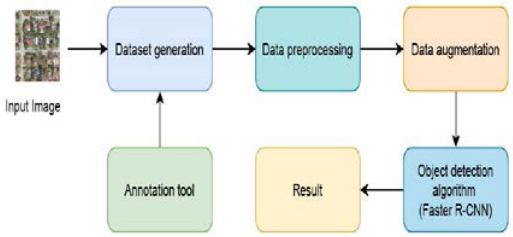


Figure 4: Overview of methodology adopted in this study

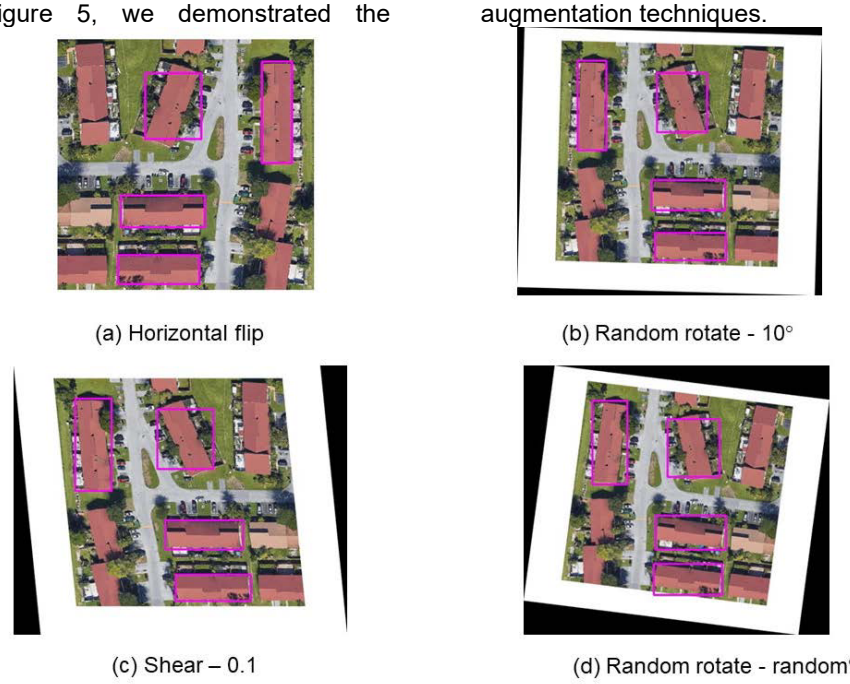
2.1 Data pre-processing

In our dataset, we have aerial images and XML files containing the annotation information of the images. XML file is an extensible markup language file where components of the file are described by tags, and texts in between the start tag and end tag are the contents of the component. From the XML files, we extract the associated bounding-box information (for our case its the aerial image file, xmin, ymin, xmax, ymax and label) of each image. In the generated dataset, we observed 37 different labels / categories of houses where most of them are redundant (e.g., typo and inconsistent labels). For example the category of T shaped houses were labelled as t shape, t-shaped, t type, type t and t-shape which is inconsistent and it can be minimized to one category. After analyzing 37 labels, we concluded that 37 different labels can be minimized to only 5 categories (T shaped, L shaped, C shaped, Rectangular shaped, and U shaped). Moreover, we had some anomalies in the extracted information such as xmin > xmax or ymin > ymax. In such cases, if possible, we exchanged min and max values without changing the bounding-box information of an object, otherwise we disregarded them due to incorrect bounding boxes.

2.2 Data augmentation

Data augmentation is a technique to artificially expand the dataset size by marginally modifying the original data. Data augmentation helps to avoid overfitting and improves model's performance. In images data augmentation technique is performed by flipping, random rotation, shifting, or shearing the original image. Deep learning is a data-hungry technique that yields better performance with larger dataset, avoids over-fitting, and improves the model's generalizability. Therefore to improve model performance and avoid overfitting, we augmented our dataset using horizontal flip, random rotation with the angle value of 10 degrees, shears with the value of 0.1, and

random rotation with randomly generated angle value. In Figure 5, we demonstrated the augmented results after applying the data augmentation techniques.



(a) Horizontal flip (b) Random rotate - 10°

(c) Shear – 0.1 (d) Random rotate - random°

Figure 5: Data augmentation: 5a Horizontal flip; 5b Random rotation with 10°;5c Shear with 0.1;5d Random rotation with a random value.

2.3 House Detection using Faster-RCNN

The most widely used state-of-the-art object detection technique of the R-CNN family is Faster R-CNN that was first published in 2015 [8]. In the R-CNN family of papers, the evolution among versions is usually in terms of computational efficiency, processing time, and performance improvement (i.e. mAP). These networks usually consist of

1. A region proposal algorithm to generate "bounding boxes" or locations of possible objects in the image.
2. A feature generation stage to obtain features of these objects (usually using a CNN).
3. A classification layer to predict which class an object belongs to. 4) A regression layer to make the coordinates of the object bounding boxes more precise.

To generate feature maps (e.g., Figure 7), ResNet50 is utilized in the initial stage where the input image goes through a set of convolutional layers, pooling layers and fully connected layers. After generating feature maps, RPN layer which is a small network, takes the feature map as an input, slides over it, and outputs a set of rectangular object proposals. Nine region proposals (anchors) are predicted at each sliding window location with respect to the center (Figure 8) of the anchor associated with scales of (128 x 128, 256 x 256, 512 x 512) and aspect ratios of (1:1, 1:2 and 2:1) (Figure 6). A binary class label of being an object or not an object is assigned to each anchor for RPN training based on the Intersection-over-Union (IoU) overlap with the ground-truth box. An anchor is considered positive if it has the highest IoU with any ground truth box or is greater than 0.7. If the IoU is less than 0.3 it is labeled as negative. The anchors which are neither positive nor negative (greater than 0.3 and less than 0.7) are disregarded from the RPN training. The loss function of RPN is defined as:

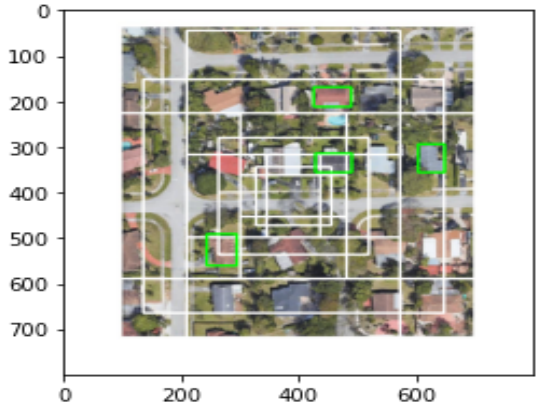


Figure 6: An example of generating 9 anchors from a single centroids with different scales and aspect ratios.

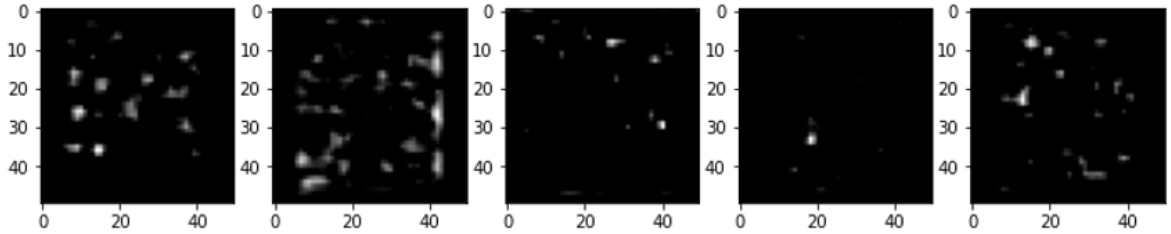


Figure 7: Sample feature map

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

Here, i is the index of an anchor in a mini-batch and p_i is the predicted probability of anchor i being an object. The ground-truth label P_i* is 1 if the anchor is positive and is 0 if the anchor is negative. t_i is a vector representing the 4 parameterized coordinates of the predicted bounding box and t_i* is that of the ground-truth box associated with a positive anchor. The classification loss L_{cls} is log loss over two classes (object vs. not object). For the regression loss, we use $L_{reg}(t_i , t_i*) = R(t_i - t_i*)$ where R is the robust loss function (smooth L1). The term P_i*L_{cls} means the regression loss is activated only for positive anchors $P_i* = 1$ and is disabled otherwise (i.e. $P_i* = 0$). The outputs of the cls and reg layers consist of p_i and t_i respectively. The two terms are normalized by N_{cls} and N_{reg} and weighted by a balancing parameter λ.

For the model training, the batch size is defined to 16 and stochastic gradient descent (SGD) optimizer is applied with the learning rate of 0.005, momentum of 0.9 and weight decay of 0.005.

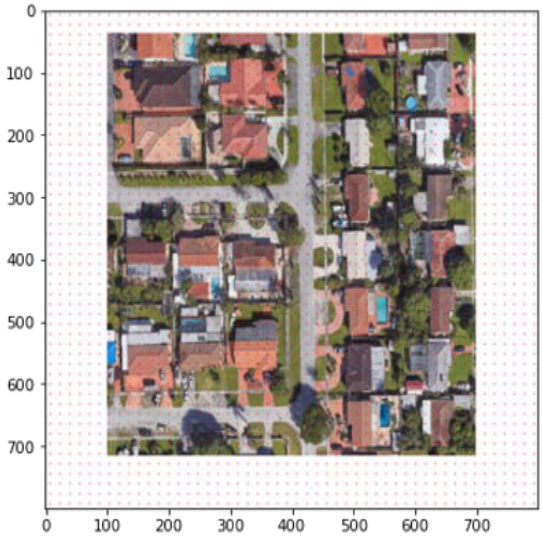


Figure 8: Centriods of RPN.

3. EXPERIMENTAL SETUP

The entire experiment is carried out in Google Colab environment developed by Google as a simulation environment. The experiment leverages Colab environment utilizing GPU runtime settings using python as the programming language. The deep learning object detection classifier has been implemented using python version 3.7.3 and the PyTorch framework.

4. EXPERIMENTAL EVALUATION

This section provides a brief description of the dataset we have used for our experiments followed by the performance evaluation of our proposed work.

4.1 Dataset Description

In this experiment, we explored google earth images to detect houses of different shapes. In Figure 9, we demonstrate the process of creating our dataset using LabelMe [34] annotation tool where house objects are manually annotated in each image. The annotation tool then generates an XML file containing the annotated information for each image. (Figure 11) shows the structure of a sample xml file after completing the annotation process and in Figure 10 we show a sample annotated image afterwards. Finally, the annotation files along with the associated aerial image dataset are downloaded from the LabelMe application for carrying out the experiment.

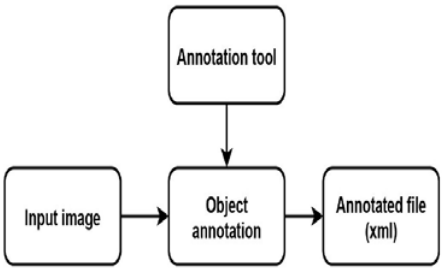


Figure 9: Flowchart for dataset annotation.

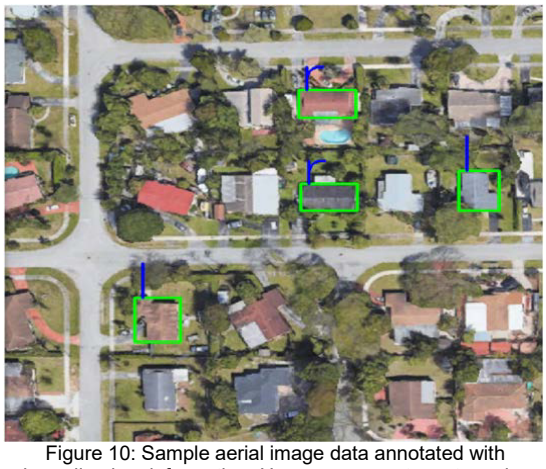


Figure 10: Sample aerial image data annotated with bounding box information. Here, r represents rectangular shaped houses and l represents l shaped houses

```
−<annotation>
   <filename>17.jpg</filename>
  −<folder>
     users/WindEngineering//topics_in_wind_engineering_final/sunil
   </folder>
  −<source>
     <submittedBy>Manuel Matus</submittedBy>
   </source>
  −<imagesize>
     <nrows>945</nrows>
     <ncols>1072</ncols>
   </imagesize>
  −<object>
     <name>building 1</name>
     <deleted>0</deleted>
     <verified>0</verified>
     <occluded>no</occluded>
     <attributes>c shaped</attributes>
    −<parts>
       <hasparts/>
       <ispartof/>
     </parts>
     <date>02-Dec-2019 03:32:24</date>
     <id>0</id>
     <type>bounding_box</type>
    −<polygon>
       <username>anonymous</username>
      −<pt>
         <x>169</x>
         <y>531</y>
       </pt>
      −<pt>
         <x>284</x>
         <y>531</y>
```

Figure 11: XML file: Annotation information of images such as shape, number, bounding-box information

4.2 Experimental results

Object detectors performance is measured by average precision (AP), image precision and loss functions. In our experiment, we evaluated our methods performance by average precision, image precision and loss function. We defined different number of epochs to observe the model's performance. In our observation, the simulation performs better with twenty epochs. In Figure 12, we demonstrate average precision in different IoU thresholds: 0.50, 0.55, 0.60, 0.65, 0.70, 0.75. As the IoU threshold increases the average precision decreases naturally. Moreover, in Figure 13, we show the average image precision by comparing all IoU thresholds. From Figure 13, we can see that image precision increases moderately for 20 epochs. In Figure 14, we show the loss function against the number of iterations where we observe that after 400 iterations with twenty epochs the loss function is converged. The equations for calculating precision, average precision are discussed in the followings where t_p = True positive; f_p = False positive; t_n = True negative; f_n = False negative.

$$Precision(P) = \frac{tp}{tp + fp} \tag{1}$$

$$Recall(R) = \frac{tp}{tp + fn} \tag{2}$$

4.2.1 Intersection over union (IoU)

IoU measures the overlap between 2 boundaries. We use that to measure how much our predicted boundary overlaps with the ground truth. In our dataset, we defined various IoU threshold r ∈ {0.5, ..., 0.75} in classifying whether the prediction is a true or a false positive. Intersection over Union (IoU) for comparing similarity between the ground-truth and predicted shapes A, B ⊆ S ∈ Rn is attained by equation 3.

$$IoU = \frac{|A \cap B|}{|A \cup B|} \tag{3}$$

4.2.2 Interpolated precision

The interpolated precision, p_{interp}, is calculated at each recall level, r, by taking the maximum precision measured for that r. The formula is given as such:

$$p_{interp}(r) = \max_{r' \geq r} P(r') \tag{4}$$

In our experiment an average for the 6-point interpolated average precision (AP) is calculated. And the formula to calculate the AP is attained by:

$$AP = \frac{1}{6} \sum_{r \in \{0.5,\ldots,0.75\}} AP_r = \frac{1}{6} \sum_{r \in \{0.5,\ldots,0.75\}} p_{interp}(r) \tag{5}$$

where

$$p_{interp}(r) = \max_{r' \geq r} P(r')$$

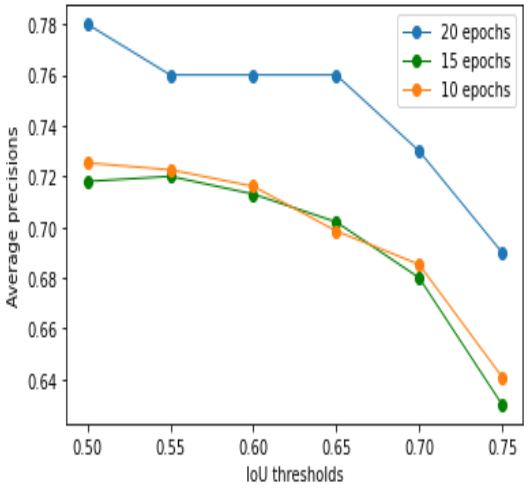


Figure 12: Average precision

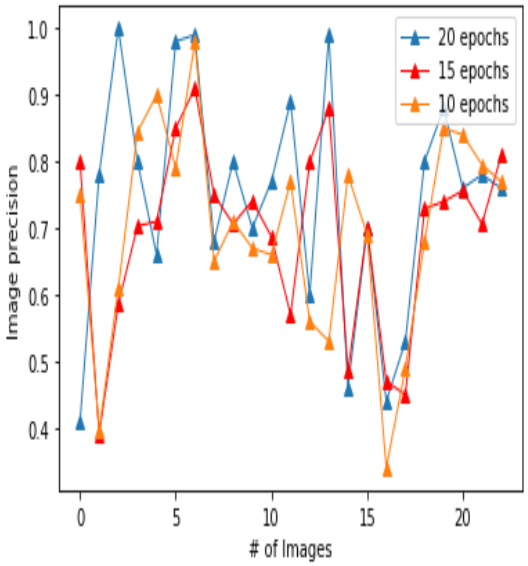


Figure 13: Image precision

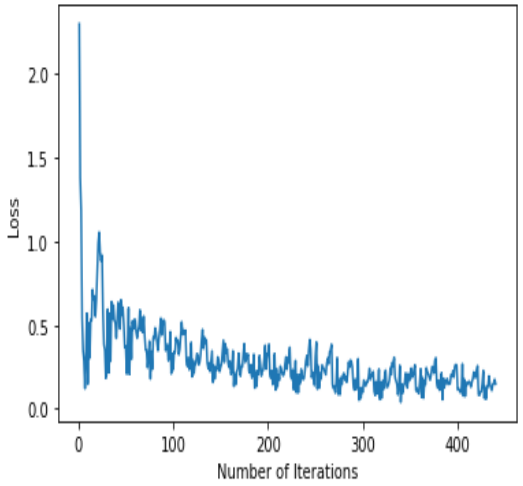


Figure 14: Loss function.

5. Conclusion And Future Works

House detection is a fundamental but challenging issue in the field of aerial and satellite image analysis. It provides valuable information in different domains including civil engineering, urbanization, and modern city planning. During the last few years, considerable efforts have been made to develop various methods for detecting houses in aerial images. In this paper, we present a Faster-RCNN based house detection method that achieved a satisfactory result. Our proposed method can be utilized in real time object/house detection scenarios. A wide range of ensembles of faster RCNN is being utilized in various contexts such as pedestrian detection, vehicle detection, and face detection. In this experiment, we have leveraged pretrained resnet-50 model to detect houses in aerial images. A performance comparison of various models, such as VGG19, SeNet, GoogleNet, MobileNetV2, DenseNet201, and InceptionResNetV2, is important for both application and academic purposes and thus remains an integral part of our future research.

Acknowledgment

In this work, we have leveraged faster RCNN algorithm and carried out the experiment in Google Colab environment. We are thankful to both communities. We also acknowledge the resources and support of Sustainability, Optimization, and Learning for InterDependent networks laboratory (solid lab) at Knight Foundation School of Computing and Information Sciences (KFSCIS), Florida International University (www.solidlab.network). We also acknowledge the effort of Divya Saxena from the KFSCIS, FIU, funded by NSF grant CNS-2018611 at the FIU High Performance Database Research Center.

This work was partially supported by the Graduate Assistantships in Areas of National Need (GAANN) fellowship from the Department of Education grant P200A210087.

Declaration Of Competing Interest

Authors declare no conflict of interest.

References

[1] S. Zou and L. Wang, “Detecting individual abandoned houses from google street view: A hierarchical deep learning approach,” ISPRS Journal of Photogrammetry and Remote Sensing, vol. 175, pp. 298– 310, 2021.

[2] S. Law, B. Paige, and C. Russell, “Take a look around: using street view and satellite images to estimate house prices,” ACM Transactions on Intelligent Systems and Technology (TIST), vol. 10, no. 5, pp. 1–19, 2019.

[3] J. A. Tullis and J. R. Jensen, “Expert system house detection in high spatial resolution imagery using size, shape, and context,” Geocarto International, vol. 18, no. 1, pp. 5–15, 2003.

[4] W. Rawat and Z. Wang, “Deep convolutional neural networks for image classification: A comprehensive review,” Neural computation, vol. 29, no. 9, pp. 2352–2449, 2017.
[5] Z.-Q. Zhao, P. Zheng, S.-t. Xu, and X. Wu, “Object detection with deep learning: A review,” IEEE transactions on neural networks and learning systems, vol. 30, no. 11, pp. 3212–3232, 2019.
[6] X. Wu, D. Sahoo, and S. C. Hoi, “Recent advances in deep learning for object detection,” Neurocomputing, vol. 396, pp. 39–64, 2020.
[7] K. M. Ahmed, T. Eslami, F. Saeed, and M. H. Amini, “Deepcovidnet: Deep convolutional neural network for covid-19 detection from chest radiographic images,” in 2021 IEEE International Conference on Bioinformatics and Biomedicine (BIBM). IEEE, 2021, pp. 1703–1710.
[8] S. Ren, K. He, R. Girshick, and J. Sun, “Faster r-cnn: Towards real-time object detection with region proposal networks,” arXiv preprint arXiv:1506.01497, 2015.
[9] “ImageNet Large Scale Visual Recognition Challenge 2016 Results (ILSVRC2016),” http://www.image-Net.org/challenges/LSVRC/2016/results, [Online; accessed 06-April-2021].
[10] J. Yu, Y. Jiang, Z. Wang, Z. Cao, and T. Huang, “Unitbox: An advanced object detection network,” in Proceedings of the 24th ACM international conference on Multimedia, 2016, pp. 516–520.
[11] S. S. Farfade, M. J. Saberian, and L.-J. Li, “Multi-view face detection using deep convolutional neural networks,” in Proceedings of the 5th ACM on International Conference on Multimedia Retrieval, 2015, pp. 643–650.
[12] S. Yang, P. Luo, C.-C. Loy, and X. Tang, “From facial parts responses to face detection: A deep learning approach,” in Proceedings of the IEEE international conference on computer vision, 2015, pp. 3676–3684.
[13] Z. Jiang and D. Q. Huynh, “Multiple pedestrian tracking from monocular videos in an interacting multiple model framework,” IEEE transactions on image processing, vol. 27, no. 3, pp. 1361–1375, 2017.
[14] D. M. Gavrila and S. Munder, “Multi-cue pedestrian detection and tracking from a moving vehicle,” International journal of computer vision, vol. 73, no. 1, pp. 41–59, 2007.
[15] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, “Gradient-based learning applied to document recognition,” Proceedings of the IEEE, vol. 86, no. 11, pp. 2278–2324, 1998.
[16] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” Advances in neural information processing systems, vol. 25, pp. 1097–1105, 2012.
[17] R. Girshick, J. Donahue, T. Darrell, and J. Malik, “Rich feature hierarchies for accurate object detection and semantic segmentation,” in Proceedings of the IEEE conference on computer vision and pattern recognition, 2014, pp. 580–587.
[18] J. R. Uijlings, K. E. Van De Sande, T. Gevers, and A. W. Smeulders, “Selective search for object recognition,” International journal of computer vision, vol. 104, no. 2, pp. 154–171, 2013.
[19] R. Girshick, “Fast r-cnn,” in Proceedings of the IEEE international conference on computer vision, 2015, pp. 1440–1448.
[20] D. A. Yudin, V. Adeshkin, A. V. Dolzhenko, A. Polyakov, and A. E. Naumov, “Roof defect segmentation on aerial images using neural networks,” in International Conference on Neuroinformatics. Springer, 2020, pp. 175–183.
[21] H. Miura, T. Aridome, and M. Matsuoka, “Deep learning-based identification of collapsed, non-collapsed and blue tarp-covered buildings from post-disaster aerial images,” Remote Sensing, vol. 12, no. 12, p. 1924, 2020.
[22] A. D. Schlosser, G. Szabo, L. Bertalan, Z. Varga, P. Enyedi, and S. Szabo, “Building extraction using orthophotos and dense point cloud ´ derived from visual band aerial imagery based on machine learning and segmentation,” Remote Sensing, vol. 12, no. 15, p. 2397, 2020.
[23] K. Stankov and D.-C. He, “Building detection in very high spatial resolution multispectral images using the hit-or-miss transform,” IEEE Geoscience and Remote Sensing Letters, vol. 10, no. 1, pp. 86–90, 2012.
[24] ——, “Detection of buildings in multispectral very high spatial resolution images using the percentage occupancy hit-or-miss transform,” IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, vol. 7, no. 10, pp. 4069–4080, 2014.
[25] B. Sirmacek and C. Unsalan, “Building detection from aerial images using invariant color features and shadow information,” in 2008 23rd International Symposium on Computer and Information Sciences. IEEE, 2008, pp. 1–5.
[26] Z. Ziaei, B. Pradhan, and S. B. Mansor, “A rule-based parameter aided with object-based classification approach for extraction of building and roads from worldview-2 images,” Geocarto International, vol. 29, no. 5, pp. 554–569, 2014.
[27] S. Lefevre, J. Weber, and D. Sheeren, “Automatic building extraction ` in vhr images using advanced morphological operators,” in 2007 Urban Remote Sensing Joint Event. IEEE, 2007, pp. 1–5.
[28] I. Grinias, C. Panagiotakis, and G. Tziritas, “Mrf-based segmentation and unsupervised classification for building and road detection in peri□urban areas of high-resolution satellite images,” ISPRS journal of photogrammetry and remote sensing, vol. 122, pp. 145–166, 2016.
[29] Z. Guo and S. Du, “Mining parameter information for building extraction and change detection with very high-resolution imagery and gis data,” GIScience & Remote Sensing, vol. 54, no. 1, pp. 38–63, 2017.
[30] L. Sahar, S. Muthukumar, and S. P. French, “Using aerial imagery and gis in automated building footprint extraction and shape recognition for earthquake risk assessment of urban inventories,” IEEE Transactions on Geoscience and Remote Sensing, vol. 48, no. 9, pp. 3511–3520, 2010.
[31] X. Huang and L. Zhang, “Morphological building/shadow index for building extraction from high-resolution imagery over urban areas,” IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, vol. 5, no. 1, pp. 161–172, 2011.
[32] O. Benarchid, N. Raissouni, S. El Adib, A. Abbous, A. Azyat, N. B. Achhab, M. Lahraoua, and A. Chahboun, “Building extraction using object-based classification and shadow information in very high reso□lution multispectral images, a case study: Tetuan, morocco,” Canadian Journal on Image Processing and Computer Vision, vol. 4, no. 1, pp. 1–8, 2013.
[33] D. Chen, S. Shang, and C. Wu, “Shadow-based building detection and segmentation in high-resolution remote sensing image.” journal of multimedia, vol. 9, no. 1, pp. 181–188, 2014.
[34] “LabelMe, the open annotation tool,” http://labelme.csail.mit.edu/Release3.0/, accessed: 2021-11-10