Rethinking Classification of Oriented Object Detection in Aerial Images

Phuc Nguyen*, Thang Truong*, Nguyen D. Vo, Khang Nguyen**
University of Information Technology, Ho Chi Minh City, Vietnam
Vietnam National University, Ho Chi Minh City, Vietnam

Abstract—**With the help of the rapid development of technology, especially the prevalence of UAVs (unmanned aerial vehicles), object detection in aerial images gains much more attention in computer vision and deep learning. However, traditional methods use horizontal bounding boxes for object representation leading to inconsistency between objects and features. Therefore, many detectors are being built to tackle this problem, and normally they use the conventional approaches of training and testing to achieve the results. Our pipeline proposed to strengthen not only the classification but also localization via independent training processes using convex-hull transformation in data pre-processing phase. We experimented with the well-designed S2ANet, R3Det, ReDet, RoI Transformer and Oriented R-CNN on the well-established oriented object detection dataset DOTA. Then we adopt the best detectors with the well-known classification network EfficientNet to our proposed pipeline and achieve promising results on the oriented object detection DOTA dataset. Moreover, our pipeline can flexibly be adapted to various oriented object detection baselines improving the results in classification via independent extensive training cycles.**

Keywords*—*Aerial images; classification; convex-hull transformation; data processing; oriented object detection

I. Introduction

Object Detection in Aerial Images (ODAI) has always been important in our real life with tremendous real-world applications such as surveillance, disaster prediction, emergency rescue, and even urban management [1] [2] [3]. Nowadays, it is becoming more feasible thanks to the increasing growth of studies in deep learning and the fast-paced development of information and communication technology. However, objects collected from aerial images appear in a variety of representations. They are often distributed in arbitrary orientations leading to confusion for many latest deep learning models, which opens a new study aspect in computer vision. To tackle this problem, a lot of experiments conducted show that using oriented bounding box representation (OBB) instead of horizontal bounding box (HBB) representation will alleviate the mismatch features and increase object detection accuracy [4] [5] [6] [7].

ODAI is extremely crucial to this world so it requires high accuracy and fewest mismatched objects in the prediction task as possible [8]. Although detecting objects in aerial images is vital, there is a lack of data about it. Many well-designed methods follow the traditional pipeline without refining class labels for output predictions, which could lead the model to behave biased toward less significant objects. Our proposed pipeline ensures that classes are treated equally, and models learn as much as data features from not only inside but also outside of the dataset through an independent classification training process.

In this study, we propose and provide a deep analysis of an effective training and testing pipeline to surge the performance of oriented object detectors in aerial images. Our pipeline applies the convex-hull transformation on ground-truth oriented bounding boxes to extract proper instances for the training and testing processes. Furthermore, we can use extra data for the independent training process to ensure the model classify proper label instances. We conduct extensive experiments on multiple baselines and apply the pipeline on them, yielding promising results.

We summarize our contributions in this paper as:

- Proposing a novel training and testing pipeline to improve classification performance flexibly adapt to many latest models.
- Providing a wise way to prepare data for classification training and an effective ensemble method in testing models.
- Using convex-hull transformation technique to transform oriented bounding boxes to horizontal ones for the further training process.
- Give a deep analysis of why we are choosing this pipeline and what are common problems of nowadays object detection methods in aerial images.
- Carrying out extensive experiments with the latest oriented object detection methods and providing an in-depth evaluation of the best deep learning models to strengthen our proposal.

The rest of the paper is: Section II is Related works; Section III is Methodology; Section IV is Our approach; Section V is Experiment and finally the last one, Section VI is Conclusion and Future Work.

II. Related Works

A. Oriented Object Detection

For the past decade, there have been various well-established object detector methods designed for tackling the horizontal object detection task, many of which have made remarkable progress such as Fast R-CNN [9], Faster R-CNN [10],

**Corresponding author
*Equal Contributor

Dynamic R-CNN [11], Deformable DETR [12], SSD [13], YOLOF [14], YOLOX [15], etc. However, these general object detection methods cannot tightly locate the object leading to the inconsistency between the classification and localization processes. Therefore, the extended branch of study using an oriented bounding box to represent the object's ground truths receives extensive attention to meet the need for applying deep learning to real-world applications (Fig. 1).

Current oriented object detection methods heavily depend on the original horizontal object detection task. They adopt the mechanism from extracting deep features and generating proposals to refining the final bounding boxes results. For example, Ding et al. introduced RoI Transformer [16] to tackle the problem of misaligned between regions of feature and objects, they applied the spatial transformation to Regions of Interest and configured the model to learn these geometric parameters using oriented bounding boxes labels. Jiang et al. introduced Rotational Region CNN [17] for detecting arbitrary-oriented texts in natural scene images. The model adopted the Faster R-CNN baseline using the region proposal network (RPN) to generate HBBs of texts and those HBBs will integrate many pooled RoI features to produce the final regressed OBBs. Han et al. introduced Single-shot Alignment Network (S2ANet) [18], addressing the issues of inconsistency between classification and localization. S2ANet consists of two main components: Feature Alignment Module (FAM) and Oriented Detection Module (ODM). The FAM specifically uses the Alignment Convolution to generate well-qualified anchors on which the active rotating filters of the ODM apply to encode the orientation information. Eventually, the network produces orientation-sensitive and orientation-invariant features to mitigate the inconsistency between classification scores and localization accuracy.

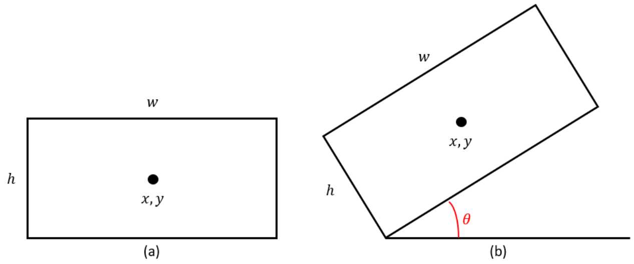

Fig. 1. Oriented Bounding Box (OBB) $[(x, y), w, h, \theta]$, where (x, y) is the Center and w, h, θ are the Width, Height and Angle of an OBB.

B. Classification

Classifying objects in aerial images using their CNN extracted features has always been a challenging problem since aerial images usually contain a tremendous number of various-shape instances. In 2020, Dosovitskiy et al. proposed a method Vision Transformer [19] following the baseline of the original architecture Transformer with the fewest modification possible. ViT splits images into many patches and connects these patches (NLP-vibes) using a sequence of linear embeddings. As the result, these patches are treated as tokens like in NLP and receive object queries for output labels. Tan and Le introduced EfficientNet [20] achieving better accuracy and efficiency than previous ConvNets by leveraging a multi-objective neural architecture search that optimizes both accuracy and FLOP. EfficientNet-B7 achieves state-of-the-art 84.3% top-1 accuracy on ImageNet while keeping inference time faster than prior ConvNets methods. Xie et al. introduced a multi-branch architecture called ResNext [21]. The deep network inherited by ResNets [22] consists of repeating building blocks aggregating a set of transformations with the same topology. By conducting extensive experiments, the authors came up with the conclusion that increasing cardinality is a more effective way of gaining accuracy than going deeper or wider, especially when depth and width start to give diminishing returns for existing models. The ResNext outperforms ResNet-101/152 [22], ResNet-200 [23], Inception-v3 [24] and Inception-ResNet-101/152 [25] on the ImageNet [26].

III. Methodology

The fundamental step to tackle Object Detection in Aerial Images is to collect a sufficient amount of data for training models, however in real life, especially in aerial images, objects are often distributed randomly, leading to hardship in data preparation steps. Therefore, humans unintentionally create many imbalanced datasets (Fig. 2) and passively bias the deep learning models.

Object detection in aerial images appears challenging when detectors have to deal with the variety in object scales and orientations, making them extremely difficult to identify. The most common way to approach these problems is image augmentation [27] (more image-more feature-high performance). In addition, the Elhagry and Saeed proposed many methods in [28] to solve this problem, such as modifying generated anchor sizes for region proposals and investigating multiple backbones and loss functions and achieved an improvement of 4.7 mAP over the baseline.

Consider image augmentation as a feasible solution for these problems. Some basic data augmentation methods are applied frequently, such as random crop, random rotate, random flip, zoom, etc. These augmentation techniques only apply to the whole image (Fig. 3), so what about class imbalance? It seems extreme to improve classification performance via improving the classification branch inside the models due to common batch sampling methods.

What if we train the classification independently from the object detection network and ensemble the results together? Although this approach might look heavy, ensure that not only you can modify each class independently (augmentation, removing noises) but also add extra necessary data for extended training cycles (Fig. 4).

IV. Our Approach

A. Pipeline

Our pipeline consists of two main parts: training and testing. In the training phase, Fig. 5, we train independently two networks (classification network and oriented object detection network). For the classification training data, we crop out oriented bounding boxes using convex-hull transformation and then carefully interpolate them to horizontal images. The data then proceeded to the classification network with a re-weighting mechanism. For the oriented object detection network, we train it with the original dataset to ensure regressed

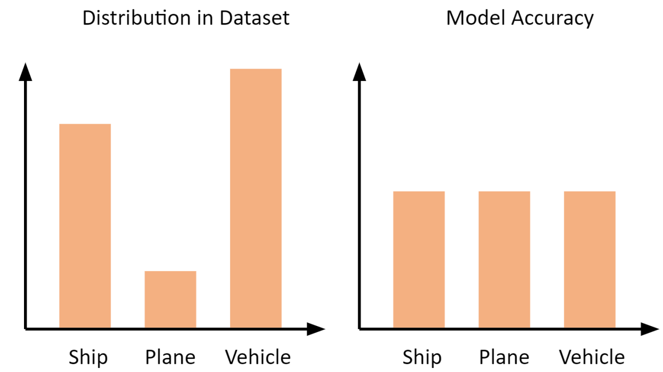

Fig. 2. Data Distribution Leads to the Problem of Class Imbalance in our Dataset. Our Target is to Implement the Model so that it Behaves Unbiased Toward Every Class.

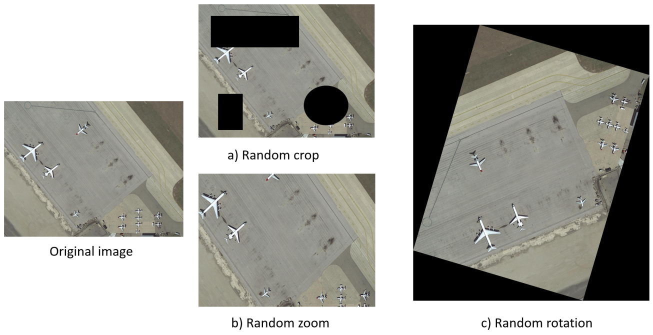

Fig. 3. Data Augmentation is a Common Method to Enrich the Amount of Data used for Training Object Detection Models. It only Enriches the Amount of Data, While we need the Instances Enrichment in Each Class.

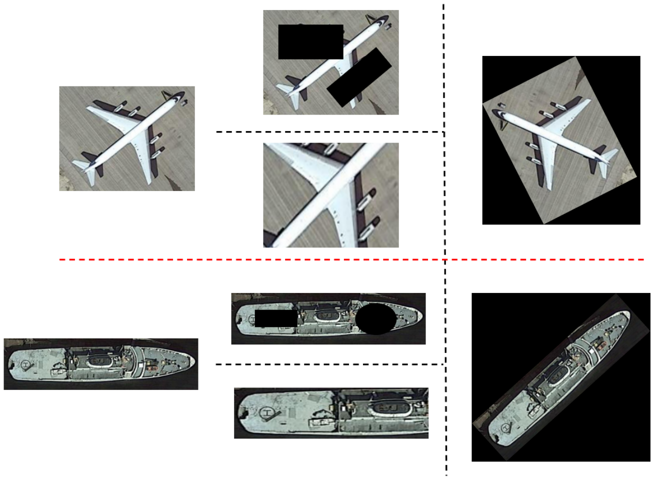

Fig. 4. Data Augmentation for Every Instance for each Class Ensures Richness in Features for the Classification Training Procedure.

bounding boxes are accurate. In the testing phase, Fig. 6, predicted oriented bounding boxes then be cropped out and interpolated to horizontal boxes. They will be re-labeled by the classification network (keeping high confidence score) and ensemble with the results of the oriented object detection network. Finally, those with low confidence scores will be removed as long with the NMS process from the prediction set.

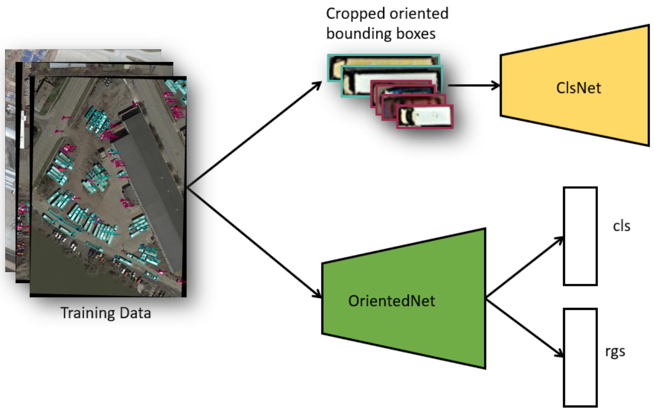

Fig. 5. Training Phase: Training Data Passed through an Oriented Object Detector's Network Contributes to Training Classification and Localization as in Other Well-Known Pipelines. However, the Required Data Preparation for Training the Classification Network is an Indispensable Step. Oriented Ground-Truths First are Interpolated to Cropped Oriented Bounding Boxes via Convex-Hull Transformation, then they're Transformed to Horizontal Images Fed to a Classification Network Together with not only the Re-Weighting Mechanism but also Image Augmentation Methods.

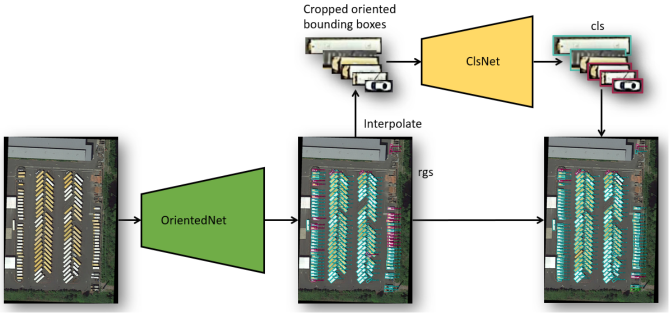

Fig. 6. Testing Phase: Different from the Training Phase, the Testing Image is Fed into Only the Oriented Object Detection Network. Then these Regressed Oriented Bounding Boxes Produced from the Network are Fed into the Classification Network after being Interpolated to Output the Predicted Label. Finally, we Ensemble the Outputs from Both Networks to Get the Appropriate Results in the Final Prediction set via Score Fine-Tuning and Non-Maximum Suppression.

B. *EfficientNet*

EfficientNet [20] adopting the idea of the CNN network can be scaled in three dimensions: depth, width, and resolution. The depth of the neural network corresponds to the number of layers in the network. The width refers to the number of neurons in each layer or the number of channels in each Conv layer (the number of channels of the output). Resolution is simply the height and width of the input image. The following Equation 1 describes the compound scaling method where d is

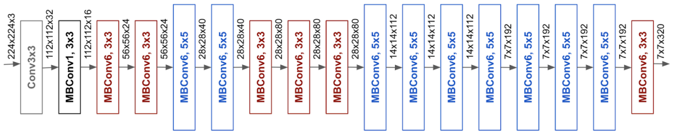

Fig. 7. EfficientNet-B0 [31] Structure.

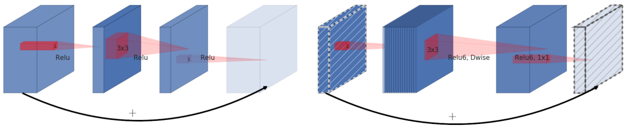

Fig. 8. Difference between Residual Block and Inverted Residual Block [30].

the depth, w is the width and r is the resolution. Moreover, ϕ is a user-defined coefficient determining the available resources for model scaling and α, β, γ specify how to assign these extra resources to network width, depth, and resolution.

$$\begin{gathered} d = \alpha^{\phi} \\ w = \beta^{\phi} \\ r = \gamma^{\phi} \\ \textit{Following the constraints:} \\ \alpha \cdot \beta^2 \cdot \gamma^2 \approx 2 \\ \alpha \geq 1, \beta \geq 1, \gamma \geq 1 \end{gathered} \tag{1}$$

The compound scaling method can generalize to existing CNN architectures such as MobileNet [29] and ResNet [22]. However, choosing the base network is important to get the best results because it only increases the predictive power of the neural network by reconstructing the parameters and structure of the base network. The author also uses Neural Architecture Search to build an efficient network architecture - EfficientNet B0. It achieves 77.3% accuracy on the ImageNet with 5.3M parameters and 0.39B FLOPS (ResNet-50 achieves 76% accuracy with 26M parameters and 4.1B FLOPS).

The main building block of EfficientNet-B0 is the MBConv block. The MBConv block, Fig. 8 is similar to the inverted residual block used in MobileNetv2 [30]. In this block, there is a shortcut connection between the beginning and the end of the block. The input is scaled with a 1x1 Conv layer to increase the number of channels or the depth of the feature map. Then they use Depthwise convolution 3x3 and Pointwise convolution (Conv layer 1x1) to reduce the number of channels of the output. A shortcut connection connects narrow layers (a small number of channels) while wider layers are in the middle of the shortcut connection (Fig. 7). This structure helps to reduce the number of parameters and the number of operations.

C. R3Det

ReDet [4] proposed to take a huge step from horizontal object detection to oriented object detection by solving feature misalignment during the feature extraction process. The model uses the Feature Refinement Module together with feature interpolation to extract position information related to the refined bounding box and reconstructs feature maps to achieve feature alignment. R3Det adopts Refined Rotation RetinaNet as the backbone with multiple stages of refinement like Cascade while speeding up the model by reducing the number of refined bounding boxes in the first stage.

D. S2ANet

S2ANet [18] proposed to solve the inconsistency between classification and localization performance. It adopts the Feature Pyramid Network to extract high-level features. Anchor generator, namely Feature Alignment Module, generates high-quality anchors which then pass through Oriented Detection Module for classification and regression.

E. RoI Transformer

RoI Transformer [16] tackles the problem of using horizontal bounding boxes (mismatch features). The model consists of the lightweight Rotated RoI learner with a 5 dimensions fully connected layer representing the offset of the rotated RoI corresponding to the HRoI. Rotated RoI warping generates fixed-size geometry robust features for classification and regression via feature maps and rotated RoIs.

F. ReDet

ReDet [32], namely the Rotation-equivariant Detector adopts ResNet [22] and Feature Pyramid Network so as to extract rotation-equivariant features. RiRoI Align proposed along with the model transforms rotated RoIs generated by an RPN and RoI Transformer [16]. The final feature extraction step regresses final oriented bounding boxes and classifies corresponding labels.

G. Oriented R-CNN

The well-designed two-stage detector Oriented R-CNN produces high-level features for oriented object detection. The main structure inherits from the two-stage object detector baseline while introducing a new representation of region proposals. The oriented RPN encodes features received from each level feature of the FPN [33] and decodes them into RoIs under the Midpoint Offset representation. Due to the Midpoint offset representation, the oriented proposal generated by oriented RPN is usually a parallelogram, so the model will slightly adjust the shorter diagonal to the same length as another diagonal obtaining oriented bounding boxes. Finally, each RoIs is divided into $m \times m$ grids to produce a fixed-size feature map F'. The idea that produces feature map F' adopts the idea of the rotation transformation the same as [5] and these fixed-size feature maps F' are then fed into fully-connected layers regressing the offset and assigning categories for each oriented bounding box.

H. Non Maximum Suppression

In the process of improving this result, after relabeling the result by EfficientNet, we use non-maximum suppression (NMS) to sift out the overlapping bounding box instances as follows (Fig. 9):

Input: a list of oriented bounding box A, corresponding confidence score S and overlap threshold N.

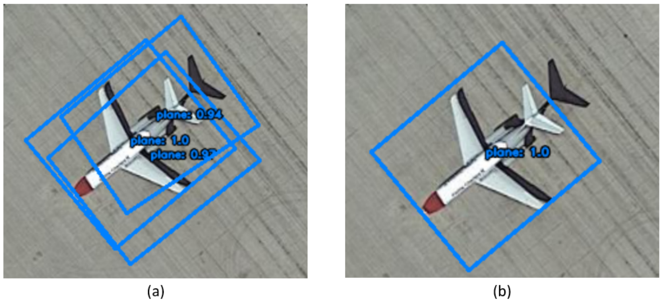

Fig. 9. Objectives of Non-Maximum Suppresion. (a) Before NMS. (b) After NMS.

Output: a list of sifted bounding box B.

Algorithm:

Step 1: Select the bounding box with the highest confidence score, remove it from A and add it to the final list B.

Step 2: Now compare this bounding box with all the bounding boxes in A (calculate the IoU). If the IoU is greater than the threshold N, remove that bounding box from A.

Step 3: This process is repeated until there is no more bounding box left in A.

We calculate IoU between two oriented bounding boxes as IoU between two rectangles (each is the smallest horizontal rectangle that includes the corresponding oriented bounding box) as Fig. 10.

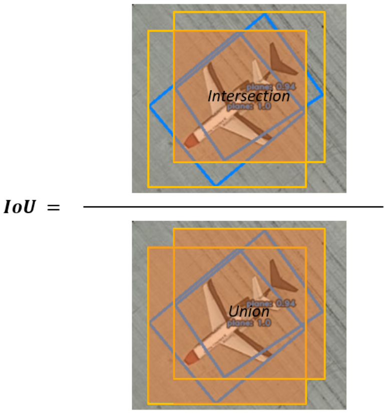

Fig. 10. Intersection over Union.

I. Results Post-Processing

After having regressed the oriented bounding box, we transform it to the horizontal one and feed it through the classification network to get the appropriate label. Also we ensemble the classification results with the oriented object detection network to make cleaner results. Finally, NMS will be activated to sift and get the right fitter bounding boxes. This approach will ensure that there will be no more overlapping bounding boxes for one object caused by class imbalance and inconsistency between regression and classification (common problems in aerial images dataset).

V. Experiment

A. Experiment

DOTA Oriented Dataset: In this study, we conducted experiments on the DOTA dataset [34], which is a large-scale dataset widely used for the oriented object detection problem in aerial images. The DOTA was introduced in 2018, containing 2,806 images, and the proportion of the training set, validation set, and testing set were 1/2, 1/6, and 1/3, respectively. The dataset contains 188,282 instances which are accurately labeled of 15 common object categories includes: plane (PL), baseball-diamond (BD), bridge (BR), ground-track-field (GTF), small-vehicle (SV), large-vehicle (LV), ship (SH), tennis court (TC), basketball court (BC), storage tank (ST), soccer-ball-field (SBF), roundabout (RA), harbor (HA), swimming pool (SP), and helicopter (HC).

Data Preparation for Classification Training: To train the EfficientNet model, we need an input dataset that is an object image set (bounding box of objects) on each DOTA class. This process is illustrated in Fig. 11.

Step 1: Determine the object bounding box to be extracted [Fig. 11(a), red quadrangle].

Step 2: The object bounding box shape we need to take out is a rectangle but DOTA's labeled bounding box is arbitrary quadrilateral, so we take the smallest oriented bounding box that covers DOTA's labeled bounding box (convex-hull transformation) [Fig. 11(b), green bounding box].

Step 3: Rotate the image so that the bounding box to be taken becomes a horizontal bounding box. To reduce the calculation cost, we define the smallest area of the image that includes the box to be extracted, then proceed to rotate this area [Fig. 11(c), 11(d)]

Step 4: Extract the object bounding box [Fig. 11(e)].

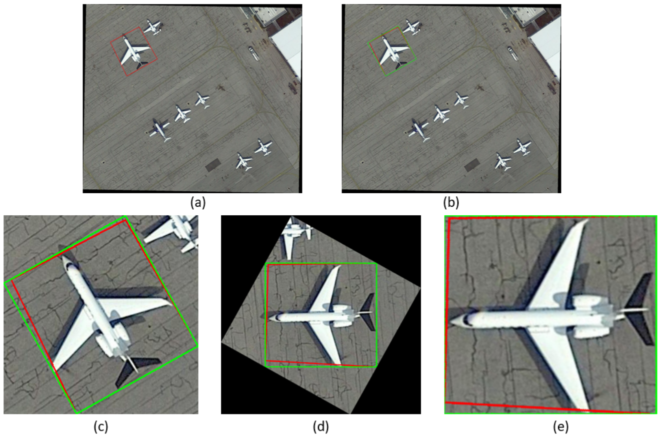

Fig. 11. Data Preparation Steps using Convex-Hull Transformation.

Apply the above steps to all instances on DOTA's train and validation set, then we get the train and validation set to train EfficientNet, the detail as shown in Fig. 12. DOTA is the imbalance dataset - a normal problem in real-life aerial images data where vehicles and ships outweigh most of them.

Experimental Configuration: We implement Oriented R-CNN, S2ANet [18], ReDet [32], R3Det [4], RoI Transformer

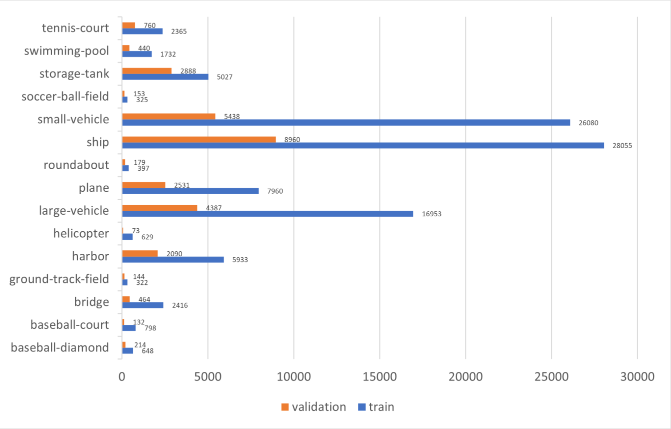

Fig. 12. Illustration of the Number of Ground-Truths in Training and Validation set of DOTA Dataset.

[16] on MMRotate [35] and EfficientNet-b3 on MMClassification [36] framework with repeat factor sampling methods [37] solving class imbalance problem using 2x GPU RTX 2080Ti

Evaluation Metric: To evaluate the performance of the models, and our pipeline on the DOTA dataset, we use the mAP score (mean Average Precision). Mean Average Precision is a popular evaluation metric used for object detection. It is the average of AP in every class in a dataset.

B. Analysis

In this study, we test five models, R3Det, RoI Transformer, ReDet, S2Anet, and the Oriented R-CNN along with post-processing methods on the DOTA dataset and achieve the results presented in Table I.

For the initial results, all models perform very well in directional object detection (mAP ranges from 70% to 77%). In which the highest resulting model is ReDet (mAP = 76.68%) and the lowest is R3Det (mAP = 69.8%). Across DOTA's 15 layers, the accuracy of each model on each class varies widely (accuracy ranges from 50% to 90%), of which the highest are tennis-court (about 91%) and plane (about 89%); Especially the lowest is the helicopter class, resulting in a sizable difference between models (the highest is on ReDet 66.71%, the lowest is on R3Det 37.44%). Besides, R3Det also achieved very low results on the bridge class (45.41%) compared to other models in this class.

However, in the results of the above models, there is the same limitation which is the discovery of many other bounding boxes of the different classes located on the same instance (Fig. 13).

The two most accessible post-processing solutions:

- The first is the result of sifting out boxes, the scores of which are less than a certain threshold. The results are pretty good [Fig. 14(b)], but there are still some overlapping bounding boxes because these wrong bounding boxes have a score greater than the sifting threshold [Fig. 15(a)]. And this besides removing bounding boxes also removes quite a lot of true bounding boxes [Fig. 15(b)];

- The second is that as a result of using multi-class non-maximum suppression, similar to the first solution, the result of removing bounding boxes is also quite good [Fig. 14(c)]. However, this also has a limitation, which is the removal of bounding boxes in case 2 objects overlap. Both solutions caused the model's mAP results to decrease (about 2%).

After applying our initial solution: relabel all bounding boxes using EfficientNet, then apply non-maximum suppression to sift out overlapping bounding boxes. The result is that there is only one bounding box per object, which is more effective than all 2 conventional solutions above [Fig. 14(d)]. However, the accuracy of the results is significantly reduced (down almost 10-40% from the original). Among them, the most affected classes are soccer-ball-field (about 20-40%), bridge (about 14-60%), and helicopter (about 17-30%), ground-track-field (about 12-50%). The subjective reason for the decrease in results on layers is due to the misidentification between classes of EfficientNet, between classes with similar characteristics that make the model easily confused (between small-vehicle and large-vehicle, ship; between roundabout, plane and helicopter) (Fig. 16). The objective reason is that the imbalance of big data between classes (Fig. 12, small-vehicle and ship over 20k bounding boxes, while the remaining classes such as soccer-ball-field, ground-track-field are only about over 300 bounding boxes) makes the quality of layering between classes uneven; And the lack of information about the surroundings, because EfficientNet is only trained on the image, are bounding boxes (the similarity between ship and small-vehicle when cut into bounding boxes, Fig. 17).

So, to improve the accuracy of our pipeline, we combine the results between the original model and the results of EfficientNet, in detail: we will not be too confident in EfficientNet, which means that there will be no relabeling if the results returned by EfficientNet belong to soccer-ball-field, helicopter, bridge, and ship (which are classes that the EfficientNet model classifies inefficiently and affects the other classes analyzed above); it only be relabeled if the original model gives an uncertain result, which means that there will be a threshold if the model gives results below this threshold, then relabel will be applied.

The final score result of our pipeline is close to the original result (on three models with highest original results, RoI Transformer, ReDet, Oriented R-CNN), higher than the usual two solutions, and also very high efficiency in sifting wrong bounding boxes (Fig. 18). The other two models (S2ANet, R3Det) have a lot of wrong boxes, having a relabel doesn't work well either.

Besides efficiency, our pipeline is still wrong in several instances [Fig. 19(a)], which is still limited in cases where wrong bounding boxes or bounding boxes are part of the instance of the true bounding box (e.g. on container truck objects surrounded by large-vehicle boxes, but another bounding box covers the front of the car as a small-vehicle, Fig. 19(b)

In summary, our pipeline has solved the problem of multiple different boxes on the same object while keeping accuracy. The solution still does not resolve the case where wrong bounding boxes or different classes bounding boxes are part of the object of the true bounding box.

TABLE I. EXPERIMENTAL RESULTS

Model	**Result Post-processing**	PL	BD	BR	GTF	SV	LV	SH	TC	BC	ST	SBF	RA	HA	SP	HC	mAP
R3Det	Original	**89.3**	**75.31**	45.41	69.24	**75.54**	**72.89**	**79.29**	**90.89**	**81.02**	**83.26**	**58.82**	**63.15**	**63.41**	**62.21**	**37.41**	**69.8**
	Sifting	89.3	73.79	45.41	68.93	74.56	72.89	79.29	90.89	78.55	78.64	58.82	62.1	63.41	57.84	36.83	68.75
	Multi-class Non-maximum Suppression	89.3	75.31	**45.42**	**69.31**	74.32	72.67	79.28	90.89	79.06	82.71	55.23	62.71	63.33	62.1	33.25	69
	Relabel by EfficientNet	68.96	15.88	8.33	16.36	24.8	47.07	32.76	78.16	21.53	59.61	15.48	22.04	13.77	37.79	1.27	30.92
	Relabel by EfficientNet with conditions	84.4	72.48	44.6	53.2	53.31	64	69.53	85.29	77.15	60.12	57.38	61.84	56.7	41.46	31.36	60.86
S2ANet	Original	89.3	80.49	**50.42**	**73.23**	**78.42**	**77.4**	86.8	90.89	85.66	84.24	**62.16**	65.93	**66.66**	**67.76**	**53.56**	**74.19**
	Sifting	89.25	76.02	48.73	71.2	74.84	76.05	86.8	90.89	**85.99**	84.2	61.19	64.31	66.59	67.8	52.96	73.12
	Multi-class Non-maximum Suppression	**89.31**	**80.88**	50.42	71.56	76.49	76.16	**86.81**	**90.9**	85.23	**84.26**	59.49	**66.32**	66.66	67.76	53.56	74.19
	Relabel by EfficientNet	53.23	15.26	10.51	11.90	22.81	37.85	26.15	39.31	13.65	37.84	9.46	16.16	10.71	28.48	0.41	22.25
	Relabel by EfficientNet with conditions	82.96	71.9	40.05	64.93	37.11	61.47	75.67	16.32	76.05	12.35	4.55	64.4	58.65	13.42	43.12	48.2
RoI Transformer	Original	**88.98**	82.17	54.59	76.28	**79.29**	**77.96**	**87.94**	**90.91**	**87.19**	**85.65**	**61.44**	**62.63**	**74.63**	**72.43**	59.23	**76.09**
	Sifting	88.98	82.17	54.59	73.64	74.77	77.96	87.94	90.91	87.19	85.65	54.55	62.63	68.87	72.43	56.83	72.43
	Multi-class Non-maximum Suppression	88.98	**82.23**	**54.6**	**76.43**	74.77	77.96	87.94	90.91	86.76	85.54	60.83	62.63	74.57	72.43	56.86	75.78
	Relabel by EfficientNet	67.55	16.15	12.98	15.5	24.83	44.75	32.97	70.33	19.33	60.52	14.08	21.76	13.78	39.02	1.49	30.33
	Relabel by EfficientNet with conditions	88.85	82.22	54.59	76.31	78.64	77.83	87.9	90.9	87.12	85.55	61.05	62.62	68.84	72.39	**59.27**	75.6
ReDet	Original	89.2	83.77	52.21	71.04	78.05	**82.5**	**88.24**	**90.86**	**87.26**	**85.98**	**65.58**	62.86	**75.86**	70.04	66.71	**76.68**
	Sifting	**89.76**	78.79	47.01	65.2	**80.98**	80	87.33	90.74	79.17	86.23	49.09	**65.87**	65.75	**71.86**	55.21	72.87
	Multi-class Non-maximum Suppression	89.2	83.8	52.21	71.1	73.88	78.01	88.24	90.86	87.26	85.97	65.58	60.42	75.83	70.04	64.29	75.78
	Relabel by EfficientNet	68.81	16.11	8.36	16.38	24.82	46.27	32.76	70.13	21.77	59.74	15.55	22	13.72	37.82	1.3	30.37
	Relabel by EfficientNet with conditions	89.17	**83.81**	**52.21**	**71.1**	73.75	81.72	88.21	90.83	87.23	85.93	65.39	60.41	75.35	70	**66.89**	76.13
Oriented R-CNN	Original	89.35	81.41	**52.6**	**75.02**	**79.03**	**82.41**	87.82	90.9	86.4	**85.3**	**63.36**	**65.7**	**68.28**	**70.48**	**57.23**	**75.69**
	Sifting	89.35	81.41	52.6	72.58	74.3	77.96	87.82	90.9	86.4	85.3	63.36	63.68	68.28	70.48	54.53	74.6
	Multi-class Non-maximum Suppression	89.36	**81.45**	52.59	72.65	74.27	77.96	87.82	90.9	**86.57**	85.29	60.73	63.68	68.28	70.48	54.78	74.45
	Relabel by EfficientNet	89.18	74.79	38.19	60.78	71.14	67.22	77.21	90.87	77.35	78.19	43.58	56.21	63.9	61.6	37.06	65.82
	Relabel by EfficientNet with conditions	**89.36**	79.77	52.56	74.37	78.6	82.15	**87.82**	**90.9**	86.14	85.24	62.03	63.48	68.27	70.45	54.33	75.03

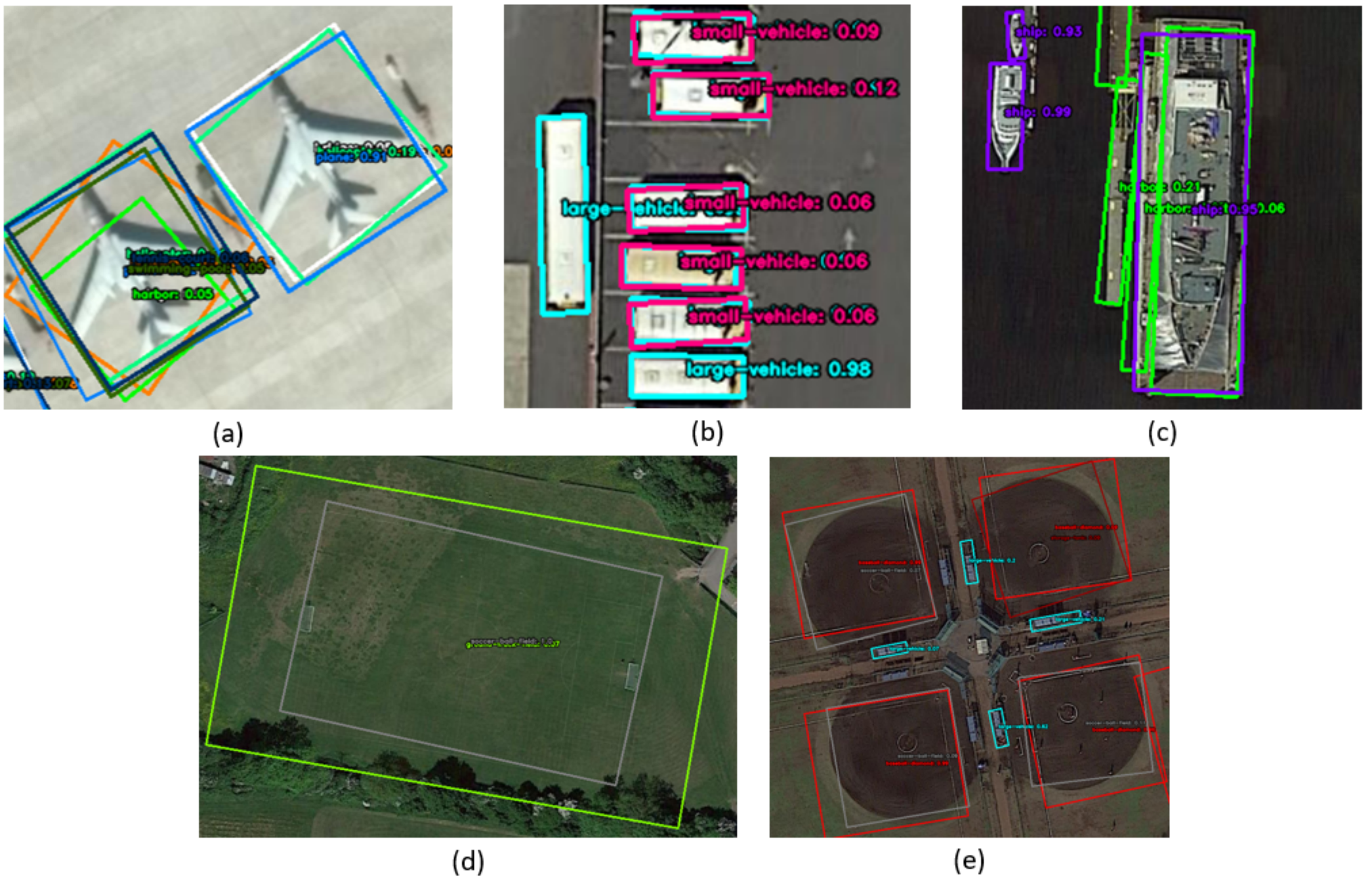

Fig. 13. Model's Result Limitation. (a) R3Det. (b) RoI Transformer. (c) ReDet. (d) S2ANet. (e) Oriented R-CNN

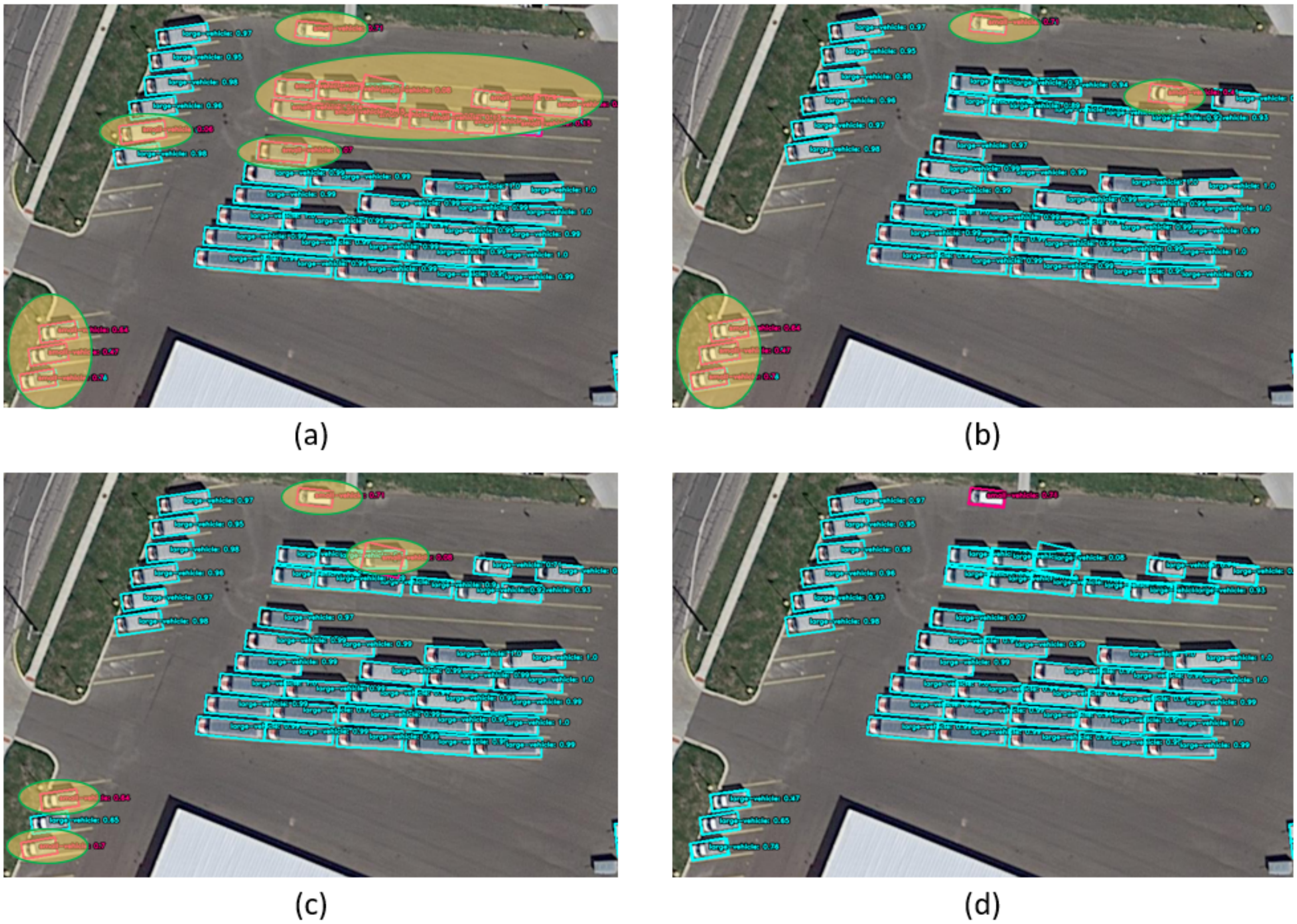

Fig. 14. Post-Processing Results. (a) Original. (b) Sifting. (c) Multi-Class Non-Maximum Suppression. (d) Relabel by EfficietNet. Covered Areas Represent Class Imbalance and Inconsistencies between Classification and Localization Leading to Bounding Boxes of Different Classes on One Object.

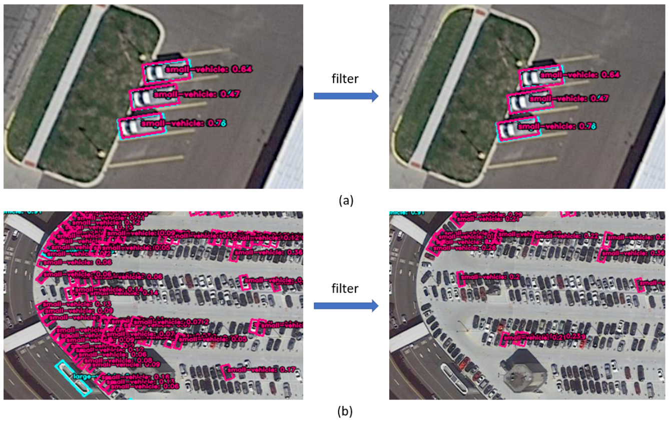

Fig. 15. Bad Case of Sifting Solution. (a) Wrong Bounding Boxes have a Score Greater than the Sifting Threshold. (b) True Bounding Boxes have a Score Smaller than the Sifting Threshold.

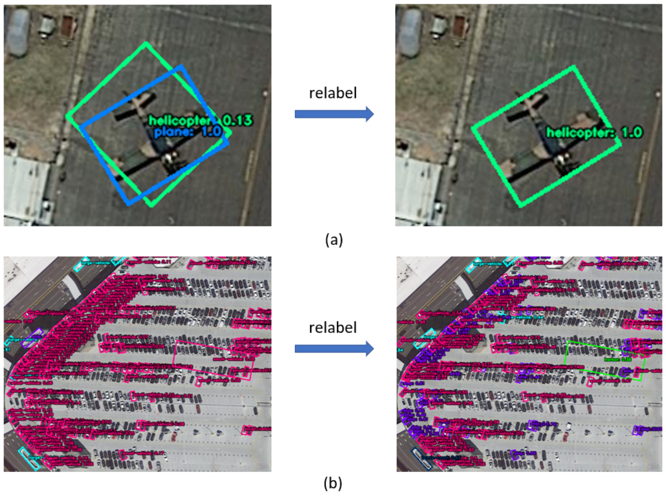

Fig. 16. Bad Case of Relabeling by EfficientNet. Misidentification between Classes of EfficientNet Resulting in the False Removal and Relabel Bouding Boxes

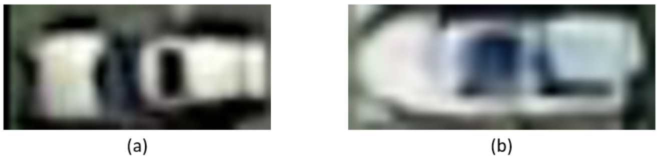

Fig. 17. Similarity between Small-Vehicle and Ship. (a) Small-Vehicle. (b) Ship.

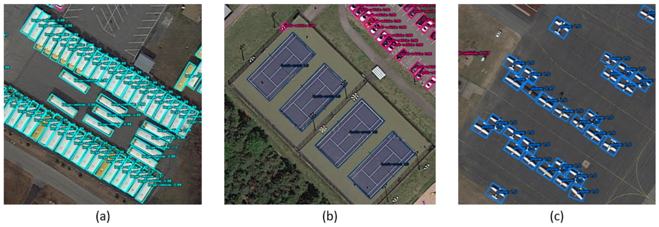

Fig. 18. Our Pipeline Result.

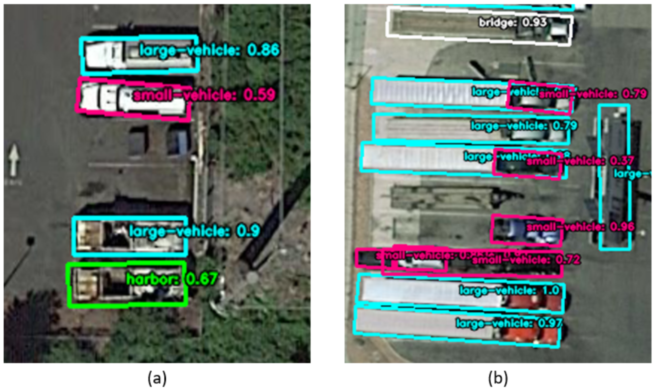

Fig. 19. Bad Case of our Pipeline. (a) Wrong Labeling. (b) Wrong Bounding Boxes of are Part of the Instance.

VI. Conclusion

Generally, our pipeline adopts very well to many lots of SOTA baselines, yielding promising results and solving problems of inconsistency between classification and localization. According to our experimental results, our pipeline yields promising results on the oriented DOTA dataset by extracting oriented bounding boxes and feeding to independent training cycles. In the future, we are researching more training and testing pipelines, seeking more baselines for oriented object detection. Our work introduces a fine pipeline for tackling mismatched features in classification, if exploited well enough, it will significantly boost detection performance.

Acknowledgment

This research is funded by Vietnam National University Ho Chi Minh City (VNU-HCM) under grant number DS2021-26-01. We also would like to show our gratitude to the UIT-Together research group for sharing their pearls of wisdom with us during this research.

References

[1] S. N. K. B. Amit and Y. Aoki, "Disaster detection from aerial imagery with convolutional neural network," in *2017 International Electronics Symposium on Knowledge Creation and Intelligent Computing (IES-KCIC)*, 2017, pp. 239–245.

[2] A. Ammar, A. Koubaa, M. Ahmed, A. Saad, and B. Benjdira, "Vehicle detection from aerial images using deep learning: A comparative study," *Electronics*, vol. 10, p. 820, 03 2021.

[3] K. Nguyen, P. Nguyen, D. C. Bui, M. Tran, and N. D. Vo, "Analysis of the influence of de-hazing methods on vehicle detection in aerial images," *International Journal of Advanced Computer Science and Applications*, vol. 13, no. 6, 2022. [Online]. Available: http://dx.doi.org/10.14569/IJACSA.2022.01306100

[4] X. Yang, J. Yan, Z. Feng, and T. He, "R3det: Refined single-stage detector with feature refinement for rotating object," 2019. [Online]. Available: https://arxiv.org/abs/1908.05612

[5] J. Ding, N. Xue, Y. Long, G.-S. Xia, and Q. Lu, "Learning roi transformer for detecting oriented objects in aerial images," 2018. [Online]. Available: https://arxiv.org/abs/1812.00155

[6] J. Ma, W. Shao, H. Ye, L. Wang, H. Wang, Y. Zheng, and X. Xue, "Arbitrary-oriented scene text detection via rotation proposals," *IEEE Transactions on Multimedia*, vol. 20, no. 11, pp. 3111–3122, nov 2018. [Online]. Available: https://doi.org/10.1109%2Ftmm.2018.2818020

[7] T. V. Le, H. N. N. Van, D. C. Bui, P. Vo, N. D. Vo, and K. Nguyen, "Empirical study of reppoints representation for object detection in aerial images," in *2022 IEEE Ninth International Conference on Communications and Electronics (ICCE)*, 2022, pp. 337–342.

[8] A. Ahmad, H. Sakidin, M. Y. A. Sari, S. F. Sufahani *et al.*, "Naïve bayes classification of high-resolution aerial imagery," *International Journal of Advanced Computer Science and Applications*, vol. 12, no. 11, 2021.

[9] R. Girshick, "Fast r-cnn," 2015. [Online]. Available: https://arxiv.org/abs/1504.08083

[10] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," 2015. [Online]. Available: https://arxiv.org/abs/1506.01497

[11] H. Zhang, H. Chang, B. Ma, N. Wang, and X. Chen, "Dynamic r-cnn: Towards high quality object detection via dynamic training," 2020. [Online]. Available: https://arxiv.org/abs/2004.06002

[12] X. Zhu, W. Su, L. Lu, B. Li, X. Wang, and J. Dai, "Deformable detr: Deformable transformers for end-to-end object detection," 2020. [Online]. Available: https://arxiv.org/abs/2010.04159

[13] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "SSD: Single shot MultiBox detector," in *Computer Vision – ECCV 2016*. Springer International Publishing, 2016, pp. 21–37. [Online]. Available: https://doi.org/10.1007%2F978-3-319-46448-0_2

[14] Q. Chen, Y. Wang, T. Yang, X. Zhang, J. Cheng, and J. Sun, "You only look one-level feature," 2021. [Online]. Available: https://arxiv.org/abs/2103.09460

[15] Z. Ge, S. Liu, F. Wang, Z. Li, and J. Sun, "Yolox: Exceeding yolo series in 2021," 2021. [Online]. Available: https://arxiv.org/abs/2107.08430

[16] J. Ding, N. Xue, Y. Long, G.-S. Xia, and Q. Lu, "Learning roi transformer for detecting oriented objects in aerial images," 2018. [Online]. Available: https://arxiv.org/abs/1812.00155

[17] Y. Jiang, X. Zhu, X. Wang, S. Yang, W. Li, H. Wang, P. Fu, and Z. Luo, "R2cnn: Rotational region cnn for orientation robust scene text detection," 2017. [Online]. Available: https://arxiv.org/abs/1706.09579

[18] J. Han, J. Ding, J. Li, and G.-S. Xia, "Align deep features for oriented object detection," 2020. [Online]. Available: https://arxiv.org/abs/2008.09397

[19] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16x16 words: Transformers for image recognition at scale," 2020. [Online]. Available: https://arxiv.org/abs/2010.11929

[20] M. Tan and Q. V. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," 2019. [Online]. Available: https://arxiv.org/abs/1905.11946

[21] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He, "Aggregated residual transformations for deep neural networks," 2016. [Online]. Available: https://arxiv.org/abs/1611.05431

[22] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” 2015. [Online]. Available: https://arxiv.org/abs/1512.03385

[23] ——, “Identity mappings in deep residual networks,” 2016. [Online]. Available: https://arxiv.org/abs/1603.05027

[24] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, “Rethinking the inception architecture for computer vision,” 2015. [Online]. Available: https://arxiv.org/abs/1512.00567

[25] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. Alemi, “Inception-v4, inception-resnet and the impact of residual connections on learning,” 2016. [Online]. Available: https://arxiv.org/abs/1602.07261

[26] 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,” 2014. [Online]. Available: https://arxiv.org/abs/1409.0575

[27] Q. M. Chung, T. D. Le, T. V. Dang, N. D. Vo, T. V. Nguyen, and K. Nguyen, “Data augmentation analysis in vehicle detection from aerial videos,” in *2020 RIVF International Conference on Computing and Communication Technologies (RIVF)*, 2020, pp. 1–3.

[28] A. Elhagry and M. Saeed, “Investigating the challenges of class imbalance and scale variation in object detection in aerial images,” 2022. [Online]. Available: https://arxiv.org/abs/2202.02489

[29] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, “Mobilenets: Efficient convolutional neural networks for mobile vision applications,” 2017. [Online]. Available: https://arxiv.org/abs/1704.04861

[30] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, “Mobilenetv2: Inverted residuals and linear bottlenecks,” 2018. [Online]. Available: https://arxiv.org/abs/1801.04381

[31] S. S. E. Mingxing Tan and G. A. Quoc V. Le, Principal Scientist. (2019) Efficientnet: Improving accuracy and efficiency through automl and model scaling. [Online]. Available: https://ai.googleblog.com/2019/05/efficientnet-improving-accuracy-and.html

[32] J. Han, J. Ding, N. Xue, and G.-S. Xia, “Redet: A rotation-equivariant detector for aerial object detection,” 2021. [Online]. Available: https://arxiv.org/abs/2103.07733

[33] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, “Feature pyramid networks for object detection,” 2016. [Online]. Available: https://arxiv.org/abs/1612.03144

[34] J. Ding, N. Xue, G.-S. Xia, X. Bai, W. Yang, M. Yang, S. Belongie, J. Luo, M. Datcu, M. Pelillo, and L. Zhang, “Object detection in aerial images: A large-scale benchmark and challenges,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–1, 2021. [Online]. Available: https://doi.org/10.1109%2Ftpami.2021.3117983

[35] Y. Zhou, X. Yang, G. Zhang, J. Wang, Y. Liu, L. Hou, X. Jiang, X. Liu, J. Yan, C. Lyu, W. Zhang, and K. Chen, “Mmrotate: A rotated object detection benchmark using pytorch,” *arXiv preprint arXiv:2204.13317*, 2022.

[36] M. Contributors, “Openmmlab’s image classification toolbox and benchmark,” https://github.com/open-mmlab/mmclassification, 2020.

[37] A. Gupta, P. Dollár, and R. Girshick, “Lvis: A dataset for large vocabulary instance segmentation,” 2019. [Online]. Available: https://arxiv.org/abs/1908.03195