```python
    def computeIoU(self, imgId, catId):
        p = self.params
        if p.useCats:
            gt = self._gts[imgId, catId]
            dt = self._dts[imgId, catId]
        else:
            gt = [_ for cId in p.catIds for _ in self._gts[imgId, cId]]
            dt = [_ for cId in p.catIds for _ in self._dts[imgId, cId]]
        if len(gt) == 0 and len(dt) == 0:
            return []
        inds = np.argsort([-d['score'] for d in dt], kind='mergesort')
        dt = [dt[i] for i in inds]
        if len(dt) > p.maxDets[-1]:
            dt = dt[0:p.maxDets[-1]]      You, 25 minutes ago • Uncommitted

        if p.iouType == 'segm':
            g = [g['segmentation'] for g in gt]
            d = [d['segmentation'] for d in dt]
        elif p.iouType == 'bbox':
            g = [g['bbox'] for g in gt]
            d = [d['bbox'] for d in dt]
        else:
            raise Exception('unknown iouType for iou computation')

        # compute iou between each dt and gt region
        iscrowd = [int(o['iscrowd']) for o in gt]
        ious = maskUtils.iou(d, g, iscrowd)
        return ious
```

cocoeval.py

cocoeval.py

```
284            if not len(ious) == 0:
285                for tind, t in enumerate(p.iouThrs):
286                    for dind, d in enumerate(dt):
287                        # information about best match so far (m=-1 -> unmatched)
288                        iou = min([t, 1-1e-10])
289                        m = -1
290                        for gind, g in enumerate(gt):
291                            # if this gt already matched, and not a crowd, continue
292                            if gtm[tind, gind] > 0 and not iscrowd[gind]:
293                                continue
294                            # if dt matched to reg gt, and on ignore gt, stop
295                            if m > -1 and gtIg[m] == 0 and gtIg[gind] == 1:
296                                break
297                            # continue to next gt unless better match made
298                            if ious[dind, gind] < iou:
299                                continue
300                            # if match successful and best so far, store appropriately
301                            iou = ious[dind, gind]
302                            m = gind
303                        # if match made store id of match for both dt and gt
304                        if m == -1:
305                            continue
306                        dtIg[tind, dind] = gtIg[m]
307                        dtm[tind, dind] = gt[m]['id']
308                        gtm[tind, m] = d['id']
```

m=》当前预测最优匹配的gt的index

lscrowd是一组对象还是单个对象

每个prediction匹配一个iou最大的ground truth

gtIg[m]==0，当前找到的gt是不是要被ignore的
gtIg[gind]==1，当前遍历的gt是不是要被忽略的（提前终止loop，因为ignore的在list的后部）

dtIg[tind,dind]
在第tind个threshold下，第dind个预测是否是被忽略的，赋值为最优匹配的gt的值

dtm[tind,dind]
在第tind个threshold下，第dind个预测的最优匹配gt

gtm[tind,m]
在第tind个threshold下，第m个gt的最优匹配预测

```
286         dtIg = np.zeros((T, D))       You, 2 hours ago • Uncommitted changes
287         if not len(ious) == 0:
288             for tind, t in enumerate(p.iouThrs):
289                 for dind, d in enumerate(dt):
290                     # information about best match so far (m=-1 -> unmatched)
291                     iou = min([t, 1-1e-10])
292                     m = -1
293                     for gind, g in enumerate(gt):
294                         # if this gt already matched, and not a crowd, continue
295                         if gtm[tind, gind] > 0 and not iscrowd[gind]:
296                             continue
297                         # if dt matched to reg gt, and on ignore gt, stop
298                         if m > -1 and gtIg[m] == 0 and gtIg[gind] == 1:
299                             break
300                         # continue to next gt unless better match made
301                         if ious[dind, gind] < iou:
302                             continue
303                         # if match successful and best so far, store appropriately
304                         iou = ious[dind, gind]
305                         m = gind
306                     # if match made store id of match for both dt and gt
307                     if m == -1:
308                         continue
                        dtIg[tind, dind] = gtIg[m]
                        dtm[tind, dind] = gt[m]['id']
                        gtm[tind, m] = d['id']
    # set unmatched detections outside of area range to ignore
    a = np.array([d['area'] < aRng[0] or d['area'] > aRng[1]
                  for d in dt]).reshape((1, len(dt)))
    dtIg = np.logical_or(dtIg, np.logical_and(
        dtm == 0, np.repeat(a, T, 0)))
```

cocoeval.py

在找到匹配后，这个预测是不是被忽略与gt相同

1. 初始全零
2. dtIg[t,d] 第t个iou阈值下，第d个预测是否被忽略

```
tps = np.logical_and(dtm,  np.logical_not(dtIg))
fps = np.logical_and(
    np.logical_not(dtm), np.logical_not(dtIg))
```

a=》预测中面积过小或过大的部分
最终，dtIg[t,d]=false需要
dtIg=0 并且（and 为false）
DtIg=0=》预测无匹配或者预测对应gt不被忽略
And为false=》dtm==0 或者a为false=》此预测有匹配或者无匹配预测面积正常
dtIg=false代表
所有面积正常的预测

1. Iscrowd will determine whether multiple prediction match one ground truth(but ground truth is always matched to one prediction, but gtm doesn’t seem important)
2. Filter by area(bbox of too big or too small areas)
3. Predictions are sorted by score when matching ground truth