Resource-Constrained Simultaneous Detection and Labeling of Objects in High-Resolution Satellite Images

Gilbert Rotich$^1$, Rodrigo Minetto$^2$ and Sudeep Sarkar$^1$

$^1$ University of South Florida (USF)
$^2$ The Federal University of Technology - Paraná (UTFPR)

rminetto@dainf.ct.utfpr.edu.br

Abstract—We describe a strategy for detection and classification of man-made objects in large high-resolution satellite photos under computational resource constraints. We detect and classify candidate objects by using five pipelines of convolutional neural network processing (CNN), run in parallel. Each pipeline has its own unique strategy for fine tunning parameters, proposal region filtering, and dealing with image scales. The conflicting region proposals are merged based on region confidence and not just based on overlap areas, which improves the quality of the final bounding-box regions selected. We demonstrate this strategy using the recent xView challenge, which is a complex benchmark with more than 1,100 high-resolution images, spanning 800,000 aerial objects around the world covering a total area of 1,400 square kilometers at 0.3 meter ground sample distance. To tackle the resource-constrained problem posed by the xView challenge, where inferences are restricted to be on CPU with 8GB memory limit, we used lightweight CNN’s trained with the single shot detector algorithm. Our approach was competitive on sequestered sets; it was ranked third.

I. INTRODUCTION

The localization and classification of geographical regions in high-resolution aerial images provides critical information for analysts and police makers around the world to make decisions about territorial defense, humanitarian assistance and environmental conservation policies. Although extensively studied [1], [2], [3], [4], such research problems are still very challenging. The technical difficulties have been exposed by the recent large challenge problems that have been constructed such as the Spacenet challenges [5] (https://spacenetchallenge.github.io/), IARPA Functional Map of the World (fMoW) challenge [6], and the DoD xView challenge [7].

SpaceNet challenge focused on the problem of building and road networks in satellite images of five metropolitan areas with over 685,000 footprints. The fMoW contest published one of the hardest and largest benchmark for region classification in aerial images to date, with 1 million images around 100,000 globe locations. The goal in fMoW was the classification of a given region as one of 62 target classes or as a false detection. The xView challenge published another complex benchmark with more than 1,100 high-resolution images, spanning a total area of 1,400 square kilometers collected at 0.3 meter ground sample distance (GSD), and containing more than 800,000 aerial objects around the world. The goal in xView was the localization and classification of such objects into 60 classes.

These novel benchmarks in remote sensing foster major breakthroughs in machine learning, by addressing complex problems such as

- **Fine grained categorization**, which is the the classification of visually-similar objects from subordinate categories. For instance, in the xView challenge for truck vehicles there are eight different sub-categories: pickup truck, utility truck, cargo truck, truck with box, truck tractor trailer, truck with flatbed, truck with liquid.

- **Resource-constrained learning** by imposing limitation on computational resources for inference and learning. This is necessary, for instance, if the final deployment is in unmanned aerial vehicle (drone), where code has to run efficiently on embedded low-power processors.

- **Class imbalance**, which is the problem of learning from an unequal number of observations per class. This is an inherent problem in remote sensing, where, we usually have many instances of common objects like cars, buses or buildings and few instances of other objects like excavators, locomotives and helipads.

- **Spatial learning**, which is the detection and classification of objects embedded in clutter background, with large scale variation and with partial occlusion by clouds or shadows. These problems result in high intra-class variations and considerable interclass confusion.

- **Temporal learning**, which is the problem of learning from image sequences of the same geographical scene, recorded in arbitrary satellite viewpoints and time periods. For instance, detecting such changes could help in damage assessment and rescue efforts in case of natural disasters or in environmental conversation in case of deforestation.

In this paper we describe a framework for simultaneous detection and classification of objects in high-resolution aerial images. The input data for this problem is an aerial image, see Figure (top), and the output consists of $n$ regions

$$\mathcal{R} = \{r_1, r_2, \ldots, r_n\}$$

where each region $r_i$ is defined by an axis-aligned rectangular box $r_i(b) = (x_1, y_1, x_2, y_2)$ where $(x_1, y_1)$ and $(x_2, y_2)$ represent the upper left and bottom right corner’s, respectively; an integer $r_i(c) \in \{1, 2, \ldots, N\}$ that represent
the object category; and a confidence score \( r_i(w) \), for the classification, expressed as a real number, within the interval \([0, 1]\). An example of an output is shown in Figure 1 (bottom).

We detect and classify candidate regions by using five pipelines of convolutional neural networks (CNN) to cope with the aforementioned problems in remote sensing (fine-grained categorization, class imbalance, etc).

Each pipeline has its own unique strategy for fine tuning parameters, proposal region filtering, and dealing with image scales. The conflicting region proposals are merged based on region confidence and not just based on overlap areas, which improves the quality of the final bounding-box regions selected. To tackle the resource-constrained problem, we used lightweight CNN’s trained with the single shot detector algorithm as the core deep learning approach.

The remainder of the paper is organized as follows. The proposed framework is described in Section II and its experimental evaluation is reported in Section III. We analyze the results and discuss research extensions in Sections IV and V.

II. PROPOSED FRAMEWORK

As shown in Figure 2, our framework consists of five pipelines for detection and labeling of objects in high-resolution aerial images. The task of the region filtering and merging module is to examine the pipeline candidate regions to discard the false detections and merge the remaining regions. The pipeline, see Figure 3, resize and split the original image for the inference stage. Then, the coordinates of the candidate regions are rescaled to lie within the image domain boundaries. In the rest of this section we provide a detailed description of these steps.

A. Image Rescale

Algorithms for region detection of high-resolution aerial images need to deal with the huge scale variation of the target objects. For instance, as shown in Figure 4 in the xView benchmark the size of the objects range from \(4 \times 4\) to \(3299 \times 3132\) pixels.

Therefore, we rescaled the input image based into three classes of object size:
- small objects
- medium objects
- large objects

Specifically, we downscaled the image to detect some instances of medium and large objects and upscaled the image to detect some instances of medium and small objects. We also used the original scale to detect the three classes. The parameter settings is detailed in Section III-C.

![Framework flowchart: the input image is fed to five detection and classification pipelines and the outcomes are filtered and merged.](image-url)
**B. Image Splitting**

The convolutional neural network rescales the image to a fixed size for training and inference. However, the rescaling transformation may destroy the object details and compromise the detection and classification. Therefore, we split the input image into pieces with the same size and with the exact dimensions expected by the network, so as to ensure that objects will not be distorted. Furthermore, as an attempt to recover those objects that are divided by region splitting, we also use regions with overlap in two of the pipelines. See Figure 5.

**C. Inference**

We used for inference the baseline models provided by the xView team [7]. These models were trained with the Single Shot Multibox Detector (SSD) [8] algorithm by using the Inception Network [9] and two different techniques of image splitting for training: single resolution (SR) and multiple resolution (MR). For SR the original image was split into slices of \(300 \times 300\) pixels, while in MR strategy they split the original image several times by using three different partitions \(300 \times 300, 400 \times 400\) and \(500 \times 500\).

We used in our pipeline a SR model fine-tuned from the baseline, with a dropout of 20% and by using 80% of the training samples for parameter optimization and 20% for validation.

**D. Region Rescaling**

The image scaling step changes the image domain. Therefore, it is necessary to map the coordinates of the detected regions \((\hat{x}, \hat{y})\) into the coordinates \((x, y)\) of the original pipeline.
The proposed framework is composed by a set of convolutional neural networks, as a consequence, some object regions may be detected multiple times. Traditionally, regions with low confidence score are filtered out by using a fixed threshold \( \lambda \) and, for the remaining regions, a greedy non-maximum suppression (NMS) algorithm is used to discard the region hypotheses supposed to belong to the same object.

A standard greedy NMS algorithm, as described by Felzenszwalb [10], initially sorts the set of detections \( \mathcal{R} = \{r_1, r_2, \ldots, r_n\} \) by the confidence score. Then, it selects a region \( r_i \in \mathcal{R} \) with highest confidence score and loops through \( \mathcal{R} \), grouping other regions \( r_j \) that have an intersection score greater than a given threshold \( \sigma \), that is,

\[
I(r_i(b), r_j(b)) = \frac{\text{area} \ (r_i(b) \cap r_j(b))}{\text{area} \ (r_i(b))} > \sigma \tag{2}
\]

The algorithm ends up by partitioning \( \mathcal{R} \) into \( k \) subsets \( \mathcal{R}_1, \mathcal{R}_2, \ldots, \mathcal{R}_k \) of overlapping regions, and it outputs \( k \) regions

\[
D = \{r_1, r_2, \ldots, r_k\} \tag{3}
\]

by selecting within each subset the region with highest confidence score.

In our framework, the merging algorithm also produces \( k \) regions

\[
D = \{\bar{r}_1, \bar{r}_2, \ldots, \bar{r}_k\} \tag{4}
\]

however, instead of discarding many overlapping regions with high confidence score, we merge the axis-aligned region rectangles within each subset \( \mathcal{R}_i \) considering a weighted average criterion, that is,

\[
\bar{r}_i(b) = \frac{\sum_{r \in \mathcal{R}_i} r(w) \times r(b)}{\sum_{r \in \mathcal{R}_i} r(w)} \tag{5}
\]

where \( r(w) \times r(b) = r(w) \times (x_1, y_1, x_2, y_2) \). See Figure 6.

A key aspect in our merging algorithm is the intersection over union metric to compute the region intersection, namely

\[
\text{IoU}(r_i(b), r_j(b)) = \frac{\text{area} \ (r_i(b) \cap r_j(b))}{\text{area} \ (r_i(b) \cup r_j(b))} > \sigma \tag{6}
\]

The IoU metric, as opposed to the intersection score, takes into account the total area of both regions. This is particularly interesting for the xView challenge, where in many cases a significant intersection of objects does not mean that they should be merged, see Figure 7.

### III. Experiments

#### A. Dataset

The xView dataset, as detailed by Darius et al. [7], contains 1,413 high-resolution images, with image area ranging from 2564 \( \times \) 2576 to 3187 \( \times \) 4994 pixels, spanning approximately one million objects from 60 categories. The dataset was split into training, evaluation and testing subsets, as shown in Table I.

| Subset    | #images | #regions | #small | #medium | #large | #common | #rare |
|-----------|---------|----------|--------|---------|--------|---------|-------|
| Train     | 847     | 601,345  | 256,793| 333,406 | 11,659 | 595,149 | 6,709 |
| Eval.     | 282     | 200,291  | —      | —       | —      | —       | —     |
| Test      | 284     | —        | —      | —       | —      | —       | —     |
| Total     | 1,413   | 800,636  | —      | —       | —      | —       | —     |

The xView contest divided the dataset into three classes of object size in order to report the algorithm’s performance:

- **Small** objects: passenger-vehicle, small-car, bus, pickup-truck, utility-truck, truck, cargo-truck, truck-tractor, trailer, truck-tractor-w-flatbed-trailer, crane-truck, motorboat, dump-truck, scraper-tractor, front-loader-bulldozer, excavator, cement-mixer, ground-grader and shipping-container.
- **Medium** objects: fixed-wing-aircraft, small-aircraft, helicopter, truck-tractor-w-box-trailer, truck-tractor-w-liquid-tank, railway-vehicle, passenger-car, cargo-container-car, flat-car, tank-car, locomotive, sailboat, tugboat, fishing-vessel, yacht, engineering-vehicle, reach-stacker, mobile-crane, haul-truck, hut-tent, shed, building, damaged-building, helipad, storage-tank, pylon and tower.

- **Large** objects: passenger-cargo-plane, maritime-vessel, barge, ferry, container-ship, oil-tanker, tower-crane, railway-vehicle, flat-car, tank-car, locomotive, maritime-vessel, sailboat, tugboat, barge, ferry, yacht, container-ship, oil-tanker, engineering-vehicle, tower-crane, container-crane, reach-stacker, straddle-carrier, mobile-crane, haul-truck, scraper-tractor, cement-mixer, ground-grader, aircraft-hangar, helipad, pylon and tower;

- **Rare** objects: fixed-wing-aircraft, small-aircraft, helicopter, truck-tractor-w-liquid-tank, crane-truck, railroad-vehicle, flat-car, tank-car, locomotive, maritime-vessel, sailboat, tugboat, barge, ferry, yacht, container-ship, oil-tanker, engineering-vehicle, tower-crane, container-crane, reach-stacker, straddle-carrier, mobile-crane, haul-truck, scraper-tractor, cement-mixer, ground-grader, aircraft-hangar, helipad, pylon and tower;

- **Common** objects: passenger-cargo-plane, passenger-vehicle, small-car, bus, pickup-truck, utility-truck, truck, cargo-truck, truck-tractor-w-box-trailer, truck-tractor, trailer, truck-tractor-w-flatbed-trailer, passenger-car, cargo-container-car, motorboat, fishing-vessel, dump-truck, front-loader-bulldozer, excavator, hut-tent, shed, building, damaged-building, facility, construction-site, vehicle-lot, storage-tank, shipping-container-lot and shipping-container-lot.

Another division of the same objects considered their presence in the dataset:

- **Rare** objects: fixed-wing-aircraft, small-aircraft, helicopter, truck-tractor-w-liquid-tank, crane-truck, railroad-vehicle, flat-car, tank-car, locomotive, maritime-vessel, sailboat, tugboat, barge, ferry, yacht, container-ship, oil-tanker, engineering-vehicle, tower-crane, container-crane, reach-stacker, straddle-carrier, mobile-crane, haul-truck, scraper-tractor, cement-mixer, ground-grader, aircraft-hangar, helipad, pylon and tower;

- **Common** objects: passenger-cargo-plane, passenger-vehicle, small-car, bus, pickup-truck, utility-truck, truck, cargo-truck, truck-tractor-w-box-trailer, truck-tractor, trailer, truck-tractor-w-flatbed-trailer, passenger-car, cargo-container-car, motorboat, fishing-vessel, dump-truck, front-loader-bulldozer, excavator, hut-tent, shed, building, damaged-building, facility, construction-site, vehicle-lot, storage-tank, shipping-container-lot and shipping-container-lot.

B. **Hardware Time restrictions**

The participants needed to submit their solutions in the xView system by using a docker container with the inference source code, trained models and required packages. The solutions also ran inferences for the validation set by respecting the following hardware limitations:

- The inference for an input image must be completed in less than 40 minutes.
- Evaluating the entire validation set should not take more than 72 hours.
- The inference process needs to use a cluster of Central Processing Units (CPUs), with a memory limit of 8 GB.

The xView challenge used the validation set, with known images but unknown regions labels, to provisionally rank the participants solutions. The final ranking was determined by the performances in a sequestered test dataset.

C. **Settings**

Table II shows the parameter configuration for each pipeline in terms of image scaling, region overlap (horizontal and vertical directions), the confidence threshold for region filtering, the baseline trained model [7] used for inference and the objects to be detected defined by the category size.

| Parameters | Value |
|------------|-------|
| Pipeline 1 | Image scaling | 1.0 |
| | Region overlap | 0 pixels (no-overlap) |
| | Confidence threshold | 0.15 |
| | Classification model | Vanilla (SR) |
| | Object of interest (by size) | Small and medium |
| Pipeline 2 | Image scaling | 1.3 |
| | Region overlap | 0 pixels (no-overlap) |
| | Confidence threshold | 0.06 |
| | Classification model | Vanilla (SR) |
| | Object of interest (by size) | Small and medium |
| Pipeline 3 | Image scaling | 0.7 |
| | Region overlap | 100 pixels |
| | Confidence threshold | 0.5 |
| | Classification model | Multires (MR) |
| | Object of interest (by size) | Medium and large |
| Pipeline 4 | Image scaling | 1.0 |
| | Region overlap | 100 pixels |
| | Confidence threshold | 0.06 |
| | Classification model | Multires (MR) |
| | Object of interest (by size) | Small, medium and large |
| Pipeline 5 | Image scaling | 0.6 |
| | Region overlap | 0 pixels (no-overlap) |
| | Confidence threshold | 0.06 |
| | Classification model | Multires (MR) |
| | Object of interest (by size) | Large |

D. **Results**

The primary quantitative criteria used by the xView challenge for ranking purposes, is the interpolated mean average precision (mAP) metric, detailed by Henderson and Ferrari [11]. Informally, this metric sorts the predicted rectangles by the confidence score, in descending order, and then, if the intersection over union (IOU) metric is above 0.5 for a pair of predicted and groundtruth regions, then we have a true positive, otherwise, the matching is considered a false positive — undetected groundtruth regions are considered false negatives. The mAP performances for xView were computed by the challenge submission system, as shown in Table III.

| Proposed framework | Mean average precision (mAP) |
|--------------------|-----------------------------|
| Vanilla (SR)       | 20.87                       |
| Multires (MR)      | 18.14                       |

TABLE III

The mean average precision (mAP) score for the xView validation subset.
IV. DISCUSSION

We outline here some ideas that we tried, but did not result in good performance. Experiments with other popular deep learning approaches for object detection and classifications such as Faster-RCNN, SSD, RetinaNet and You Only Look Once version 3 (YOLOv3) [12], resulted in only meager performance increases.

Strategies that typically go with deep learning approaches such as optimization techniques, data augmentation, dropout, using different feature extraction architectures and varying learning rates were applied with limited success.

The predictions before and after post-processing stage were significantly high, for example, the number of predictions made on a subset of the training data was at least five times more than the number of ground truth in which a good number were false detections. Reducing this number using by thresholding led to significant increase in accuracies, however, a better approach might lead to a higher gain.

Input augmentation during inference also appeared to be a technique that could enable better detection and classification, for the challenge we successfully utilized zoom, however, horizontal flips was not as effective and probably other forms of augmentation may yield better results.

V. ONGOING RESEARCH

A rich source of structural information, that could be exploited to improve the classification, is the topological spatial relationship inherent to many classes of objects in the aerial imagery context. A graph formed by such relationships is shown in Figure 9, where the vertices are the object regions and the edges represent the shortest distance.

The spatial context for the xView benchmark is depicted by the co-occurrence matrix, shown in Figure 10, where one can easily see some clusters of objects that are part of the same scene shot.

We are working to use this spatial graph to filter out the false positives regions. Specifically, to verify if such neighborhood relations make sense in the real world one can use the training set or semantic networks such as ConceptNet [13] or geographical statistics from OpenStreetMap [14] and change the classifications based on this prior knowledge.

VI. ACKNOWLEDGEMENT

We appreciate the discussions with our colleagues Sathyanarayanan Aakur, Mauricio Pamplona Segundo, and Daniel Sawyer as we were developing this approach.

REFERENCES

[1] G.-S. Xia, X. Bai, J. Ding, Z. Zhu, S. Belongie, J. Luo, M. Datcu, M. Pelillo, and L. Zhang, “DOTA: A Large-Scale Dataset for Object Detection in Aerial Images,” in IEEE Computer Vision and Pattern Recognition (CVPR), 2018.
[2] R. D. Eastman, J. L. Moigne, and N. S. Netanyahu, “Research issues in image registration for remote sensing,” in IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2007, pp. 1–8.
[3] J. Porway, K. Wang, B. Yao, and S. C. Zhu, “A hierarchical and contextual model for aerial image understanding,” in EEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2008, pp. 1–8.
[4] F. Bastani, S. He, M. Alizadeh, H. Balakrishnan, S. Madden, S. Chawla, S. Abbar, and D. DeWitt, “RoadTracer: Automatic Extraction of Road Networks from Aerial Images,” in Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, June 2018.
[5] A. Van Etten, D. Lindenbaum, and T. M. Bacastow, “Spacenet: A remote sensing dataset and challenge series,” arXiv preprint arXiv:1807.01232, 2018.
[6] G. Christie, N. Fendley, J. Wilson, and R. Mukherjee, “Functional Map of the World,” in Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, Utah.
[7] D. Lam, R. Kuzma, K. McGee, S. Dooley, M. Laielli, M. Klaric, Y. Bulatov, and B. McCord, “xView: Objects in Context in Overhead Imagery,” CoRR, 2018. [Online]. Available: http://arxiv.org/abs/1802.07856
[8] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, “SSD: Single Shot Multibox Detector,” in European Conference on Computer Vision (ECCV). Springer, 2016, pp. 21–37.
C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, “Going deeper with convolutions,” in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, “Object detection with discriminatively trained part-based models,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 9, pp. 1627–1645, Sep. 2010. [Online]. Available: [http://dx.doi.org/10.1109/TPAMI.2009.167](http://dx.doi.org/10.1109/TPAMI.2009.167)

P. Henderson and V. Ferrari, “End-to-end training of object class detectors for mean average precision,” *CoRR*, vol. abs/1607.03476, 2016. [Online]. Available: [http://arxiv.org/abs/1607.03476](http://arxiv.org/abs/1607.03476)

J. Redmon and A. Farhadi, “Yolo3: An incremental improvement,” *arXiv preprint arXiv:1804.02767*, 2018.

H. Liu and P. Singh, “ConceptNet - A practical commonsense reasoning tool-kit,” *BT Technology Journal - Springer*, vol. 22, no. 4, pp. 211–226, Oct 2004.

M. Haklay and P. Weber, “Openstreetmap: User-generated street maps,” *IEEE Pervasive Computing*, vol. 7, no. 4, pp. 12–18, 2008.

![Fig. 10. Spatial context: the co-occurrence matrix and six clusters of objects. Group 1: fixed-wing aircraft, helipad, helicopter, small-aircraft, aircraft-hangar and passenger cargo-plane. Group 2: sailboat, fishing-vessel, motorboat, yacht, maritime-vessel, tugboat, barge, ferry, container-ship, oil-tanker. Group 3: container-crane, reach-stacker, shipping-container, mobile-crane, shipping-container-lot and truck-tractor-w-flatbed-trailer. Group 4: building, small-car, bus, truck, cargo-truck, vehicle-lot, utility-truck, truck-tractor-w-box-trailer. Group 5: dump-truck, construction-site, excavator and front-loader-bulldozer. and Group 6: tank-car, cargo-container-car, passenger-car, locomotive.](image-url)