Object Detection in Aerial Images

A Case Study on Performance Improvement

Muhammed Uzair Khattak Adnan Khan Khaled Dawoud
Mohamed Bin Zayed University of Artificial Intelligence
{muhammad.uzair, adnan.khan, khaled.dawoud}@mbzuai.ac.ae

Abstract

Object Detection (OD) in aerial images has gained much attention due to its applications in search and rescue, town planning, and agriculture yield prediction etc. Recently introduced large-scale aerial images dataset, iSAID has enabled the researchers to advance the OD tasks on satellite images. Unfortunately, the available OD pipelines and ready-to-train architectures are well-tailored and configured to be used with tasks dealing with natural images. In this work, we study that directly using the available object detectors, specifically the vanilla Faster RCNN with FPN is sub-optimal for aerial OD. To help improve its performance, we tailor the Faster R-CNN architecture and propose several modifications including changes in architecture in different blocks of detector, training & transfer learning strategies, loss formulations, and other pre-post processing techniques. By adopting the proposed modifications on top of the vanilla Faster-RCNN, we push the performance of the model and achieve an absolute gain of 4.44 AP over the vanilla Faster R-CNN on the iSAID validation set.

1. Introduction

Object Detection (OD), in computer vision, is a task of classifying and localizing the objects within the images and videos. Given an image, OD aims to localize objects by assigning bounding box locations and then it classifies each localized object [1]. OD due to its widespread applications has gained much attention in recent years. A new and emerging task nowadays is the detection of objects in aerial images which has its unique challenges as compared to natural images [2]. The availability of large-scale satellite imagery datasets such as DOTA [3] and iSAID [4] has enabled researchers to make progress on the challenging task of aerial OD.

Before the realm of Deep Learning (DL), the task of OD used to be tackled using traditional machine learning techniques for example histogram of gradients (HoG) [5] which involved the designing of the hand-crafted features. After the leap of DL, not only the features extraction has become automated in the OD pipeline but also it has improved the results substantially leading to promising real-world applications. Therefore, DL is used as the de-facto method for developing OD-based applications.

In this work, we apply one of the most widely used OD architectures, Faster R-CNN [6] for detecting objects in a challenging satellite images dataset namely the iSAID dataset. We modify the baseline architecture to adapt it for providing better performance on satellite images. Our main study encloses the following:

1. We explore backbones from recently introduced vision models including ConvNext and SWIN transformers

2. We examine the use of different loss functions for OD

3. We investigate different transfer learning and pretraining techniques for better OD results

4. We investigate appropriate data augmentations for the iSAID dataset

5. We perform an experimental study on improving other various blocks of Faster-RCNN including RPN and RoI Pooling

2. Related Work

Object detection (OD), in aerial images, is a hotline of research due to its unique challenges and has therefore gained much attention in the recent past. OD detection in satellite images is different from natural images due to the presence of the humongous amount of small objects available in these images [7]. These objects can be of irregular shapes and have existing orientation variations (Fig. 1). Many state-of-the-art OD methods trained for natural images cannot easily adapt to these varying shapes, scales, and orientations in satellite images. In this section, we briefly mention the methodologies that are being used for the OD tasks in natural and aerial images.

On a higher level, the OD methods can be classified into two main categories: one-stage object detectors and two-stage object detectors. The two-stage object detection frameworks [8, 9, 6], first apply region proposal techniques to get regions of interest, and then extract the features of the regions and apply the predictions of categories and bounding boxes. Single-stage detectors [10, 11, 12, 13] require only a single pass to the neural network and then directly get the detection results. Both categories are widely used nowadays, with one stage being much faster in inference time and two-stage yielding higher results.
Recent studies have shown improvement in OD tasks with satellite images. An architecture based on a convolutional neural network (CNN) is presented in [14] for automatic image classification and detection in aerial images. Another recent work [15] proposes a novel two-stage detection, D2Det, to address both precise localization and accurate classification to obtain superior performance on object detection in satellite images. The proposed model with a ResNet101[16] and FPN backbone is evaluated on the UAVDT dataset [17]. In [18], an architecture based on Fast R-CNN and Faster R-CNN is used to detect vehicles in aerial images and validate their results on two publicly available datasets. In [19], a cluster proposal-based network to alleviate the problems of minor objects in aerial images. In our work, we conduct experiments in the similar fashion as followed in literature to improve OD performance on satellite images, and modernize a vanilla Faster R-CNN by proposing a number of modifications which leads to improved performance over the vanilla baseline model.

3. Methodology

3.1. iSAID Dataset

The aerial image datasets are slowly prevailing day by day and efforts are made to tailor the datasets so that real-life challenges with satellite images are addressed. For example, increasing the number of categories and the number of instances in these datasets is a crucial task to perform better on these images. Most popular of these datasets are: DIOR [20], DOTA [3], xVIEW [21] and iSAID [4]. The dataset chosen for our study is iSAID due to its unique challenges and higher number of categories and instances.

iSAID dataset is the first benchmark in the instance segmentation task of computer vision using aerial images. In the iSAID, there are a total of 2,806 high-resolution images with 6,55,451 object instances. There are a total of 15 categories namely ground track field, bridge, large vehicle, small vehicle, helicopter, ship, storage tank, baseball diamond, tennis court, basketball court, swimming pool, roundabout, soccer ball field, plane, and harbor. iSAID dataset images are of large resolutions and it is cumbersome for the DL models to handle. To address the issue of high resolution, a preprocessing step is used that is, patch sizes of 800x800 are extracted from the original images, now equalling 28,029 and 9,512 images for training and validation respectively. A few samples from the iSAID dataset can be seen in Fig. 1.

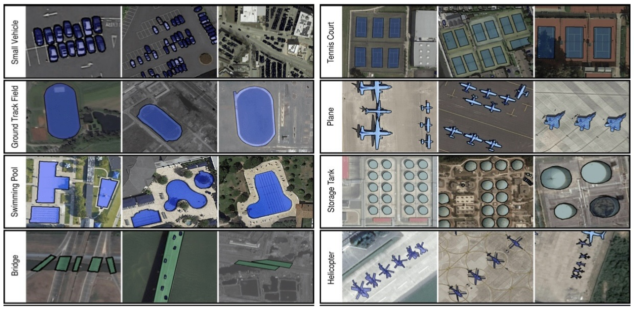

Figure 1: Samples of annotated images in iSAID dataset [4]

3.2. Baseline Model: FasterRCNN

We chose our baseline as the Feature Pyramid Network (FPN) based Faster R-CNN which is a two-stage OD architecture. Faster R-CNN as shown in Fig. 2 contains a backbone and a region proposal network (RPN) followed by a region-of-interest (ROI) pooling and prediction heads for classification and bounding box regression. The backbone consists of a convolutional neural network (CNN) and encodes a high-resolution input image into low dimensional semantically rich features. These features are then passed to RPN for generating class-agnostic object proposals for the image. Top N proposals (e.g. 1000, 2000, etc.) from RPN are selected and ROI pooling is applied to extract the corresponding convolutional features, which are then passed to output heads for object category classification and bounding box regression.

We choose a variant of Faster R-CNN that is FPN-based Faster R-CNN. FPN utilizes bottom-up and top-down paths along with lateral connections in a convolutional neural network (CNN) to construct semantically stronger features at multiple scales. To extract features at multiple levels, FPN-based Faster R-CNN uses the hierarchical property of convolution neural networks which acts as a feature extractor. This feature extractor acts as a backbone of the detector. In this case, ResNet-101 [16] is used as the backbone. The multi-scale feature maps helps the model to be trained on high-quality image data as compared to that of the single-level feature map used in the detector

The Fig. 3 shows our overall baseline. The backbone network is used to extract feature maps from the input image at different scales. The output feature maps generated are called P2 (1/4 scale), P3 (1/8), P4 (1/16), P5 (1/32), and P6

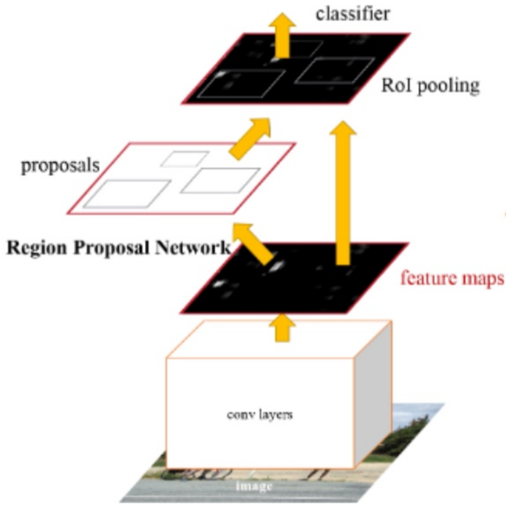

Figure 2: Faster R-CNN Pipeline [6]. Image is encoded using convolutional backbone and passed to RPN for object proposal generation. ROI pooling is used to extract region features which are passed to ROI head for classfication and bounding box regression.

(1/64). The non-FPN ('C4') architecture's output feature is only from the 1/16 scale. Following the backbone network is the Region Proposal Network (RPN) which detects object regions from multi-scale features. Specifically, anchor boxes strategy is used which proposes candidate bounding boxes on top of the FPN feature maps, with various sizes and scales. In the baseline model, anchor box sizes of 32, 64, 128, 256 and 512 are used. For each size, boxes are further replicated with three different aspect ratio scales including 0.5, 1, 2. From all of the anchor candidate samples, RPN network detects the positive and negative samples. A total of 1000 box proposals are obtained by default along with the confidence scores. Following the RPN, the Box Head generates the fixed-size features by cropping and warping the feature maps using the proposal boxes. Finetuned box locations and classification results are obtained using fully connected layers. Finally, 100 boxes (by default) in maximum are filtered out using non-maximum suppression (NMS). The box head is one of the sub-linear layer of ROI Head.

3.3. Modernization of Baseline Architecture and training strategy

With the chosen baseline FasterRCNN with R101 imageNet-1k [22] pretrained backbone, we apply several modifications in its different blocks to enhance the performance in the aerial image detection setting. Below we introduce each proposed modification one by one and in experiments section, we experiment with each modification separately and build on top of it for the next series of modifications if the proposed modification helps in improving the performance, otherwise the proposed changes are not incorporated in the subsequent set of experiments.

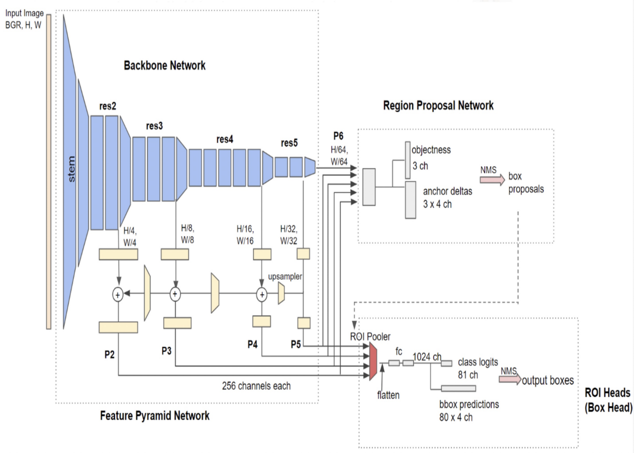

Figure 3: Detailed architecture of Faster-RCNN-FPN. Best viewed when zoomed in.

3.3.1 Backbone Changes

Having a suitable backbone for rich feature extraction in the object detection pipeline is crucial, as the classification and bounding boxes regression estimates are done on the feature maps produced from the backbone. Different choices of the backbone are available in the literature, here we explore backbones derived from competitive architectures, namely Deformable ResNet, SWIN transformer, and ConvNext.

Deformable-ResNet: Deformable convolution [23] has the ability to have a dynamic receptive field. As shown in the Fig. 4, during the convolution operation, sampling of points via network filters is not restricted in a fixed square window, rather it adaptively selects the feature points which are important in a neighborhood. Keeping in mind the nature of the iSAID dataset, which has objects of very high scale variation (Fig, 1), such convolution layers are important as they will adaptively change the receptive field to cover the object, whether it is small or large. However, a critical issue in using deformable layers is that it provides an overhead as firstly it has to estimate the sampling points and then apply the convolution process. To reduce the overhead, we apply deformable convolution in only 2 stages out of 5 stages of standard ResNet architecture.

Deformable-ResNet model constitutes deformable convolutional layers instead of normal convolutional layers in its bottleneck layers. Rest of the architecture is same as of the original ResNet model. Specifically, we build upon ResNet-101 architecture and design its deformable variant named called Deform ResNet-101.

SWIN Transformer: Vision transformer (ViT) model

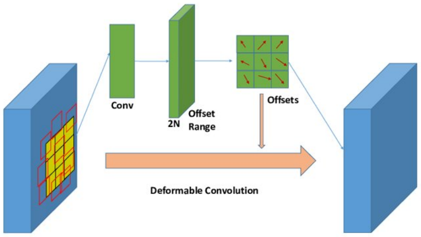

Figure 4: Block diagram of a single deformable convolution layer

based on shifted windows (SWIN) [24] have demonstrated competitive performance over other previously introduced vision transformers as well as CNN models. It uses SWIN blocks which contains shifted windows-attention layers. While restricting the self-attention to a window using shifted windows over image patches, it brings greater efficiency. For incorporating global interaction, it also allows the cross-attention between patches by using shifted windows scheme. Using patch merging technique, SWIN transformers proposes hierarchical architectures and are common choice of backbone.

ConvNext: Recently, another family of convolutional neural networks called ConvNext [25] have been introduced which performs at par with the recent popular transformer-based image models including SWIN. ConvNext is designed by building on top of the vanilla ResNet with modifications that are truly inspired by the vision transformer architectures. Major highlights of ConvNext include using long training schedules, strong data augmentations, and Macro and Micro designs. The macro design includes reshaping the architectural aspects of ResNet to that of SWIN transformers, such as changing the stage compute ratio, using multilayer perceptron (MLP) like block instead of bottleneck block, and using aggressive down-sampling block in the stem. The micro designs include the use of the GELU activation function instead of RELU, layer norms instead of the batch norm, etc. With the proposed model, ConvNext can exceed the performance of the SWIN transformer in a scalable manner on a variety of computer vision tasks including classification and detection.

In this work, we explore the performance of these different choices to be used as a backbone for the task of aerial object detection.

3.3.2 Loss functions

Loss functions puts the foundation of training on which the entire model is optimized. In the object detection pipeline, loss functions are formulated for both classification and regression. We use different loss functions in our experiments which are briefly described below.

Focal Loss for classification: Focal loss [26] is an extension over the normal cross-entropy loss which is mainly designed for solving the class imbalance problem. Generally, positive examples (foreground) are less than negative examples (background), the use of standard cross-entropy loss demands a high confidence score for positive examples to be classified. Due to the low number of positive examples, the network focuses more on understanding/differentiating the negative samples as compared to positive samples. Cumulative negative sample loss overwhelms the positive sample loss, which leads to degenerated model. Focal loss turns the model's attention to focus on hard examples by increasing the loss for hard examples (which are mainly miss-classified by the model) by assigning more weight to such cases. A comparison between focal loss and cross-entropy is given in Fig. 5.

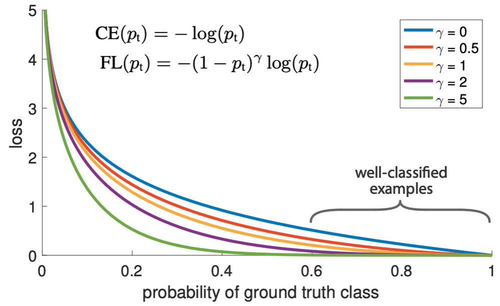

Figure 5: Focal loss adds a factor of $(1 - p_t)^{\gamma}$ to the standard cross-entropy loss. By setting $\gamma > 0$, the relative loss for well classified examples are reduced.

Federated Loss for classification: Federated loss [27], is a recently introduced loss formulation for class imbalanced dataset. As for an imbalanced dataset, some categories have very less annotations while some are densely annotated, so for a category having less annotation, the normal cross-entropy loss will pay more attention to the densely annotated classes (which the network is seeing more) and will not model a balanced class learning attributes. To mitigate this issue, federated loss uses only a subset of the total number of categories based on the square root frequency of each class in the training set. During the loss calculation, all positive classes and the subset from the negative classes are used. For our experiments on the iSAID dataset, we set the negative class sample count (subset length) equal to 10.

GIoU Loss for Regression: For letting the model learn to regress the bounding boxes accurately, regression losses such as l1 and l2 are used, and they are evaluated on the bases of the Intersection over Union (IoU) metric. However, the IoU formulation does not differentiate well between the very bad box prediction and a prediction that is relatively less bad. Thus there is no strong co-relation between minimizing these losses and improving the IoU metric. Generalized IoU [28] loss is formulated to take into account such cases where the IoU metric fails. Mathematically,

$$GIoU = \frac{|A \cap B|}{|A \cup B|} - \frac{|C \backslash (A \cup B)|}{|C|} = IoU - \frac{|C \backslash (A \cup B)|}{|C|} \tag{1}$$

Where A and B are the prediction and ground truth bounding boxes. C is the smallest enclosed area, which covers both A and B.

3.4. RoI Align Pooler Resolution

To provide the detailed level instance features to the RoI head, the RoI align pooler resolution is increased from 14x14 to 35x35, meaning that every shortlisted proposal is represented by a larger resolution of 35x35 which is input to the final classification and regression head. With this change, the final classification and regression layers gets region embedding with more detailed representations.

3.5. Modifications in Region Proposal Network (RPN) Architecture

Experiments are also conducted to validate if extending the default RPN layers helps in improving the performance or not. Specifically, ablation studies are carried out by introducing different architectural blocks in RPN before the 1x1 Conv layers for predicting the objectness and localization scores. A simple bottleneck block with 2 Conv layers is used which reduces the total channels by a factor of 2 and then again increases the channels by a factor of 2. For the squeeze and excitation block [29], we append the squeeze and excitation branch on top of the residual connection. The squeeze and excitation branch is shown in the Fig. 6.

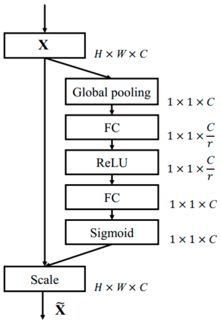

Figure 6: Squeeze and excitation block used in RPN

3.6. Pretraining techniques

We explore different pretraining techniques when transferring a model for the iSAID object detection task. The baseline model uses an imageNet-1k pretrained ResNet backbone. We study that, the use of a pretrained model already tailored for an object detection task is more suitable to be used in the downstream task. To validate this, we study the effect of using a Faster-RCNN model which is firstly trained on the COCO detection dataset. Additionally, we explore how a model trained on a segmentation task performs on our iSAID dataset. As segmentation is more dense prediction task, its trained model will be more aggressively attend to even small highlights in the images. For checking the performance of using the pretrained segmentation model, we use the Mask-RCNN model trained on the COCO instance segmentation dataset. When transferring this model for the detection task, its segmentation mask head is simply removed and the rest of the model is used as a detector.

3.7. Data Augmentations

To address the problem of high inter-class and intra-class variations, we make use of Data Augmentation at different scales. The baseline considers a single scale of 800 only (shorter side). We use scale augmentations at six different scales (1200,1000,800,600,400). Using scaling at such extreme scales takes into the account of category instances of varying scales.

3.8. Pre-Post processing techniques

Anchor Box Sizes: The region proposal network uses anchor box techniques to find possible class agnostic object proposals. The default baseline configuration for sizes of anchor boxes for each feature map is tailored for object detection for natural images. As shown in the plot in Fig. 7, the iSAID dataset contains high size variations having extremely large objects and extremely small objects. To have anchor box sizes of a similar proportion of the dataset, two new anchor box size configurations are added on top of the default one. Additionally, anchor boxes of 16x16 and 384x384 resolution are also provided for all feature maps at 3 different scales (0.5, 1.0, 1.5).

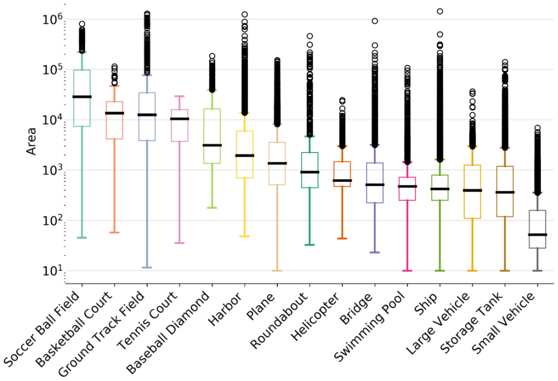

Figure 7: Box plot showing the high size variations of different objects present in the iSAID dataset. Plot adapted from [4]

Non-maximum suppression thresholds at training

and testing: The instance count per image for the iSAID dataset is very high i-e up to 200 annotations can be present in a single image. This count number is very high as compared to the instance count per image in natural images such as of COCO. Thus, we propose to increase the top proposals from RPN during training and testing time. Specifically, we increase the training and testing time RPN proposals from 2000 and 1000 to both 3000. This helps to increase the recall rate of the detector.

4. Experiments and Results

Evaluation details: For our experiments, we train all models on the iSAID official training dataset and report validation results on the official iSAID validation dataset. For every evaluation, we report the standard MS-COCO evaluation metrics to compare the performance of models. Specifically, we are interested in the AP (averaged across IoU thresholds) and AP_{50}.

Training details: For all experiments, we use detectron2 framework [30]. For backbone experiments, we use learning rates of 0.0025 and 0.00025. We observe that the ConvNext and SWIN backbone are very sensitive to learning rate and thus to stabilize training, we use a lower learning rate of 0.0025. For the rest of the experiments, we fix the learning rate to 0.0025. All models are trained for 100k iterations with a batch size of 2. Experiments with Deformable-ResNet backbone takes roughly 8 hours on a single Quadro RTX 6000 24GB GPU. For standard ResNet backbone, a single training time is 7 hours approximately.

Choosing the best backbone: For the initial set of experiments, we use a learning rate of 0.00025. The quantitative results are shown in Table 1. The SWIN-Base and ConvNext-Base models provide degraded performance as compared to the baseline ResNet-101 model. SWIN backbone provides 29.11% AP and 53.11% AP50 while ConvNext provides 28.35% AP and 51.45% AP50. Baseline ResNet-101 backbone provides 32.47% and 55.47% AP and AP50 respectively. The Deformable-ResNet backbone provides the highest AP and AP50 of 32.60% and 56.35% respectively. Further, the baseline model and Deformable-ResNet model are trained with a scaled learning rate of 0.0025 in Table 1. Scaled learning rate results for SWIN-Base and ConvNext-Backbone are not available because of unstable training in this setting. The Deformable ResNet backbone exceeds the performance of the baseline model by 1.36% AP and 1.65% AP50. For the experiments in the subsequent sections, we fix the backbone with the ResNet Deformable backbone.

Choosing better transfer-learning technique: With the chosen backbone as Deformable-ResNet-101, we ablate on using different pretrained models to be used on iSAID object detection tasks. Table 2 shows the performance results when different pretrained models are used. All experiments use the Deformable-ResNet backbone. It can be observed from the results that using completely pretrained models perform better than using the model with only the backbone pretrained. Specifically, when Faster-RCNN with COCO-detection pretrained model is used, it provides 41.86% AP and 63.28% AP50. Interestingly, a Mask-RCNN pretrained model on the COCO segmentation model provides further improved performance of 42.10% AP and 63.34% AP50. For our next series of experiments, we use both COCO-Segmentation pretrained model and the Deformable-ResNet backbone.

Backbone	LR	AP	AP50	AP75	APs	APm	APl
ResNet 101(baseline)	0.00025	32.47	55.47	33.98	19.93	39.8	40.13
SWIN Base	0.00025	29.11	53.11	28.44	19.43	36.32	33.63
ConvNext Base	0.00025	28.35	51.45	28.23	17.57	35.73	30.87
Deformable Resnet101	0.00025	32.6	56.35	33.19	20.24	39.53	38.1
Using higher learning rate							
ResNet 101(baseline)	0.0025	38.68	61.38	41.73	24.54	46.21	51.14
Deformable Resnet 101	0.0025	**40.04**	**63.02**	**43.02**	**25.87**	**47.36**	**51.6**

Table 1: shows the effect using different backbones. Deformable ResNet101 backbone provides improvement over the baseline model as compared to others. Using higher LR for SWIN and ConvNext results in unstable training, thus they are not included.

Model	Pretraining strategy	AP	AP50	AP75	APs	APm	APl
Deformable Resnet101	ImageNet 1k	40.04	63.02	43.02	25.88	47.37	51.60
Deformable Resnet101	MS COCO Detection	41.86	63.29	**46.09**	26.11	48.29	57.76
Deformable Resnet101	MS COCO Segmentation	**42.11**	**63.35**	45.98	**26.36**	**49.46**	**57.81**

Table 2: Effect of choosing the transfer-learning technique. Generally, the pre-training provided better results. Pretraining using MS COCO Segmentation dataset results in better performance compared to the MS COCO detection.

Performance evaluation of different loss functions: After selecting the suitable pretrained model and backbone, we study the use of different loss functions on top of the previous best model. The results are shown in Table 3. When the GIoU loss is used, it provides better performance as compared to the baseline model by providing 42.70% AP and 63.64%. Using federated loss does not help much in achieving gain on top of baseline instead slightly reduces the accuracies. Specifically, it provides 42.03% AP and 63.64% AP50. The use of focal loss also shows degraded results providing 17.62% AP and 25.50% AP50 respectively. The combination of losses also resulted in less performance as compared to the baseline results. Use of focal and federated loss has been studied for datasets having considerable number of classes, which is likely to be the reason for not showing improved results for iSAID dataset where there are only 15 categories. From these experiments,

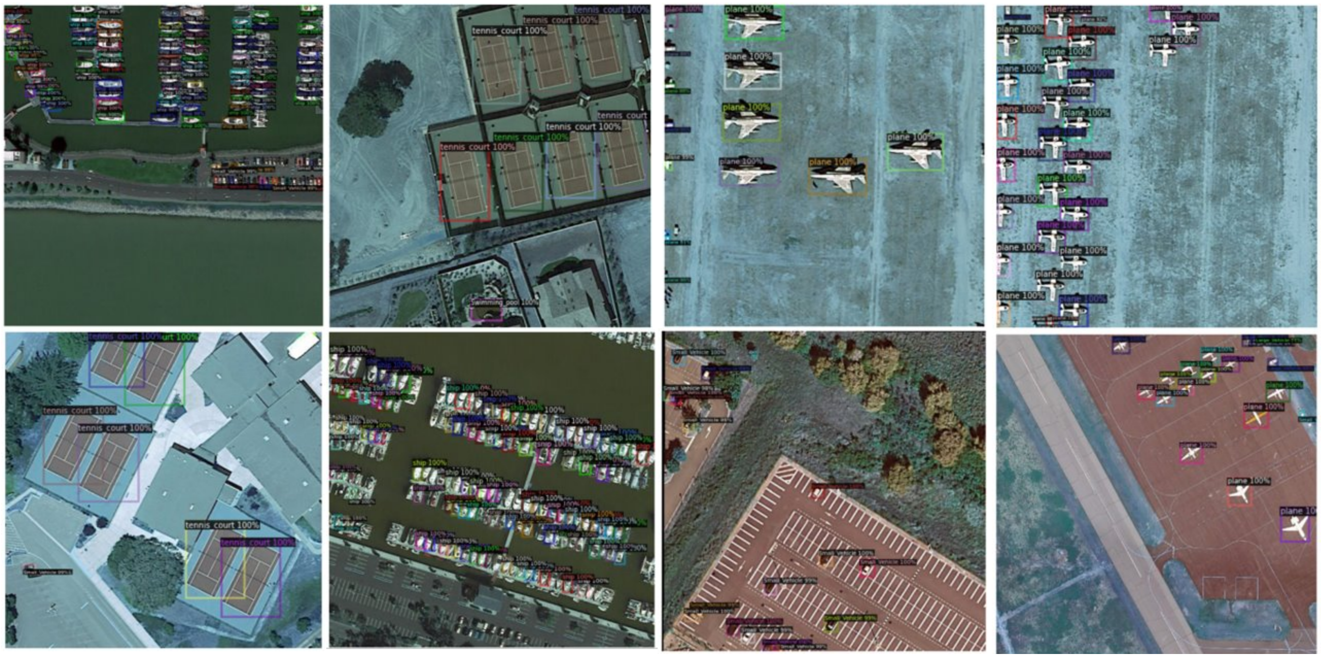

Figure 8: shows qualitative results of detection on the ISAID validation set. The model is able to accurately detect objects with varying scales.

we add the GIoU loss in our best model configurations, and report subsequent experiments on the best model configuration achieved so far.

Loss functions	AP	AP50	AP75	APs	APm	APl
Baseline*	42.11	63.35	45.98	**26.36**	49.46	57.81
GIOU	**42.70**	**63.95**	**47.08**	26.28	**50.60**	**58.53**
Only Federated	42.03	63.65	46.18	26.12	49.38	56.54
Federated + GIOU	41.86	63.29	46.09	26.11	48.29	57.76
Focal Loss	17.62	25.50	19.56	11.20	20.65	18.55
Focal + GIOU	13.78	19.41	15.16	8.53	16.53	15.79

Table 3: Shows the effect using different losses. GIOU loss achieved higher accuracy over he baseline in each metric except APs. * Here, the baseline is ResNet101 deformable with MS COCO Segmentation pretraining.

Results of architectural changes in RPN: Table 4 shows the effect of using different blocks instead of the default 3x3 single convolution layer in RPN. From the results, both variants having squeeze and excitation block as well as the bottleneck residual block shows less performance than the baseline. The Squeeze and Excitation block provides 41.91% AP and 63.23% AP50 while the residual block provides 37.101% AP and 54.93% AP50 accuracies. This shows that extending the RPN network by stacking new blocks and layers generally does not help. Based on these experiments, we do not include these modifications as they do not provide clear improvements over the current best model.

Introduced block in RPN	AP	AP50	AP75	APs	APm	APl
Baseline*	42.70	63.95	47.08	26.28	50.60	58.53
Baseline + Residual block	37.10	53.93	41.55	20.12	46.30	56.97
Baseline + Squeeze and Excitation Net	41.91	63.23	45.79	25.65	50.26	57.74

Table 4: Effect of using residual block and squeeze and excitation network in RPN. No clear results were observed by extending the RPN network. * Baseline here is ReseNet101 deformable backbone with MS COCO segmentation pretraining and GIOU loss.

Pre-post processing and RoI pooling: The results using different pre-post processing techniques and increasing RoI pooling are summarized in table 5. The addition of new anchor box sizes decreases 0.45% AP and increases AP50 by 0.42% AP. Increasing pooler resolution only helps in increasing the APs. By combining all proposed techniques, the model provides 42.735 AP and 65.16% AP50 indicating improvements over its default version. We have also conducted detailed ablation experiments on the effect of using different pooler resolution and anchor box sizes. These results are shown in the Table 8 and 9 for pooler resolution and anchor box sizes respectively. Those experiments use the original default baseline (vanilla Faster-RCNN with FPN (R-101) and imageNet-1k pretraining) with no other changes. We have observed that combining the best modifications in some cases does not provide better performance. Additionally, increasing the NMS from its default values of

2000 to 3000 resulted in higher performance. For the subsequent experiments, we use this improved model with all the above-mentioned modifications.

Proposed modification	AP	AP50	AP75	APs	APm	APl
Baseline*	42.70	63.95	**47.08**	26.28	**50.60**	**58.53**
+ additional anchor box sizes	42.24	64.37	45.27	26.98	48.78	57.38
+increased pooling resolution (35x35)	41.99	63.87	45.82	**26.77**	48.21	57.13
+ NMS techniques	**42.73**	**65.161**	46.146	27.69	49.644	57.05

Table 5: Effect of introducing additional anchor boxes sizes, increased pooling resolution and NMS. * Baseline here is ReseNet101 deformable backbone with MS COCO Segmentation pretraining and GIOU loss.

Adding data-augmentation: As our final modification, resizing training images with extreme scales including 1200 and 400 scales helps in improving the model. On top of our modernized model (configuration in the last section), the results after adding the scale augmentation are shown in Table 6. Specifically, the model achieves 43.12% AP and 65.84% AP50. It also improves the other metrics' performance except AP small (APs).

Proposed modification	AP	AP50	AP75	APs	APm	APl
Baseline*	42.73	65.16	46.14	**27.69**	49.64	57.05
+ Resizing at different scales	**43.12**	**65.84**	**46.98**	27.20	**49.74**	**58.15**

Table 6: Effect of introducing data Augmentation at different scales. * Baseline here is ReseNet101 deformable backbone with MS COCO segmentation pretraining, GIOU loss and pre-post processing techniques.

Modernized Faster-RCNN for Aerial OD

Here we summarize the final proposed model with all the positive modifications which helped in improving the performance of the model. Compared to the default vanilla Faster-RCNN with R-101 FPN backbone and imageNet-1k pretraining, our model uses Deformable-R-101 backbone with the COCO-Instance segmentation pretraining. The training objective uses GIoU loss. In the RPN, we increase the anchor box sizes proportion, and, increased pooler resolution (35x35) is used in the RoI Pooler block. The NMS topk proposals at training and test time are increased to 3000 and data augmentations with resizing at different scales are also incorporated. Results for these modifications on top of the default baseline are summarised in Table 7. The qualitative results for the final model are shown in Fig. 8.

Unuseful Modifications

Given the considerable improvement of the final proposed model over the baseline, there are number of modifications that were expected to yield better performance but provided sub-optimal results. Use of SWIN and ConvNext backbone could not push the performance much. Full potential of SWIN and ConvNext backbone can be utilized by using scaled hyper parameters and multi-GPU training, such as batch size of 16 and high learning rate. It could not be covered here due to the limited resources. Additionally, for the losses, the focal and federated loss provided poor results. For the focal loss, this might be because it is better at handling class imbalance datasets and requires more number of classes, as compared to only 15 classes present in the iSAID dataset. For the federated loss it requires optimal weight assignment. Experiments of extending RPN layers did not help and resulted in redundant model parameters, this shows that stacking new blocks and layers generally just to increase the model complexity does not always help. Overall studying the effect of each change was useful when modernizing a vanilla baseline architecture.

Proposed modifications	AP	AP50	AP75	APs	APm	APl
Basline (Faster-RCNN R-101 with ImageNet-1k)	38.68	61.38	41.73	24.54	46.21	51.14
+ Deform-R-101 backbone	40.05	63.02	43.02	25.88	47.37	51.60
+MS COCO Segmentation pretraining	42.11	63.35	45.98	26.36	49.46	57.80
+GIOU Loss	42.70	63.95	**47.08**	26.28	**50.60**	**58.53**
+ anchor + pooler + NMS modifications	42.73	65.16	46.15	27.69	49.64	57.05
+ Aug (resizing at different scales)	**43.12**	**65.84**	46.98	**27.20**	49.74	58.15
Evaluation of final model on iSAID test dataset						
Final proposed model	42.10	61.70	47.70	43.50	52.40	25.90

Table 7: Summary of the effect of each positive proposed modification. Finally the results of proposed model on iSAID test set are provided.

5. Conclusion

In this work, we studied the use of a two-stage object detector, Faster-RCNN for object detection on the iSAID dataset. Particularly, we observed that the vanilla Faster-RCNN is sub-optimal to be directly used for detecting objects in satellite images. We found that one of the major causes for that is the domain shift of the dataset. Vanilla Faster-RCNN is designed to be used for natural images which are very different from satellite images. To tailor the detector model to perform better for satellite images, we proposed several modifications over various blocks of the model and conducted a detailed set of experiments. Specifically, performance gains over the baseline model were achieved by proposing changes in the backbone, pretraining, loss function, RoI Pooler, and different pre & post processing techniques. Additionally, it is observed that extending the model layers in a detector such as in RPN does not always help to increase the performance. Overall, the proposed model provides significant improvements over the vanilla baseline model and verifies the importance of each proposed component introduced.

References

[1] Syed Sahil Abbas Zaidi, Mohammad Samar Ansari, Asra Aslam, Nadia Kanwal, Mamoona Asghar, and Brian Lee. A survey of modern deep learning based object detection models. *Digital Signal Processing*, page 103514, 2022. 1

[2] Jiaming Han, Jian Ding, Jie Li, and Gui-Song Xia. Align deep features for oriented object detection. *IEEE Transactions on Geoscience and Remote Sensing*, 60:1–11, 2021. 1

[3] Gui-Song Xia, Xiang Bai, Jian Ding, Zhen Zhu, Serge Belongie, Jiebo Luo, Mihai Datcu, Marcello Pelillo, and Liangpei Zhang. Dota: A large-scale dataset for object detection in aerial images. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3974–3983, 2018. 1, 2

[4] Syed Waqas Zamir, Aditya Arora, Akshita Gupta, Salman Khan, Guolei Sun, Fahad Shahbaz Khan, Fan Zhu, Ling Shao, Gui-Song Xia, and Xiang Bai. isaid: A large-scale dataset for instance segmentation in aerial images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 28–37, 2019. 1, 2, 5

[5] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)*, volume 1, pages 886–893. Ieee, 2005. 1

[6] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28, 2015. 1, 2, 3

[7] Seyed Majid Azimi, Eleonora Vig, Reza Bahmanyar, Marco Körner, and Peter Reinartz. Towards multi-class object detection in unconstrained remote sensing imagery. In *Asian Conference on Computer Vision*, pages 150–165. Springer, 2018. 1

[8] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587, 2014. 2

[9] 2015. 2

[10] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016. 2

[11] Joseph Redmon and Ali Farhadi. Yolo9000: better, faster, stronger. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7263–7271, 2017. 2

[12] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*, 2018. 2

[13] Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao. Yolov4: Optimal speed and accuracy of object detection. *arXiv preprint arXiv:2004.10934*, 2020. 2

[14] Igor Ševo and Aleksej Avramović. Convolutional neural network based automatic object detection on aerial images. *IEEE geoscience and remote sensing letters*, 13(5):740–744, 2016. 2

[15] Jiale Cao, Hisham Cholakkal, Rao Muhammad Anwer, Fahad Shahbaz Khan, Yanwei Pang, and Ling Shao. D2det: Towards high quality object detection and instance segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11485–11494, 2020. 2

[16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 2

[17] Dawei Du, Yuankai Qi, Hongyang Yu, Yifan Yang, Kaiwen Duan, Guorong Li, Weigang Zhang, Qingming Huang, and Qi Tian. The unmanned aerial vehicle benchmark: Object detection and tracking. In *Proceedings of the European conference on computer vision (ECCV)*, pages 370–386, 2018. 2

[18] Lars Wilko Sommer, Tobias Schuchert, and Jürgen Beyerer. Fast deep vehicle detection in aerial images. In *2017 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 311–319. IEEE, 2017. 2

[19] Fan Yang, Heng Fan, Peng Chu, Erik Blasch, and Haibin Ling. Clustered object detection in aerial images. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8311–8320, 2019. 2

[20] Ke Li, Gang Wan, Gong Cheng, Liqiu Meng, and Junwei Han. Object detection in optical remote sensing images: A survey and a new benchmark. *ISPRS Journal of Photogrammetry and Remote Sensing*, 159:296–307, 2020. 2

[21] Darius Lam, Richard Kuzma, Kevin McGee, Samuel Dooley, Michael Laielli, Matthew Klaric, Yaroslav Bulatov, and Brendan McCord. xview: Objects in context in overhead imagery. *arXiv preprint arXiv:1802.07856*, 2018. 2

[22] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009. 3

[23] Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. Deformable convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, pages 764–773, 2017. 3

[24] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10012–10022, 2021. 4

[25] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. *arXiv preprint arXiv:2201.03545*, 2022. 4

[26] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017. 4

[27] Xingyi Zhou, Vladlen Koltun, and Philipp Krähenbühl. Probabilistic two-stage detection. *arXiv preprint arXiv:2103.07461*, 2021. 4

[28] Hamid Rezatofighi, Nathan Tsoi, JunYoung Gwak, Amir Sadeghian, Ian Reid, and Silvio Savarese. Generalized intersection over union: A metric and a loss for bounding box regression. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 658–666, 2019. 5

[29] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7132–7141, 2018. 5

[30] Yuxin Wu, Alexander Kirillov, Francisco Massa, Wan-Yen Lo, and Ross Girshick. Detectron2. https://github.com/facebookresearch/detectron2, 2019. 6

Appendix

The following experiments are done over Faster-RCNN with ResNet101-deformable backbone to show the effect of introducing additional anchor boxes sizes and increasing the pooler resolution.

Modification	AP	AP50	AP75	APs	APm	APl
Resnet101-Deformable (baseline)	40.05	63.02	43.02	**25.88**	47.37	51.60
Pooler Resolution 14	39.92	**63.18**	42.72	24.97	47.57	51.86
Pooler Resolution 24	**40.16**	62.86	**43.21**	24.75	**47.62**	52.37
Pooler Resolution 34	39.47	62.46	42.09	24.07	47.18	**52.63**

Table 8: Shows the effect over the ResNet101-Deformable baseline when only increasing the pooler resolution, it indicate that pooler resolution provided better performance solely in some of metrics but when used collectively it provided sub-optimal results

Introducing additional anchor boxes size

Modification	AP	AP50	AP75	APs	APm	APl
Basline (Resenet101-Deformable)	40.05	63.02	43.02	**25.88**	47.37	51.60
Add 16 X 16 Anchor to P2	40.42	63.56	**43.86**	25.42	**47.94**	52.71
Add 384 X 384 Anchor to P5	40.12	63.04	43.07	24.56	47.77	52.37
Add 16 and 384 to P2 and P5, respectively.	**40.51**	**64.00**	43.48	25.75	47.65	**52.88**

Table 9: represents the effect of introducing 16, 384 to P2 and P5, respectively.