AlphaRotate: A Rotation Detection Benchmark using TensorFlow

Xue Yang yangxue-2019-sjtu@sjtu.edu.cn
Junchi Yan* yanjunchi@sjtu.edu.cn
Department of Computer Science and Engineering
MoE Key Lab of Artificial Intelligence, AI Institute
Shanghai Jiao Tong University, Shanghai, China, 200240

Abstract

AlphaRotate is an open-source Tensorflow benchmark for performing scalable rotation detection on various datasets. It currently provides more than 18 popular rotation detection models under a single, well-documented API designed for use by both practitioners and researchers. AlphaRotate regards high performance, robustness, sustainability and scalability as the core concept of design, and all models are covered by unit testing, continuous integration, code coverage, maintainability checks, and visual monitoring and analysis. AlphaRotate is open source with an Apache-2.0 License and available at `https://github.com/yangxue0827/RotationDetection`.

Keywords: Rotation Detection, Convolutional Neural Network, Tensorflow,

1. Introduction

Despite the rich literature (Girshick, 2015; Ren et al., 2015; Dai et al., 2016; Lin et al., 2017a,b) in visual object detection in computer vision, existing models are mostly agnostic to the object orientation, which only output a horizontal bounding box. This can be restricted in real-world settings whereby either the rotation information itself can be critical e.g. for aerial observation, or the rotated detecting bounding box can better align the object for more accurate recognition, especially for small and densely arranged objects.

For the above reasons, recently rotation detectors emerge with the development in terms of both backbone and loss design. The applications are arranged across aerial images (Yang et al., 2018a,b, 2019, 2021b), scene text (Zhou et al., 2017; Jiang et al., 2017; Liu et al., 2018; Ma et al., 2018; Liao et al., 2018), faces (Shi et al., 2018), 3D objects (Zheng et al., 2020), and retail scenes (Chen et al., 2020; Pan et al., 2020) etc.

However, there lacks an open-source benchmark integrating recent advance rotation detection models for evaluation and use. The most popular object detection benchmarks, e.g. MMDetection (Chen et al., 2019a), Detectron2 (Wu et al., 2019), SimpleDet (Chen et al., 2019b), are all focused on horizontal detection. AerialDetection[1] is an earlier rotation detection benchmark based on MMDetection. However, it only provides some baselines and few methods, and it lacks maintenance and integration of new methods. Moreover, all these benchmarks are based on Pytorch (Paszke et al., 2017) which can be less efficient than Tensorflow (Abadi et al., 2016) in terms of industrial deployment.

*. Junchi Yan is the correspondence author.
1. `https://github.com/dingjiansw101/AerialDetection`

Baseline	Improvement	Box Def.	v1.0	v1.0 train/val			v1.5	v2.0
			AP_{50}	AP_{50}	AP_{75}	$AP_{50:95}$	AP_{50}	AP_{50}
RetinaNet-H (Lin et al., 2017b)	-	*O.C.*	65.73	64.70	32.31	34.50	58.87	44.16
	-	*L.E.*	64.17	62.21	26.06	31.49	56.10	43.06
	IoU-Smooth L1 (Yang et al., 2019)	*O.C.*	66.99	64.61	34.17	36.23	59.16	46.31
	RSDet (Qian et al., 2021)	*O.C.*	66.05	63.50	33.32	34.61	57.75	45.17
	RSDet (Qian et al., 2021)	Quad.	67.20	65.15	40.59	39.12	61.42	46.71
	RIDet (Ming et al., 2021)	Quad.	66.06	64.07	40.98	39.05	58.91	45.35
	CSL (Yang and Yan, 2020)	*L.E.*	67.38	64.40	32.58	35.04	58.55	43.34
	DCL (BCL) (Yang et al., 2021a)	*L.E.*	67.39	65.93	35.66	36.71	59.38	45.46
	GWD (Yang et al., 2021c)	*O.C.*	68.93	65.44	38.68	38.71	60.03	46.65
	KLD (Yang et al., 2021d)	*O.C.*	71.28	68.14	44.48	42.15	62.50	47.69
RetinaNet-R	-	*O.C.*	67.25	65.00	33.68	35.16	56.50	42.04
FCOS (Tian et al., 2019)	-	Quad.	67.69	65.73	35.70	36.62	61.05	48.10
	RSDet (Qian et al., 2021)	Quad.	67.91	66.07	38.90	38.25	62.18	48.81
R^3Det (Yang et al., 2021b)	-	*O.C.*	70.66	67.18	38.41	38.46	62.91	48.43
	DCL (BCL) (Yang et al., 2021a)	*L.E.*	71.21	67.45	35.44	37.54	61.98	48.71
	GWD (Yang et al., 2021c)	*O.C.*	71.56	69.28	43.35	41.56	63.22	49.25
	KLD (Yang et al., 2021d)	*O.C.*	71.73	68.87	44.48	42.11	65.18	50.90
FPN (Lin et al., 2017a)	R^2CNN (Jiang et al., 2017)	*O.C.*	72.27	68.43	34.74	37.08	66.45	52.35
	KLD (Yang et al., 2021d)	*O.C.*	72.16	68.78	39.25	39.58	65.59	51.30
Faster RCNN (Ren et al., 2015)	SCRDet (Yang et al., 2019)	*O.C.*	71.73	-	-	-	-	-

Table 1: Accuracy comparison of rotation detectors on DOTA (v1.0, v1.5 and v2.0) (Xia et al., 2018). *O.C.* and *L.E.* represent OpenCV Definition ($\theta \in [-90^\circ, 0^\circ)$) and Long Edge Definition ($\theta \in [-90^\circ, 90^\circ)$) of RBox. 'H' and 'R' represent the horizontal and rotating anchors, respectively. All models are initialized by ResNet50 (He et al., 2016) without using data augmentation and multi-scale training and testing.

To fill this gap, we propose and implement AlphaRotate – a Tensorflow which consists of many state-of-the-art detection techniques and models (keep on extension) – mostly from our previous works (Yang et al., 2018a,b, 2019; Yang and Yan, 2020; Qian et al., 2021; Yang et al., 2021a,b,c,d), which can be readily used for both industry and academic community, with the following features, as shown in Figure 1 and Table 1.

- AlphaRotate is one of the first rotation detection benchmarks (based on Tensorflow), and it supports training and testing on various datasets including aerial images, scene text, and face. It supports state-of-the-art rotation detectors (including hybrid methods), and provides comprehensive evaluation on DOTA (Xia et al., 2018) dataset.

- A clean and modular implementation, which eases future integration of new methods and facilitates assembling different modules, such as Backbone, Neck, Loss, etc.

- AlphaRotate includes a detailed installation and tutorials across all models for clarity and ease of use, and all models are covered by unit testing, continuous integration, code coverage, maintainability checks, and visual monitoring and analysis.

- AlphaRotate supports multi-GPU training and multi-process testing, and provides commonly used techniques such as data augmentation, multi-scale training and cropping, stochastic weights averaging, etc. to further boost model performance.

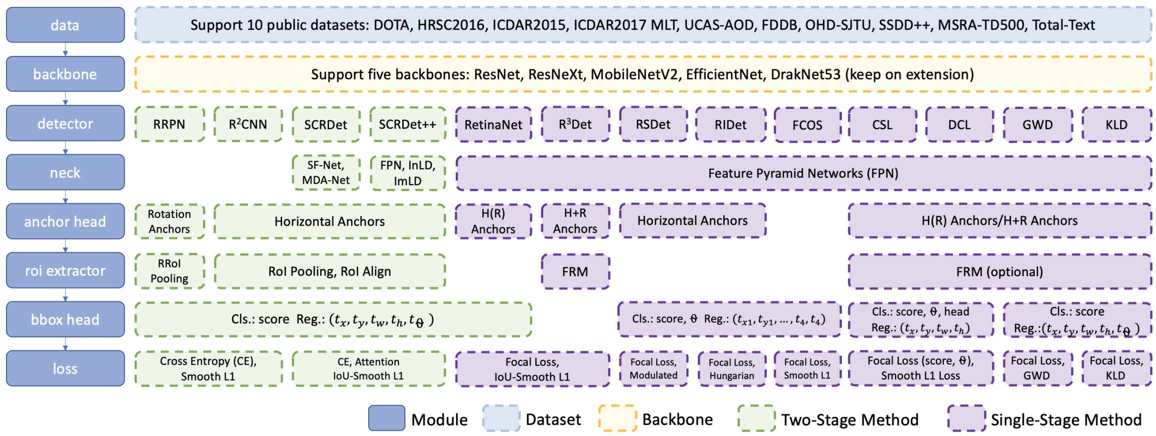

Figure 1: AlphaRotate supports state-of-the-art rotation detection methods, and supports training/testing on public datasets with aerial images, scene text, and face etc.

2. Project Focus

Modular implementation: As shown in Figure 1, the detectors are organized by eight components: *data*, *backbone*, *detector*, *neck*, *anchor head*, *roi extractor*, *bbox head*, *loss*. The core of AlphaRotate is a set of base classes and functions that is designed to allow for rapid and easy development of such models. Developers only need to add/delete/improve specific modules to build a new rotation detection model. The modular design improves code utilization and also helps developers to debug and troubleshoot. Through our test procedure, all functions and classes are tested, with a line coverage of over 92% of the code.
Rich models and tools: AlphaRotate supports more than 18 state-of-the-art rotation detection methods (keep on extension), including single-stage/two-stage methods, anchor-based/anchor-free methods, and supports training and testing on nearly ten datasets such as aerial images, scene text, and face, as shown in Figure 1. Besides, it provides fair comparisons of all methods (including hybrid methods) on representative DOTA dataset (Xia et al., 2018) to provide researchers with an accurate and comprehensive baseline, as listed in Table 1. Unless otherwise specified, all models are trained on the trainval set by default, and the result file of the test set is submitted to the official evaluation server[2] to obtain the final evaluation result. We have provided download links of all baseline model weights.
Open and collaborative: We have implemented AlphaRotate with an open and collaborative spirit and placed it under Apache-2.0 License. AlphaRotate is hosted on GitHub[3], and developers can consult and discuss issues through the platform. In addition, external contributions and requests are encouraged and we enforce a strict rule on providing several tests for every new contribution and detected bug. At the time of writing this paper, about 500 stars and 100 forks have been created and 37 issues have been resolved.
Installation and tutorials: AlphaRotate provides installation instructions, including dependent libraries and platform environment. We also provide developers with the Docker

2. `https://captain-whu.github.io/DOTA/evaluation.html`
3. `https://github.com/yangxue0827/RotationDetection`

image matched by the benchmark. We also provide detailed tutorials[4], including how to convert dataset, compile files, train and test, and use visualization tools, etc.

Dependencies: AlphaRotate is built as a Python 2 or 3 application on top Tensorflow (Abadi et al., 2016). It allows to use the benchmark on various platforms and devices like CPUs and GPUs. AlphaRotate relies on open source libraries such as *NumPy*, *SciPy*, *Cython*, *OpenCV-Python*, *Matplotlib*, *Shapely*, which are commonly used in literature.

3. Toolbox Usage

The benchmark provides recent state-of-the-art rotation detection models related to deep learning. AlphaRotate has sub-modules dedicated to different stages, such as data pre-processing, model training, model testing and evaluation, visual analysis, etc. In this section, we give a short description of the provided benchmark.

Data pre-processing: The input of the model mainly includes the image I, the four-point coordinates of the object $(x_1, y_1, .., x_4, y_4)$ and the category corresponding to the object L. If the image size of the data set is very large, such as the DOTA dataset, sliding window method is used to slice the image and generate the label of the corresponding sub-image. AlphaRotate provides the corresponding cropping script under the *dataloader/dataset*, which can set the cropped window size and sliding stride in a custom way. Finally, trained dataset should be converted into tensorflow's exclusive data reading format *tfrecord* by using *dataloader/convert_data_to_tfrecord.py*.

Model training: First, select the detector and dataset, and create the corresponding configuration file according to the plan under *libs/configs*. Then run the training script under *tools* until the training is complete. During the training process, tensorbord can be used to observe the training status, including the loss curve of each stage and the image visualization. All trained models and log files are saved in the *output*.

Model testing and evaluation: Under the *tools*, it provides test and evaluation scripts for evaluation (e.g. mAP, F-measure), visualizing results, and generating result files.

4. Conclusion

With the practical importance and academic emergence for visual rotation detection, AlphaRotate is a deep learning benchmark for visual object rotation detection in Tensorflow under the Apache-2.0 license. The architecture is designed for both flexibility and ease of use, with the goal of facilitating the deployment of rotation detection in diverse domains, both in industrial applications and in academic research. We will continue to improve the entire optimized benchmark and support representative detection methods in the future. We also welcome the community to participate in the development.

Acknowledgments

The work was supported by China Key Research and Development Program (2020AAA0107600) and Shanghai Municipal Science and Technology Major Project (2021SHZDZX0102).

4. https://alpharotate-tutorial.readthedocs.io/

References

M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, et al. Tensorflow: A system for large-scale machine learning. In *12th {USENIX} symposium on operating systems design and implementation ({OSDI} 16)*, pages 265–283, 2016.

K. Chen, J. Wang, J. Pang, Y. Cao, Y. Xiong, X. Li, S. Sun, W. Feng, Z. Liu, J. Xu, Z. Zhang, D. Cheng, C. Zhu, T. Cheng, Q. Zhao, B. Li, X. Lu, R. Zhu, Y. Wu, J. Dai, J. Wang, J. Shi, W. Ouyang, C. C. Loy, and D. Lin. MMDetection: Open mmlab detection toolbox and benchmark. *arXiv preprint arXiv:1906.07155*, 2019a.

Y. Chen, C. Han, Y. Li, Z. Huang, Y. Jiang, N. Wang, and Z. Zhang. Simpledet: A simple and versatile distributed framework for object detection and instance recognition. *Journal of Machine Learning Research*, 20(156):1–8, 2019b. URL `http://jmlr.org/papers/v20/19-205.html`.

Z. Chen, K. Chen, W. Lin, J. See, H. Yu, Y. Ke, and C. Yang. Piou loss: Towards accurate oriented object detection in complex environments. *Proceedings of the European Conference on Computer Vision*, 2020.

J. Dai, Y. Li, K. He, and J. Sun. R-fcn: Object detection via region-based fully convolutional networks. In *Advances in neural information processing systems*, pages 379–387, 2016.

R. Girshick. Fast r-cnn. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1440–1448, 2015.

K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.

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. *arXiv preprint arXiv:1706.09579*, 2017.

M. Liao, Z. Zhu, B. Shi, G.-s. Xia, and X. Bai. Rotation-sensitive regression for oriented scene text detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5909–5918, 2018.

T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2117–2125, 2017a.

T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2980–2988, 2017b.

X. Liu, D. Liang, S. Yan, D. Chen, Y. Qiao, and J. Yan. Fots: Fast oriented text spotting with a unified network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5676–5685, 2018.

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*, 20(11): 3111–3122, 2018.

Q. Ming, Z. Zhou, L. Miao, X. Yang, and Y. Dong. Optimization for oriented object detection via representation invariance loss. *arXiv preprint arXiv:2103.11636*, 2021.

X. Pan, Y. Ren, K. Sheng, W. Dong, H. Yuan, X. Guo, C. Ma, and C. Xu. Dynamic refinement network for oriented and densely packed object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 11207–11216, 2020.

A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer. Automatic differentiation in pytorch. 2017.

W. Qian, X. Yang, S. Peng, J. Yan, and Y. Guo. Learning modulated loss for rotated object detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 2458–2466, 2021.

S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015.

X. Shi, S. Shan, M. Kan, S. Wu, and X. Chen. Real-time rotation-invariant face detection with progressive calibration networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2295–2303, 2018.

Z. Tian, C. Shen, H. Chen, and T. He. Fcos: Fully convolutional one-stage object detection. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 9627–9636, 2019.

Y. Wu, A. Kirillov, F. Massa, W.-Y. Lo, and R. Girshick. Detectron2. `https://github.com/facebookresearch/detectron2`, 2019.

G.-S. Xia, X. Bai, J. Ding, Z. Zhu, S. Belongie, J. Luo, M. Datcu, M. Pelillo, and L. Zhang. Dota: A large-scale dataset for object detection in aerial images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3974–3983, 2018.

X. Yang and J. Yan. Arbitrary-oriented object detection with circular smooth label. In *Proceedings of the European Conference on Computer Vision*, pages 677–694. Springer, 2020.

X. Yang, H. Sun, K. Fu, J. Yang, X. Sun, M. Yan, and Z. Guo. Automatic ship detection in remote sensing images from google earth of complex scenes based on multiscale rotation dense feature pyramid networks. *Remote Sensing*, 10(1):132, 2018a.

X. Yang, H. Sun, X. Sun, M. Yan, Z. Guo, and K. Fu. Position detection and direction prediction for arbitrary-oriented ships via multitask rotation region convolutional neural network. *IEEE Access*, 6:50839–50849, 2018b.

X. Yang, J. Yang, J. Yan, Y. Zhang, T. Zhang, Z. Guo, X. Sun, and K. Fu. Scrdet: Towards more robust detection for small, cluttered and rotated objects. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 8232–8241, 2019.

X. Yang, L. Hou, Y. Zhou, W. Wang, and J. Yan. Dense label encoding for boundary discontinuity free rotation detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 15819–15829, 2021a.

X. Yang, J. Yan, Z. Feng, and T. He. R3det: Refined single-stage detector with feature refinement for rotating object. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 3163–3171, 2021b.

X. Yang, J. Yan, M. Qi, W. Wang, Z. Xiaopeng, and T. Qi. Rethinking rotated object detection with gaussian wasserstein distance loss. In *International Conference on Machine Learning*, 2021c.

X. Yang, X. Yang, J. Yang, Q. Ming, W. Wang, Q. Tian, and J. Yan. Learning high-precision bounding box for rotated object detection via kullback-leibler divergence. *arXiv preprint arXiv:2106.01883*, 2021d.

Y. Zheng, D. Zhang, S. Xie, J. Lu, and J. Zhou. Rotation-robust intersection over union for 3d object detection. In *European Conference on Computer Vision*, pages 464–480. Springer, 2020.

X. Zhou, C. Yao, H. Wen, Y. Wang, S. Zhou, W. He, and J. Liang. East: an efficient and accurate scene text detector. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5551–5560, 2017.