dotademo

December 2, 2021

1 Usage for parse ground truth file and show

This is a demo for parse, visulize and split the data, the demo use the data in example folder, but the folder does not neccessaryly include all the categories.

[24]:
```
%matplotlib inline

os.getcwd()
p = r'D:\geo\devkitmod'
os.chdir(p)

import numpy as np
import matplotlib.pyplot as plt
import os
from DOTA import DOTA
import dota_utils as util
import pylab

import matplotlib.pyplot as plt
from matplotlib.collections import PatchCollection
from matplotlib.patches import Polygon, Circle
import numpy as np
import dota_utils as util
from collections import defaultdict
import cv2
pylab.rcParams['figure.figsize'] = (20.0, 20.0)
```

[31]:
```
example = DOTA('example')
example = DOTA('part1')
```

[29]:
```
imgids = example.getImgIds(catNms=['ship', 'storage-tank'])
imgid = imgids[0]
img = example.loadImgs(imgid)[0]
```

```
isarralike: False
imgids: ['P1234']
filename: example\images\P1234.png
```

[30]:
```
plt.axis('off')

plt.imshow(img)
plt.show()
```

[1]:
```
anns = example.loadAnns(imgId=imgid)
# print(anns)
example.showAnns(anns, imgid, 2)
```

```
 ␣
↪---------------------------------------------------------------------------
```

```
NameError                                 Traceback (most recent call␣
↪last)

<ipython-input-1-87c6f3b91494> in <module>
----> 1 anns = example.loadAnns(imgId=imgid)
      2 # print(anns)
      3 example.showAnns(anns, imgid, 2)


NameError: name 'example' is not defined
```

1.1 Split Image And Label

We provide the scale param before split the images and labels. Sometimes, the instance is too large that it can be easily cut down(for example, ground track filed), in such case you need to set the param "rate" less than 1.

Before going on, first create folder to store the split data

```
mkdir examplesplit
mkdir examplesplit/images
mkdir examplesplit/labelTxt
```

For test images, you only need to split images, refer to "SplitOnlyImage.py"

[32]:
```
from ImgSplit import splitbase
```

[33]:
```
split = splitbase(r'part1',
                  r'part1split', choosebestpoint=True)
split.splitdata(0.5)
split.splitdata(1)
split.splitdata(2)
```

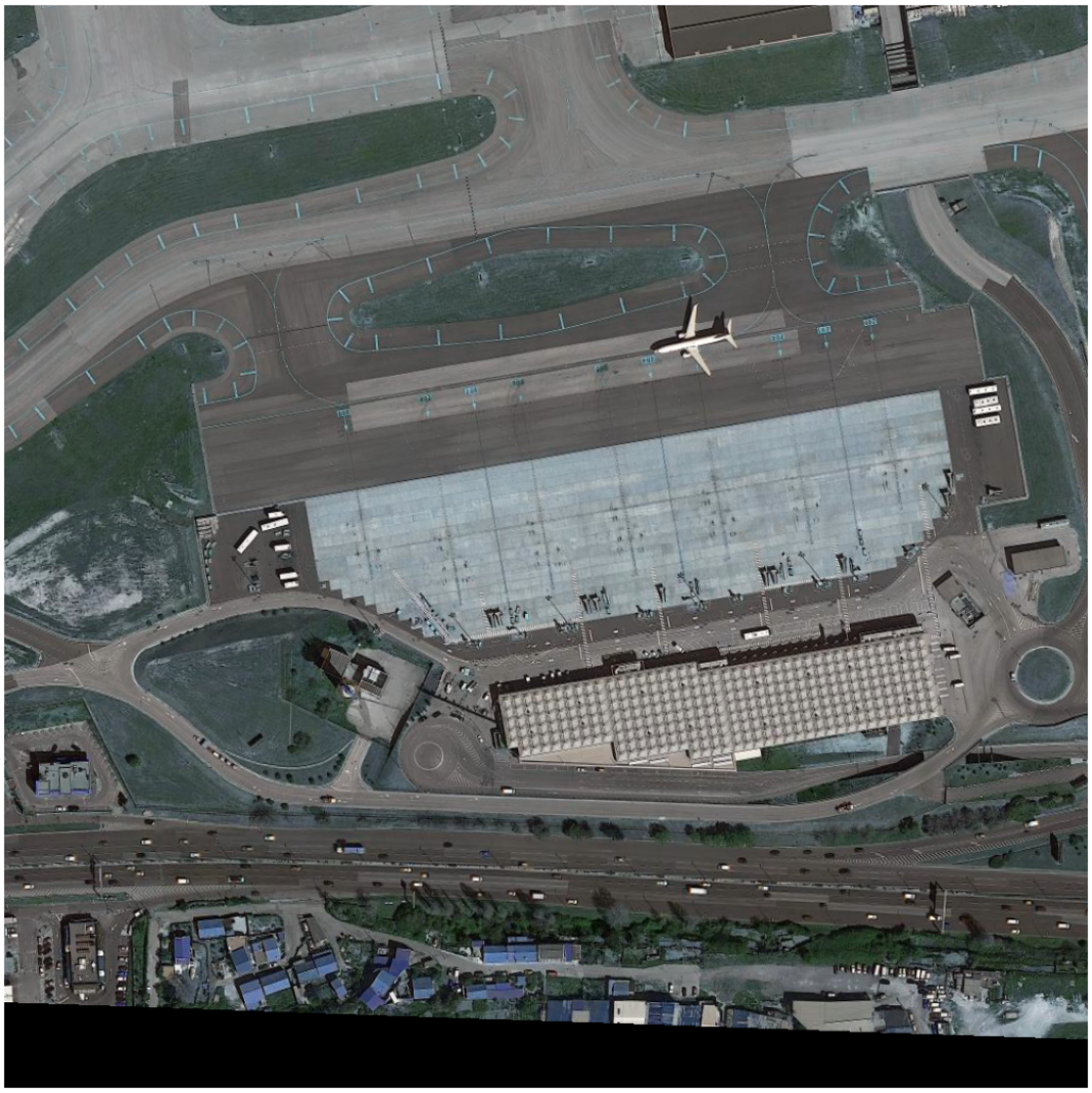
[34]:
```
#examplesplit = DOTA('examplesplit')
examplesplit = DOTA('part1split')
```

[35]:
```
imgids = examplesplit.getImgIds(catNms=['plane'])
imgid = imgids[1]
img = examplesplit.loadImgs(imgid)[0]
```

```
isarralike: False
imgids: ['P0160__1__924___1791']
filename: part1split\images\P0160__1__924___1791.png
```

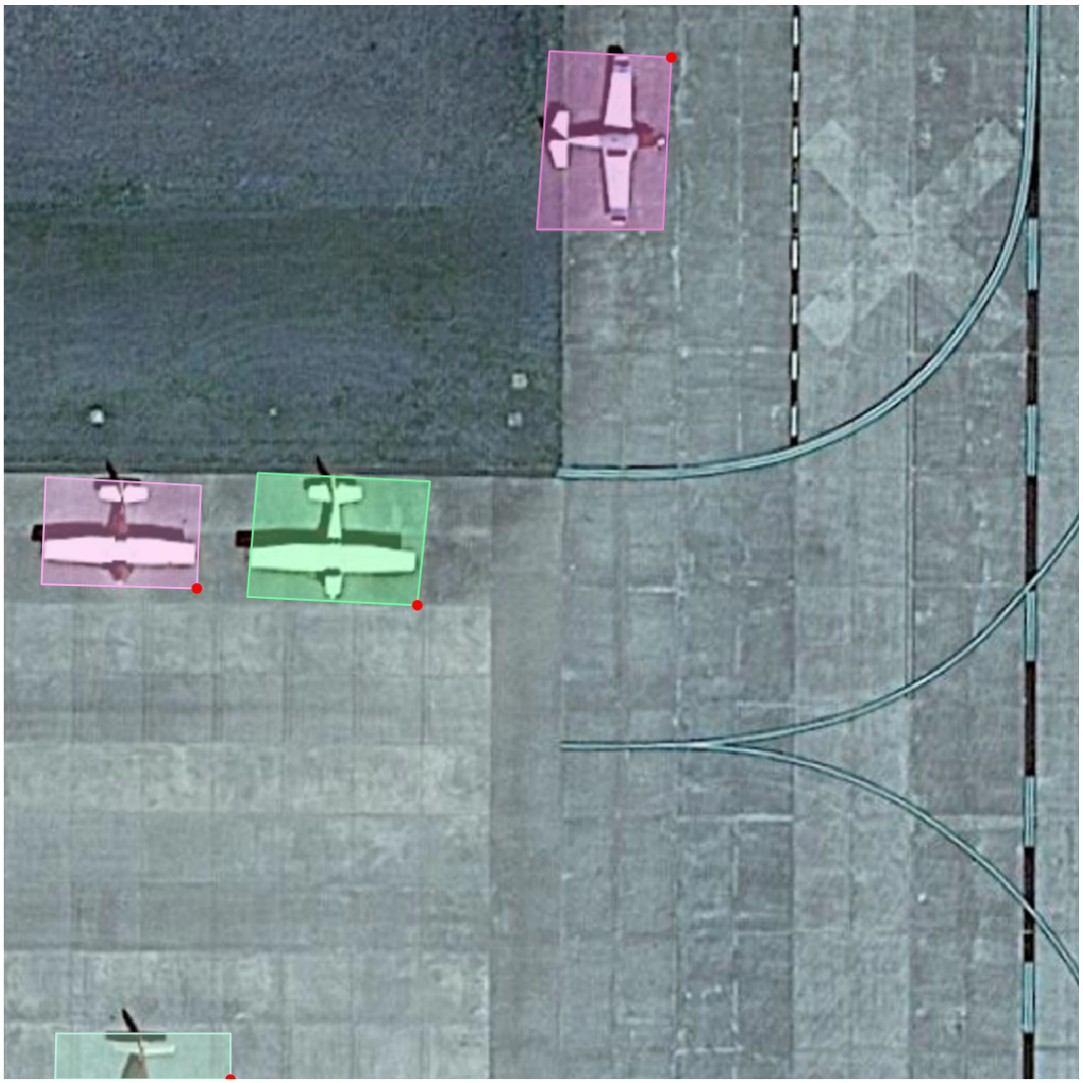
[36]:
```
plt.axis('off')

plt.imshow(img)
plt.show()
```



[10]:
```
anns = examplesplit.loadAnns(imgId=imgid)
# print(anns)
examplesplit.showAnns(anns, imgid, 2)
```

```
isarralike: False
imgids: ['P1088__2__1848___0']
filename: examplesplit/images/P1088__2__1848___0.png
```



1.2 Merge patches

Now, we will merge these patches to see if they can be restored in the initial large images

[11]:
```
from ResultMerge import mergebypoly
```

[12]:
```
util.groundtruth2Task1(r'examplesplit/labelTxt',
                       r'Task1')
mergebypoly(r'Task1',
            r'Task1_merge')
util.Task2groundtruth_poly(r'Task1_merge',
                           r'restoredexample/labelTxt')
```

[13]:
```
filepath = 'example/labelTxt'
imgids = util.GetFileFromThisRootDir(filepath)
imgids = [util.custombasename(x) for x in imgids]
print(imgids)
```

```
['P0770', 'P1234', 'P1088', 'P2709', 'P0706', 'P1888', 'P2598']
```

[14]:
```
example = DOTA(r'example')
num = 2
anns = example.loadAnns(imgId=imgids[num])
# print(anns)
example.showAnns(anns, imgids[num], 2)
```

```
isarralike: False
imgids: ['P1088']
filename: example/images/P1088.png
```

[15]:
```
restored = DOTA(r'restoredexample')
num = 2
anns = restored.loadAnns(imgId=imgids[num])
# print(anns)
restored.showAnns(anns, imgids[num], 2)
```

```
isarralike: False
imgids: ['P1088']
```

filename: restoredexample/images/P1088.png