

Cascaded Zoom-in Detector for High Resolution Aerial Images

Akhil Meethal

Eric Granger

Marco Pedersoli

LIVIA lab, Dept. of Systems Engineering, ÉTS Montreal, Canada

akhilpm135@gmail.com, {marco.pedersoli, eric.granger}@etsmtl.ca

Abstract

Detecting objects in aerial images is challenging because they are typically composed of crowded small objects distributed non-uniformly over high-resolution (in terms of pixel size) images. Density cropping is a widely used method to improve this small object detection where the crowded small object regions are extracted and processed in high image-resolution. However, this is typically accomplished by adding other learnable components, thus complicating the training and inference over a standard detection process. In this paper, we propose an efficient Cascaded Zoom-in (CZ) detector that re-purposes the detector itself for density-guided training and inference. During training, density crops are located, labeled as a new class, and employed to augment the training dataset. During inference, the density crops are first detected along with the base class objects, and then input for a second stage of inference. This approach is easily integrated into any detector, and creates no significant change in the standard detection process, like the uniform cropping approach popular in aerial image detection. Experimental results on the aerial images of the challenging VisDrone and DOTA datasets verify the benefits of the proposed approach. The proposed CZ detector also provides state-of-the-art results over uniform cropping and other density cropping methods on the VisDrone dataset, increasing the detection mAP of small objects by more than 3 percentage points.¹

1. Introduction

With the advent of deep learning, object detection methods achieved significant progress [2, 4, 21, 28, 29, 35]. This has resulted in the fast growth of their adoption to many downstream applications, including aerial images from drones or satellites, for earth monitoring, surveillance, inspection, etc [6, 15, 17, 26, 37]. However, unlike natural images in the Pascal VOC [12] and MS-COCO [23] datasets, aerial images are captured in high image-resolution, and are

¹Our code is available at: <https://github.com/akhilpm/DroneDetectron2>

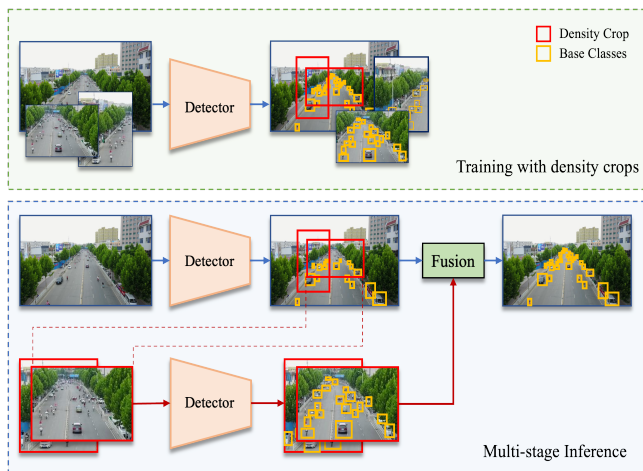


Figure 1. Overview of our proposed Cascaded Zoom-in detector. During training (top), density crops are extracted, and labeled as a new class (red boxes) on the original image. The training set is augmented with the rescaled density crops and the corresponding ground truth boxes within these crops. During the first stage of inference (bottom), the base class objects and density crops (red boxes) are detected on the whole image. In the second stage, the density crops are rescaled to a common larger size, and a second inference is performed. Finally, the detections on density crops are combined with the detections on the whole image.

typically comprised of many small objects, that are sparsely distributed in crowded object regions. As a comparison, the average number of objects in Pascal VOC and MS-COCO images are 3 and 7, respectively, whereas images in the VisDrone [43] and DOTA [37] datasets – two popular benchmarks in the aerial detection community – have an average number of 53 and 67 objects, respectively. The average width of Pascal VOC and MS-COCO images are 500 and 640 pixels, respectively, while the same in VisDrone and DOTA images are 1500 and 4000 pixels, respectively. Therefore, improvements observed in object detection methods applied to natural images do not easily translate to object detection in high-resolution images from drones and satellites.

The high-resolution imagery and tiny objects raise sev-

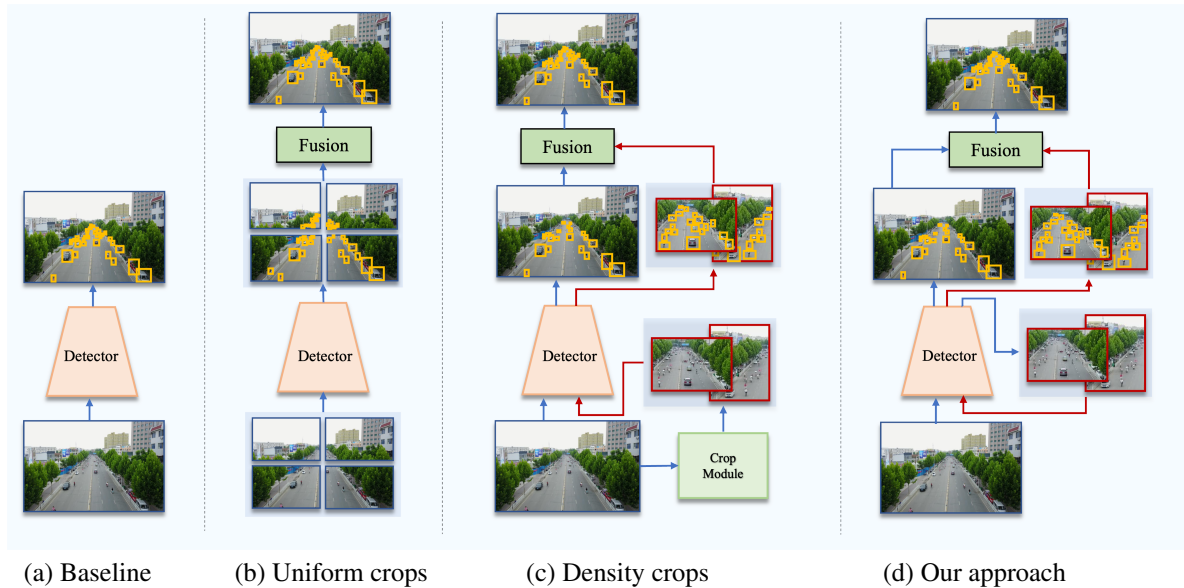


Figure 2. An illustration of different approaches for the detection of small objects in high-resolution aerial images. (a) The image is down-scaled and processed at the detector’s input size. (b) The image is split into uniform, possibly overlapping crops, and each crop is processed by the detector. (c) An external learnable module crops the image into dense object regions. Each crop is re-scaled and processed at the detector’s input size. (d) Our proposed CZ detector is re-purposed to detect the density crops along with the base class objects, eliminating the need for an external module. Each crop is re-scaled and processed at the detector’s input size in a second stage of inference. Blue arrows show the path of the original image and red shows the path of density crops.

eral challenges for object detection in aerial images, including the loss of information due to rescaling and feature down-sampling, low tolerance to bounding box shifts, and noisy feature representations [5, 8] among others. Since it is difficult to input high-resolution aerial images directly to a detector due to the computational cost and large memory footprint, they are often resized to the standard input size range of 300-512 pixels (see Fig. 2 (a)). This rescaling, coupled with the feature down-sampling in ConvNets, often results in feature representations linked to small objects diminished or corrupted by the noisy background activations [38]. Regarding the tolerance to bounding box shifts, a small shift in the bounding box may cause a large decline in the Intersection over Union (IoU), raising the false positive detections [5].

To utilize the higher resolution, and also mitigate the information loss, a popular approach consists in cropping the input image into uniform patches, and then performing object detection on these patches in high image-resolution by upsampling (see Fig. 2 (b)). Although these uniform patches help to improve the accuracy, this approach does not respect the distribution of the objects in the image, and hence the scale normalization achieved is not optimal [18, 39]. As the objects in aerial images usually appear crowded in sparsely distributed regions of the image, it is desirable to perform density-based cropping, and then process high-resolution versions of those crowded object re-

gions for better scale balance.

To extract these density crops, existing methods utilize additional learnable modules (see Fig. 2 (c)) such as density maps [18], cluster proposal networks [39], global-local detection pipeline [8] etc. This usually results in additional learnable components in the pipeline, often with multiple stages of training. Even with single-stage end-to-end methods, the crops obtained are noisy in the beginning and are only useful for aiding small object detection in the later stages of the training [39]. Moreover, as the learning is evolving the crops also evolve, so the model is not receiving a consistent indication of what exactly is a “density crop”. Thus, practitioners still widely use uniform cropping over the advanced density crop-based approaches due to the practical simplicity it offers [33, 34].

In this paper, we tried to bridge this gap between research and practice by proposing an efficient cascaded zoom-in (CZ) detection method that can leverage density crops within the training of a standard detector, offering the simplicity of uniform cropping, yet providing the benefits of density crops (see Fig. 2 (d)). We simply make use of the existing detector itself to discover the density crops, by adding the “crop” as a new class to the detector. The crops are labeled as a pre-processing step using a crop labeling algorithm, and hence the detector receives a consistent signal of what constitutes a crop. During inference, while other methods require complex post-processing to filter the noisy

crops, we can simply perform it based on the confidence of the "crop" class from the detector.

Fig. 1 illustrates the training and testing of our method. First, the density crops are extracted from each training image as a pre-processing step, using our crop labeling algorithm. These density crops are added as a new class to be detected in the corresponding image. Then we augment the training set with the higher resolution version of the density crops, and the corresponding ground truth (GT) boxes of objects inside the crop. Then, the detector is trained as usual. This training process has an almost negligible overhead over standard detector training, and it is similar to that of uniform crop based training. Inference is performed in two stages. In the first stage, the base class objects and density crops are detected from each input image. In the second stage, high-quality density crops are selected based on their confidence score, and another inference based is performed on an up-sampled version of these crops. Finally, the detections from stages one and two are fused to get the output detection. Compared to standard object detector learning, the extra work required at training time is the crop labeling which can be performed as a pre-processing step. While making predictions, the extra work required is one more inference. As both of these processes don't require any significant modification on a normal object detection pipeline, similar to uniform crops, our method can be easily incorporated for accelerating small object detection.

Our main contribution can be summarized as follows:

(1) An efficient cascaded zoom-in (CZ) object detector based on density cropping is proposed for high-resolution aerial images, where a given detector is re-purposed to extract density crops (thus not using additional modules) along with base class objects. At train time, this approach relies on a simple pre-processing step for density crop labeling, and at test time, an additional inference step is needed to process the predicted density crops.

(2) We empirically validate the benefits of our cascaded zoom-in detection on aerial images from drones (VisDrone) and satellites (DOTA), and obtain state-of-the-art performance. A consistent improvement in the detection accuracy is observed on both datasets over the baseline detection. Particularly, the mAP of small object detection is improved by more than 3 percentage points.

2. Related works

Object Detection. General purpose visual object detectors are built primarily focusing on natural images. They can be broadly classified into two-stage [21, 29] and one-stage [24, 28] object detectors. Two-stage object detectors first extract potential object regions (*i.e.*, object proposals) in the first stage. Then in the second stage, the proposal regions are classified into object categories. Earlier approaches extract these RoIs using low-level image features [13, 14],

etc. Later, a learnable component called RPN (Region Proposal Network) is proposed for object proposal extraction, giving birth to end-to-end two-stage detectors [7, 21, 29]. One-stage object detectors in contrast avoid the RoI extraction stage and classify and regress directly from the anchor boxes. They are generally fast and applicable to real-time object detection [22, 24, 27, 28]. But generally speaking, two-stage detectors are more accurate and hence used more in aerial detection [8, 18]. Recently, one-stage detectors are being explored in aerial images [38]. These days, anchor-free detectors [10, 35] is getting popular, since they avoid the need for hand-crafted anchor box dimensions and their matching process, which is often specifically tuned depending on the size distribution of the objects. We performed our empirical study mostly on the two-stage detector for a fair comparison with existing approaches, but also report results on the modern anchor-free one-stage detector to validate the generality of our approach.

Detection of Small Objects. Small object detection is studied extensively in the literature. Cheng et al. [5] provides a comprehensive survey of small object detection techniques which is broadly classified into: scale aware training [3, 11, 20, 21, 25, 31, 32, 41], super-resolution methods [19], context modeling [1], and density guided detection [9, 18, 39] among others. Due to the crowded distribution of small objects at sparse locations in the high-resolution images, density cropping is a popular strategy in aerial image object detection [8, 9, 18, 39]. But this is usually obtained by additional learnable components and multi-stage training. Thus, such methods are more complex to use than uniform cropping. We also use the density cropping strategy but re-purpose the detector itself for extracting the crops, so no additional learnable components are needed. Hence, we believe that our method has the practical simplicity of the uniform cropping strategy, yet retains the benefit of density guided detection.

3. Proposed Method

This section provides details on how to transform any detector such that it can efficiently process high-resolution images with many small target objects, leveraging density crops. Our proposed cascaded zoom-in (CZ) object detector allows us to re-purpose the detector to extract density crops along with base class objects. Let us consider the *original image*, which is kept at its high image-resolution, the *down-sampled image*, which is an image containing the same view of the original, but down-scaled to the detector input size, and the *cropped images*, which are the selected regions of the image that are up-scaled to the detector input size. The first component of our pipeline is the density crop labeling algorithm that labels the crowded object regions as "density crops" and augments the training data by adding up-scaled versions of those regions. Then, the

density crops are also added to the original image as a new class to be detected. Thus, the augmented training set will include high-resolution versions of the regions containing crowded small objects, allowing the detector to see those small objects in high image-resolution. Once the model is trained, the inference consists of the detection of the base classes and "density crop" class on the original image, and then a second detection on the predicted density crops after up-scaling. Finally, the detection from the original image and density crops are combined to produce the output detection.

3.1. Training with Density Crops

In order to use a standard detector for our approach, we need to add a new class that we call "density crop" to the training annotations. In this way, our approach is detector agnostic (as we don't change the internals of the detector, we just add one additional class to the list of target classes) and does not require any additional component than the detector itself. The density crop class should label those parts of the image that contains many small objects and include them in a bounding box. This will allow training and inference to focus on those parts by analyzing them in higher image-resolution. Several different ways could be considered for defining the density crop. The quality constraints we used to define density crops are: (i) they should enclose groups of small target objects, (ii) they are easy to localize at inference time, and (iii) they are optimal in number to reduce the computational cost.

Algorithm 1 describes the procedure we used for discovering and labeling density crops from the groundtruth (GT) annotations. In summary, we perform an iterative merging of the GT boxes to discover the density crops. In the first step, all GT boxes \mathcal{B} are scaled by expanding the min and max coordinates of the boxes by σ pixels ($\text{scale}(\mathcal{B}, \sigma)$). Then we calculate the pairwise Intersection over Union (IoU) between the scaled boxes ($\text{pairwise_IoU}(\mathcal{D})$) in O as a $|\mathcal{D}| \times |\mathcal{D}|$ matrix. Connections are labeled in C by assigning one to all overlap values above a threshold θ in the pairwise IoU matrix O . Then we select in C the row m^* with the maximum number of connections. An enclosing box is computed ($\text{enclosing_box}(C_{m^*})$) by finding the min and max coordinates of all crop members connected to m^* . The newly obtained crop box is added to the list of crops and the row C_{m^*} is set to zeros. Subsequently, the crops that are bigger than a maximum threshold π are removed from the list \mathcal{D} ($\text{filter_size}(\mathcal{D}, \pi)$). This procedure of iterative merging is performed N times. The crop size threshold π used here is the ratio of the area of the crop to that of the image.

The quality of the crops is important for our method. It is in fact the iterative merging that brings out the best quality crops. Naive scaling and merging to find the maximum

Algorithm 1: Density Crop Labeling Algorithm.

Input: \mathcal{B} : GT boxes in an image
Output: \mathcal{D} : Density crops
Parameters: N : no. of merging steps,
 σ : expansion pixels,
 θ : overlap threshold,
 π : maximum crop size

1. $\mathcal{D} \leftarrow \text{scale}(\mathcal{B}, \sigma)$;
2. **for** $i \leftarrow 1$ **to** N **do**
 - a) $O = \text{pairwise_IoU}(\mathcal{D})$
 - b) $C = O > \theta$ (connection matrix)
 - c) $\mathcal{D} \leftarrow \emptyset$
 - while** $|C| > 0$ **do**
 - i) $m^* = \arg \max_m (\sum_i C_{m,i})$
 - ii) $d = \text{enclosing_box}(C_{m^*})$
 - iii) $\mathcal{D} \leftarrow \mathcal{D} + d$
 - iv) $C_{m^*} = 0$
 - end**
 - d) $\mathcal{D} \leftarrow \text{filter_size}(\mathcal{D}, \pi)$

end

enclosing boxes based on pairwise IoU results in either bad crops or too many small crops (with fewer objects in them) depending on the value of the scaling factor. Iterative merging produces good-quality crops enclosing groups of small objects respecting the quality constraints. In section 4, we present the ablation studies validating the effectiveness of our density crop labeling algorithm. We also show that hyperparameters of the algorithm can be easily set. Note that by labeling the density crops apriori, we are giving the detector a consistent signal of what constitutes a "density crop" throughout the training period, unlike other methods where density crops are also evolving while the training is progressing [9, 18, 39]. This is consistent with the observations in [42] that simple heuristic methods that can give a consistent groundtruth label during training are superior to complex prediction-based groundtruth labels that are continuously evolving.

With the newly obtained crop labels, we can also augment the training set with additional image crops. The original image and its annotations \mathcal{B} are down-scaled using the maximum training size $W \times H$. Note that it is expected the detector will not detect many small objects in the down-scaled image. But the augmented up-scaled version of the density crop $d \in \mathcal{D}$ of a given image will have those small objects that fall inside the crop in a higher pixel size. This will reduce the extreme scale variation at training time. We used bilinear interpolation for up-scaling the density crops. The crop labeling can be performed as a pre-processing step. The up-scaled version augmentation of density crops is simply a data augmentation process. Thus our method is not introducing any change in the standard training pipeline

of a detector, except the new class "density crop" is added. In this regard, it is practically easy to use like uniform cropping.

3.2. Multi-stage Inference

As the detector is trained to recognize density crops, at inference time, we can get the density crop from its prediction itself. Figure 1 bottom explains our inference process in detail. It consists of two stages. In stage one, it predicts the base class objects and density crops on the input image. Then we select the high-quality density crops based on their confidence score. In stage two, the upscaled density crops are passed through the same detector again, producing small object detection on the density crops. Finally, we re-project the detections on the crops to the original image and concatenate them with the detections on the original image. Let $c \in \mathcal{C}$ be an up-scaled crop image of size (I_c^W, I_c^H) defined by its bounding box coordinates $(c_{x1}, c_{y1}, c_{x2}, c_{y2})$ in the original image. Given the scaling factors $(S_c^W, S_c^H) = (\frac{c_{x2}-c_{x1}}{I_c^W}, \frac{c_{y2}-c_{y1}}{I_c^H})$, the re-projection box p_i scales down and shifts the detection boxes $(x_{1,i}, y_{1,i}, x_{2,i}, y_{2,i}) \in \mathcal{B}^c$ in the crop c as:

$$p_i = (S_c^W x_{1,i}, S_c^H y_{1,i}, S_c^W x_{2,i}, S_c^H y_{2,i}) + (c_{x1}, c_{y1}, c_{x1}, c_{y1}) \quad (1)$$

The Non-Maximal Suppression(NMS) is then applied to remove duplicate detections.

While other methods need complex post-processing to filter the noisy crops [39], we can simply use the confidence score of the density crops to do the same. Stage one of the inference is the standard inference procedure in any detector. The filtering of the noisy crops can be easily performed with the confidence scores given by the detector. The second stage of the inference is performed with the same detector, but a different input (the up-scaled density crops). So, we are simply repeating the standard inference procedure of a detector one more time. All of these operations can be easily wrapped on top of the inference procedure of any detector, thus keeping the simplicity of the uniform cropping approach at inference time too.

4. Experiments

Datasets and evaluation measures. For evaluation of methods, we employed two popular challenging benchmark datasets for Aerial Image Detection, namely the VisDrone [43] and DOTA [37] datasets. The measure used for assessing and comparing the performance of methods is COCO style average precision (AP) [23]. The AP of small, medium and large objects are also reported, particularly to understand the performance of our method for small object detection. Finally, the number of frames per second (FPS) is reported as a measure of time complexity.

VisDrone. This dataset contains 8,599 drone-captured images (6,471 for training, 548 for validation, and 1,580 for testing) with a pixel size of about 2000×1500 pixels. The objects are from ten categories with 540k instances annotated in the training set, mostly containing different categories of vehicles and pedestrians observed from drones. It has an extreme class imbalance and scale imbalance making it an ideal benchmark for studying small object detection problems. As the evaluation server is closed now, following the existing works, we used the validation set for evaluating the performance.

DOTA. This dataset is comprised of satellite images. The images in this dataset have a pixel size ranging from 800×800 to 4000×4000 . Around 280k annotated instances are present in the dataset. The objects are from fifteen different categories, with movable objects such as planes, ships, large vehicles, small vehicles, and helicopters. The remaining ten categories are roundabouts, harbors, swimming pools, etc. Many density crop based detection papers reports results only on movable objects [39] with the assumption that immovable objects usually won't appear crowded. But they are also small objects, so we kept all classes to assess the improvement in small object detection. The training and validation data contain 1411 images and 458 images, respectively.

Implementation details. The Detectron2 toolkit [36] was used to implement our CZ detector. The backbone detector used in our study is primarily Faster RCNN [29], but we also show results on the modern anchor-free one-stage detector FCOS [35]. This validates our claim that our approach gives a consistent improvement in performance regardless of the detector used. We used Feature Pyramid Network (FPN) [21] backbone with ResNet50 [16] pre-trained on ImageNet [30] dataset for our experimental validation. For data augmentation, we resized the shorter edge to one randomly picked from (800, 900, 1000, 1100, 1200), and applied horizontal flip with a 50% probability. The model was trained on both datasets for 70k iterations. The initial learning rate is set to 0.01 and decayed by 10 at 30k and 50k iterations. For training, we used one NVIDIA A100 GPU with 40 GB of memory.

4.1. Comparison with Baselines

Table 1 presents a comparison between uniform cropping and density cropping on the VisDrone dataset, with and without the last feature map of the feature pyramid (P2), which has a strong impact in memory and computation [38]. For the uniform cropping, we crop the original image into 4 equal-sized crops by splitting at half height and width. In order to have a fair comparison, we use our method with a confidence threshold of 0.7 to obtain an average of 1-3 crops per image. The observations in the table suggest that uniform cropping improves performance com-

Settings	AP	AP ₅₀	AP ₇₅	AP _s	AP _m	AP _l	FPS
<i>Without P2</i>							
Baseline	29.48	51.68	29.55	22.33	38.66	39.30	26.31
Uniform crops	30.68	54.44	30.54	22.91	40.62	41.03	12.30
CZ Det. (ours)	33.02	57.87	33.09	25.74	42.93	41.44	11.64
<i>With P2</i>							
Baseline	30.81	55.06	30.68	23.97	39.19	41.17	18.25
Uniform crops	31.73	56.31	31.57	25.13	40.41	41.06	9.85
CZ Det. (ours)	33.22	58.30	33.16	26.06	42.58	43.36	8.44

Table 1. Comparison of detection performance between a baseline detector, uniform crops and density crops on VisDrone dataset (1.5K pixels). The results are in %. The small, medium, and large objects are grouped according to the coco evaluation protocol [23].

pared to vanilla training on the whole image, but it is still inferior to our density-based cropping. When high-resolution feature maps P2 are not used, density cropping gains more than 3.5 percentage points in AP and the AP of small objects is improved by 3.4 percentage points. It is worth noting that compared to uniform cropping, our approach introduces additional parameters to recognize one extra class and no changes in learning and inference dynamics. So this can be easily used as a plug-and-play replacement for the uniform crop-based training, popular among the community. In terms of frame rate, our approach is slightly slower than uniform crops. However, we observe that our method without the expensive P2 features performs better than uniform crops with P2, while also being faster. In Figure 3, a visual comparison of the highly confident detections between the baseline model and our density crop-based model is shown. When the density crops are used, we can observe an increase in the number of detections. It can be observed that more objects are getting discovered in the crop regions when the detection results from the second inference are augmented. This explains the impact of our zoom-in detector for small object detection in high-resolution images.

To further verify the observations, we repeated the same type of study in the satellite images of the DOTA dataset. In this dataset images are at higher pixel size (4k pixels), thus due to memory constraints, the baselines are already performing uniform cropping. Table 2 shows the results of a uniform cropping baseline and our CZ detector for two different configurations. Similar to VisDrone, significant improvement is seen in the case of not using high-resolution features P2, with a gain of 2.9 percentage points. APs of small and medium objects are improved by 3.0 and 4.0 percentage points respectively from the baseline without using high-resolution features. In terms of computation, we can see that, as expected, our approach has a slightly slower frame rate than the baseline. However, this is compensated by the higher detection accuracy. We see for instance that the best baseline with P2 features has an AP of 33.44% with an FPS of 0.49, while our CZ detector without P2 features has a higher AP (34.14%) while being also faster (0.62 FPS).

Settings	AP	AP ₅₀	AP ₇₅	AP _s	AP _m	AP _l	FPS
<i>Without P2</i>							
Baseline	31.29	51.57	33.10	12.69	34.04	42.83	0.93
CZ Det. (Ours)	34.14	56.69	35.69	15.66	38.16	44.20	0.62
<i>With P2</i>							
Baseline	33.44	54.03	35.56	16.86	36.76	43.65	0.49
CZ Det. (Ours)	34.62	56.86	36.17	18.17	37.84	43.83	0.30

Table 2. Performance comparison of our method against baselines on DOTA dataset (4k pixels). The results are in %.

Train	Test	AP	AP ₅₀	AP ₇₅	AP _s	AP _m	AP _l
		29.48	51.68	29.55	22.33	38.66	39.30
	✓	29.93	53.29	29.52	22.33	39.35	39.46
✓		32.64	57.36	32.78	24.81	43.04	41.07
✓	✓	33.02	57.87	33.09	25.74	42.93	41.44

Table 3. Detection results with and without density crops at train time and test time (results in %).

4.2. Ablation Studies

The effectiveness of our proposed CZ detector is characterized by ablation experiments on the VisDrone dataset. Additional studies showing the impact of hyperparameters in the crop labeling algorithm, different backbone networks for the detector, and an analysis of the detection errors are presented in the supplementary material.

a) Density Crops effect at Training and Inference. We used density crops at the training and test time to achieve the best performance. In particular, while training, the rescaled density crops are augmented with the training images; while testing we do the two-stage inference where stage one performs inference on the whole image and stage two performs inference on the density crops. In this section, we study the importance of this configuration. Table 3 shows the results. When the density crops are not augmented with the training set but only used in the two-stage inference, the improvement is marginal over the baseline (most importantly, AP_s has no change). This is because the scale imbalance in the input image is not mitigated as the detector is not seeing the small objects at a bigger scale. When density crops are added to the training set, the detection accuracy improves significantly. However, the inference is still happening on the whole image so the detection accuracy of small objects is affected. When inference is performed on the density crops and fused with the detection on the whole image, we get the best results.

b) Impact of the Quality of Crops. Figure 4 illustrates how the confidence of crops impacts the detection accuracy and the number of density crops extracted. The impact is studied for two settings, with and without the high-resolution features P2. This is to verify how the density crops aid small object detection with and without utilizing expensive high-resolution feature maps. The crop confidence, which is used as the proxy for crop quality, is varied from 0.1 to 0.9. In general, with lower confidence values, we are ob-

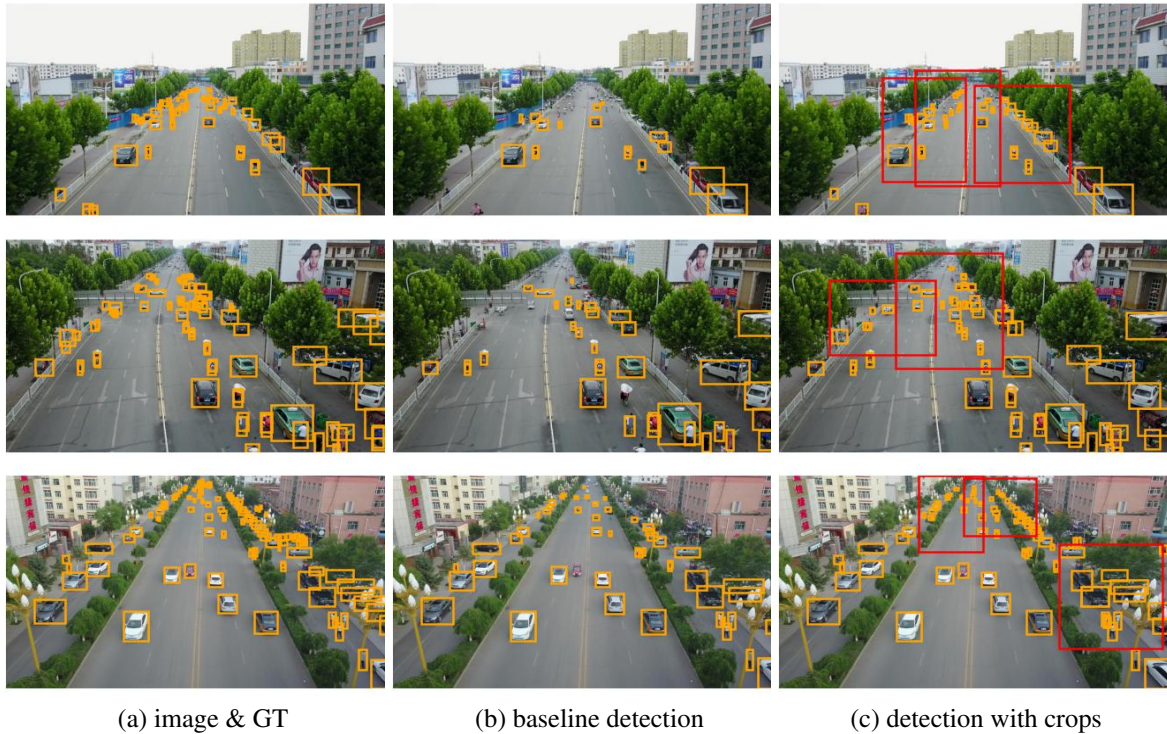


Figure 3. Visualization of density crop-based detection. (a) the original image and its GT. (b) detection with the baseline detector. (c) detection with density crops; the density crops are shown in red color. Our method detects more objects, especially inside the crop regions.

serving more crops but many of them are noisy and redundant even after Non-Maximal Suppression. So when the quality of the crops is low, the detection accuracy decreases (Figure 4 left). When the quality is increased, the accuracy increases until 0.7, and then it is gradually coming down as we use very few crops in that case. The trend is the same with and without P2.

From Tables 1 and 2, we observed that density crops obtained better gain in detection accuracy over the baseline without high-resolution features. Though this is expected, we decided to understand how exactly this is happening. We analyzed the number of density crops retained after filtering out the low-quality crops at multiple confidence levels ranging from 0.1 to 0.9. Figure 4 right shows the results with and without high-resolution features P2. It can be observed that for "without P2", we are getting more density crops at all confidence levels. With higher crop confidence levels, we get more high-quality crops for "without P2" case, hence we observe a better gain in detection accuracy over the baseline. We used a confidence of 0.7 in all our experiments to have the best trade-off between detection precision and speed. While other methods use post-processing on the crop detections [39] or density maps [18] to filter the noisy crops during inference, we can filter them out based on their confidence score simplifying the inference procedure.

c) Why Iterative Merging for Crop Discovery? Simply

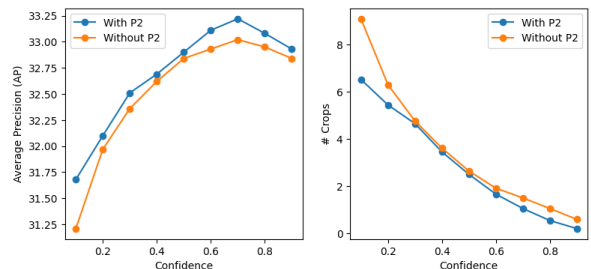


Figure 4. Change in detection precision and the number of crops according to crop confidence. The crop confidence is varied from 0.1 to 0.9. The crop confidence for best detection accuracy is 0.7.

scaling and doing a one-step merging operation to create density crops results in sub-optimal crops. We empirically verify this with multiple scaling strategies and argue that the iterative merging strategy is superior to them. Authors of [39] also used iterative crop merging on the output of their crop detection module to reduce the redundant crops. This has to be performed at training and test time to refine the initial crop detections. To label the crops for training, they used a single-step aggregation. Our iterative merging for labeling crops can be performed as a pre-processing step before training. We avoided redundant crops at inference, by filtering them out based on the confidence score.

Table 4 top provides the comparative results of single-

Scaling	# crops	AP	AP ₅₀	AP ₇₅	AP _s	AP _m	AP _l
Baseline	0	29.48	51.68	29.55	22.33	38.66	39.30
factor = 2.0	74417	24.39	44.38	23.73	15.96	34.96	47.24
3.0	67906	30.64	53.71	30.68	23.23	40.64	39.35
6.0	43300	31.30	55.33	31.42	23.79	41.06	38.40
8.0	34663	30.95	55.18	30.31	23.38	40.93	39.24
pixels = 30	62677	31.26	54.55	31.50	23.83	40.78	50.07
60	46753	31.98	55.84	32.07	25.12	41.11	45.52
90	35442	31.47	55.62	30.96	24.03	41.27	44.08
120	25146	31.07	54.74	30.95	23.18	41.43	42.39
Ours	14018	33.02	57.87	33.09	25.74	42.93	41.44

Table 4. Comparison of iterative merging strategy with single-step merging where GT boxes are scaled according to scaling factors, and scaled uniformly by pixel values (results in %).

step merging with our iterative merging strategy where GT boxes are scaled by a scaling factor. Using a low scaling factor creates too many crops, containing fewer objects. More specifically, it is producing multiple small crops containing fewer objects in crowded regions in the image. When the scaling factor is increased, the number of crops decreases and performance increases up to a point but declines later as the crops become too big and the object density of the image is less respected. This is because large scaling factors significantly blow up the big GT boxes and it alters the density of the crops. The detection performance obtained is also far below our iterative merging. Table 4 bottom shows the same comparison when GT boxes are scaled by constant pixel values. As this avoids the blowing of large bounding boxes due to the constant scaling, the detection performance is better than the former one. Iterative merging produces the optimal number of crops with the best performance. The scaling used in the iterative merging is small and only performed at the first stage of merging. We used 20 pixels as the scaling magnitude. Large values are not possible here since the `filter_size` operation while restricting the crop size will reduce the number of crops. Thus it is easy to set. In the supplementary material, we provide more experiments studying the ease of tuning the hyperparameters N , π , θ of the crop labeling algorithm, and the visualization of the crops obtained by different merging strategies.

4.3. Results with Other Detectors

To validate the effectiveness of our approach with other detection architectures, we conducted experiments on the modern anchor-free one-stage detector FCOS [35]. Table 5 shows the performance comparison of the vanilla FCOS detector with our density crop based FCOS detector. Similar to the results in Table 1, AP is improved by a significant margin, and AP_s has gained almost 5 percentage points. We can also see that density crop based FCOS has superior performance than their Faster RCNN counterpart in terms of most of the metrics. This is interesting because other density based approaches weren't producing better results with one-stage detectors than the two-stage ones.

Settings	AP	AP ₅₀	AP ₇₅	AP _s	AP _m	AP _l	FPS
Base FCOS	29.51	50.40	29.92	21.25	40.51	37.29	26.01
CZ FCOS Det.	33.67	56.20	34.15	26.16	43.98	46.87	12.69

Table 5. Results with anchor free detector FCOS on the Visdrone dataset (results in %). All results are without using P2.

Method	AP	AP ₅₀	AP ₇₅	AP _s	AP _m	AP _l
ClusterNet [39]	26.72	50.63	24.70	17.61	38.92	51.40
DensityMap [18]	28.21	47.62	28.90	19.90	39.61	55.81
CDMNet [9]	29.20	49.50	29.80	20.80	40.70	41.60
GLSAN [8]	30.70	55.40	30.00	-	-	-
QueryDet [38]	28.32	48.14	28.75	-	-	-
CascadeNet [40]	28.80	47.10	29.30	-	-	-
CascNet+MF [40]	30.12	58.02	27.53	-	-	-
CZ Det. (Ours)	33.22	58.30	33.16	26.06	42.58	43.36

Table 6. Performance of our proposed method compared against state-of-art approaches with Faster RCNN detector on the VisDrone validation set (results in %). "MF" stands for model fusion.

4.4. Comparison with State-of-the-Art Methods

Table 6 compares our approach with the existing methods on the VisDrone dataset. Similarly to us, some methods perform density cropping [8, 9, 18, 39], while QueryNet [38] and CascadeNet [40] use other approaches to improve the detection performance on aerial images. We obtained the best detection AP among the state-of-the-art methods. Only for large objects, DensityMap performs better than our approach. This is probably because our method gets biased to detect small objects, thanks to the additional crops on training. In fact, for small object detection, we obtained the best AP_s, significantly outperforming all existing approaches. AP_m also shows a good improvement of more than 2 percentage points.

5. Conclusion

We proposed an efficient method for utilizing density crops for aerial image detection. Our method is as simple to use as the uniform cropping approach widely used by practitioners. The training step simply adds an additional class called "density crop" to the detector whose labels are obtained from a crop labeling algorithm. The inference is performed in two steps, one on the original image and then on the up-scaled version of the high-quality crops detected from it. For both modifications, we re-purpose the original detector, thus alleviating the need for additional components, unlike existing approaches. Empirical results verify the superiority of our approach in terms of detection accuracy and ease of use. At present, the density crops are up-scaled to a fixed pixel resolution. In the future, we plan to up-scale adaptively, respecting the density of the crops.

Acknowledgments: This work was supported by the Natural Sciences and Engineering Research Council of Canada, and the Digital Research Alliance of Canada.

References

- [1] S. Bell, C. L. Zitnick, K. Bala, and R. Girshick. Inside-outside net: Detecting objects in context with skip pooling and recurrent neural networks. In *CVPR*, 2016. 3
- [2] Z. Cai and N. Vasconcelos. Cascade r-cnn: Delving into high quality object detection. In *CVPR*, 2018. 1
- [3] Z. CaiZhaowei, C. Fan, R. S. Feris, and N. Vasconcelos. A unified multi-scale deep convolutional neural network for fast object detection. In *ECCV*, 2016. 3
- [4] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko. End-to-end object detection with transformers. In *ECCV*, 2020. 1
- [5] G. Cheng, X. Yuan, X. Yao, K. Yan, Q. Zeng, and J. Han. Towards large-scale small object detection: Survey and benchmarks. <https://arxiv.org/abs/2207.14096>, 2022. 2, 3
- [6] G. Cheng, P. Zhou, and J. Han. Rfcd-cnn: Rotation-invariant and fisher discriminative convolutional neural networks for object detection. In *CVPR*, 2016. 1
- [7] Jifeng Dai, Yi Li, Kaiming He, and Jian Sun. R-fcn: Object detection via region-based fully convolutional networks. In *NeurIPS*, 2016. 3
- [8] S. Deng, S. Li, K. Xie, W. Song, X. Liao, A. Hao, and H. Qin. A global-local self-adaptive network for drone-view object detection. *IEEE Transactions on Image Processing (TIP)*, 2020. 2, 3, 8
- [9] C. Duan, Z. Wei, C. Zhang, S. Qu, and H. Wang. Coarse-grained density map guided object detection in aerial images. In *ICCVW*, 2021. 3, 4, 8
- [10] K. Duan, S. Bai, L. Xie, H. Qi, Q. Huang, and Q. Tian. CenterNet: Keypoint triplets for object detection. *arXiv:1904.08189v3*, 2019. 3
- [11] Adam Van Etten. You only look twice: Rapid multi-scale object detection in satellite imagery. *CoRR*, abs/1805.09512, 2018. 3
- [12] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. *IJCV*, 2010. 1
- [13] R. Girshick. Fast r-cnn. In *ICCV*, 2015. 3
- [14] R. B. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *CVPR*, 2014. 3
- [15] J. Han, J. Ding, N. Xue, and G. Xia. Redet: A rotation-equivariant detector for aerial object detection. In *CVPR*, 2021. 1
- [16] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. 2016. 5
- [17] A. Lacoste, E. Sherwin, H. Kerner, H. Alemohammad, B. Lutjens, J. Irvin, D. Dao, A. Chang, M. Gunturkun, A. Drouin, P. Rodriguez, and D. Vazquez. Toward foundation models for earth monitoring: Proposal for a climate change benchmark. In *NeurIPS*, 2021. 1
- [18] C. Li, T. Yang, S. Zhu, C. Chen, and S. Guan. Density map guided object detection in aerial images. In *CVPRW*, June 2020. 2, 3, 4, 7, 8
- [19] J. Li, X. Liang, Y. Wei, T. Xu, J. Feng, and S. Yan. Perceptual generative adversarial networks for small object detection. In *CVPR*, 2017. 3
- [20] Y. Li, Y. Chen, N. Wang, and Z. Zhang. Scale-aware trident networks for object detection. In *ICCV*, 2019. 3
- [21] T.-Y. Lin, P. Dollár, R. B. Girshick, K. He, B. Hariharan, and S. J. Belongie. Feature pyramid networks for object detection. In *CVPR*, 2017. 1, 3, 5
- [22] T. Lin, P. Goyal, R. Girshick, K. He, and P. Dollar. Focal loss for dense object detection. In *ICCV*, 2017. 3
- [23] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollar, and C. L. Zitnick. Microsoft coco: Common objects in context. In *ECCV*, 2014. 1, 5, 6
- [24] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. E. Reed, C.-Y. Fu, and A. C. Berg. SSD: single shot multibox detector. In *ECCV*, 2016. 3
- [25] X. Long, K. Deng, G. Wang, Y. Zhang, Q. Dang, Y. Gao, H. Shen, J. Ren, S. Han, E. Ding, and S. Wen. PP-YOLO: an effective and efficient implementation of object detector. *CoRR*, abs/2007.12099, 2020. 3
- [26] Y. Long, Y. Gong, Z. Xiao, and Q. Liu. Accurate object localization in remote sensing images based on convolutional neural networks. *IEEE Transactions on Geoscience and Remote Sensing*, 2017. 1
- [27] J. Redmon, S. K. Divvala, R. B. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection. In *CVPR*, 2016. 3
- [28] J. Redmon and A. Farhadi. Yolo9000: Better, faster, stronger. In *CVPR*, 2017. 1, 3
- [29] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NeurIPS*, 2015. 1, 3, 5
- [30] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 2015. 5
- [31] B. Singh and L. S. Davis. An analysis of scale invariance in object detection snip. In *CVPR*, 2018. 3
- [32] B. Singh, M. Najibi, and L. S. Davis. Sniper: Efficient multi-scale training. In *NeurIPS*, 2018. 3
- [33] Jacob Solawetz. How to Detect Small Objects. <https://blog.roboflow.com/detect-small-objects/>, 2020. 2
- [34] Jacob Solawetz. Tiling for Small Object Detection. <https://blog.roboflow.com/edge-tiling-during-inference/>, 2022. 2
- [35] Z. Tian, C. Shen, H. Chen, and T. He. Fcos: fully convolutional one-stage object detection. In *ICCV*, 2019. 1, 3, 5, 8
- [36] Yuxin Wu, Alexander Kirillov, Francisco Massa, Wan-Yen Lo, and Ross Girshick. Detectron2. <https://github.com/facebookresearch/detectron2>, 2019. 5
- [37] G. 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 *CVPR*, 2018. 1, 5
- [38] C. Yang, Z. Huang, and N. Wang. Querydet: Cascaded sparse query for accelerating high-resolution small object detection. In *CVPR*, 2022. 2, 3, 5, 8

- [39] F. Yang, H. Fan, P. Chu, E. Blasch, and H. Ling. Clustered object detection in aerial images. In *ICCV*, 2019. 2, 3, 4, 5, 7, 8
- [40] X. Zhang, E. Izquierdo, and K. Chandramouli. Dense and small object detection in uav vision based on cascade network. In *ICCVW*, 2019. 8
- [41] P. Zhou, B. Ni, C. Geng, J. Hu, and Y. Xu. Scale-transferrable object detection. In *CVPR*, 2018. 3
- [42] X. Zhou, R. Girdhar, A. Joulin, P. Krähenbühl, and I. Misra. Detecting twenty-thousand classes using image-level supervision. In *ECCV*, 2022. 4
- [43] P. Zhu, L. Wen, D. Du, X. Bian, H. Ling, Q. Hu, Q. Nie, H. Cheng, C. Liu, and X. Liu. Visdrone-det2018: The vision meets drone object detection in image challenge results. In *ECCVW*, 2018. 1, 5