Introduction

Rotation Object Detection

- Task: design a novel multi-category rotation detector for small, cluttered and rotated objects.
- Challenges
  - Large aspect ratio. The Sketch Intersection over Union (SketchIoU) score between large aspect ratio objects is sensitive to change in angle.
  - Densely arranged. Many objects usually appear in densely arranged forms.
  - Arbitrary orientations. Objects in images can appear in various orientations, which requires the detector to have accurate direction estimation capabilities.
- Our main contributions
  - For large aspect ratio object detection, an accurate and fast rotation single-stage detector is devised in a refined manner, for high-precision detection. In contrast to the recent learning based methods for feature alignment, which lacks an explicit mechanism to compensate the misalignment, we propose a direct and effective pure computing based approach which is further extended to handle the rotation case.
  - For densely arranged objects, we develop an efficient coarse-to-fine progressive recognition approach to better exploring the two forms of anchors in a more flexible manner, tailored to each detection stage.
  - For arbitrarily-rotated objects, a derivable approximate SkewIoU loss is devised for more accurate rotation estimation.
- Codes: https://github.com/Thinklab-SJTU/R3Det_TensorFlow

Proposed Approach

Pipeline

We give an overview of our method as sketched in Fig. 1. The embodiment is a refined single-stage rotation detector based on the RetinaNet, namely Refined Rotation RetinaNet (R3Det). The refinement stage (which can be added and repeated by multiple times) is added to the network to refine the bounding box, and the feature refinement module FRM is added during the refinement stage to reconstruct the feature map. In a single-stage rotating object detection task, continuous refinement of the predicted bounding box can improve the regression accuracy, and feature refinement is a necessary process for this purpose.

Feature Refinement Module

- Fig. 2 shows the structure of feature refinement module. Specifically, the feature map is added by two-way convolution to obtain a new feature (large kernel, LK). Only the bounding box with the highest score of each feature point is preserved in the refinement stage to increase the speed (box filtering, BF), meanwhile ensuring that each feature point corresponds to only one refined bounding box. The filtering of bounding boxes is a necessary step for feature reconstruction (FR). For each feature point of the feature map, we obtain the corresponding feature vector on the feature map according to the five coordinates of the refined bounding box (one center point and four corner points). A more accurate feature vector is obtained by bilinear interpolation. We add the five feature vectors and replace the current feature vector. After traversing the feature points, we reconstruct the whole feature map. Finally, the reconstructed feature map is added to the original feature map to complete the whole process.

A Derivable Approximate SkewIoU Loss

- As shown in Fig. 3, each box set has the same center point, height and width. The angle difference between the two box sets is the same, but the aspect ratio is different. As a result, the smooth l1 loss value of the two sets is the same (mainly from the angle difference), but the SkewIoU is quite different. The IoU related loss is an effective regression loss function that can solve above problem and is already widely used in horizontal detection. However, the SkewIoU calculation function between two rotating boxes is undervisible, which means that we cannot directly use the SkewIoU as the regression loss function. Inspired by SCRDet, we propose a derivable approximate SkewIoU loss, the multi-task loss is defined as follows:

\[
L = \frac{1}{N} \sum_{i=1}^{N} \sum_{j=1}^{M} \left[ \text{IoU}(\hat{b}_i, b_j) - \left( \frac{w_i \cdot w_j}{w_{ij}} \right) \right] + \frac{1}{N} \sum_{i=1}^{N} \sum_{j=1}^{M} \left[ \text{IoU}(\hat{b}_i, b_j) \right]
\]

\[
L_{\text{smooth}}(\hat{b}_i, b_j) = \frac{1}{N} \sum_{i=1}^{N} \text{IoU}(\hat{b}_i, b_j) - \text{IoU}(\hat{b}_i, b_j) \frac{w_i \cdot w_j}{w_{ij}} \frac{w_i \cdot w_j}{w_{ij}}
\]

Experiments

- Ablative study of each component in our method on the DOTA dataset.
- Ablation study for number of stages on DOTA.
- Comparison between R3Det and R3Det on three datasets.
- Experiments with our FRM with different interpolation formulas.
- Detection accuracy on DOTA.
- Experiments with different SkewIoU functions.
- Evaluation on HRSC2016.
- Detection accuracy on UCAS-AOD.