Temdet

Project use thg NMS & Template that choosen by manual work algorithm to detect object with minimum interclass variance.

Algorithm Guide

```
import matplotlib.pyplot as plt
from temdet import nmsdet
detector=nmsdet("/workspace/data/clip/clip1.tif",CHANNEL_INDEX=
[0,1,2,3])#For NDVI
#detector=nmsdet("/workspace/data/clip/clip1.tif",CHANNEL_INDEX=
[0,1,2])#For RGB
```

```
# -------------------------------------------------------------------------
--- #
#                    NMS&Template MApping Toolkit
#
# -------------------------------------------------------------------------
--- #
-----Template NMS detect in :
 /workspace/data/clip/clip1.tif
 Mon Apr 27 16:21:57 2020
# -------------------------------------------------------------------------
--- #
#                            TIFF process Toolkit
#
# -------------------------------------------------------------------------
--- #
-----TIFF Class Init with : /workspace/data/clip/clip1.tif
-----Original Data Shape :  (4, 1303, 1631)
image type : int16
image shape (1303, 1631, 4)
```

```
ndvi=detector.buildndvi()
plt.imshow(ndvi),plt.show()
```

png

```
(<matplotlib.image.AxesImage at 0x7f0bfebe0908>, None)
```

```
detector.addbox(0,0,100,100)
detector.detect(drawbox=True)
print("boxes",detector.boxes,"conf",detector.conf)
```

```
  0%|          | 0/2 [00:00<?, ?it/s]
```

png

```
 50%|█████     | 1/2 [00:00<00:00,  4.36it/s]

feature  0  mapping : 2
```

png

```
100%|██████████| 2/2 [00:00<00:00,  4.99it/s]
1it [00:00, 5275.85it/s]

feature  0  mapping : 1



Clipping input data to the valid range for imshow with RGB data ([0..1] for
floats or [0..255] for integers).
```

png

```
boxes [(0, 0, 100, 100)] conf [0.7265027]
```

```
detector.addbox(100,100,200,200)
detector.detect(drawbox=True)
print("boxes",detector.boxes,"conf",detector.conf)
```

```
  0%|          | 0/3 [00:00<?, ?it/s]
```

png

```
 33%|███        | 1/3 [00:00<00:00,  4.13it/s]

feature  0  mapping : 2
```

png

```
 67%|██████     | 2/3 [00:00<00:00,  4.47it/s]

feature  0  mapping : 2
```

png

```
100%|██████████| 3/3 [00:00<00:00,  5.17it/s]
2it [00:00, 3833.92it/s]

feature  0  mapping : 2



Clipping input data to the valid range for imshow with RGB data ([0..1] for
floats or [0..255] for integers).
```

png

```
boxes [(0, 0, 100, 100), (100, 100, 200, 200)] conf [0.78325445 0.79081626]
```

```

```

AI Label Toolkit for GIS

This part work for data format transform on GIS imagery to generate dataset for AI training work.

Install

Guide to start

Document

- Temdet
  - Algorithm Guide
  - AI Label Toolkit for GIS

TIF file IO

Read

getfiles_from_dir

Set

Write

writeimagery

writetif

writethreshold2shp

Process

fast_percentager_strentching

resize_raster

Cord Transform

geo2lonlat

lonlat2geo

imagexy2geo

geo2imagexy

#####   lonlat2imagexy

imagexy2lonlat

Shapefile IO

readshp

feature_field_Defn

statsic

format transform

shp2LabelmeJson

json2Voclike

```
    Init shapes contents
    Use like : initlabeljson()
['shapes'].append(initcordarrary(Type,Cordlist))
    :param Type: Type in Shapefile with (['Storge yard', 'Container', 'Oil
Tank', 'Berth'])
    :param line_color: init by null
    :param fill_color: init by null
    :param Cordlist: like [ [ [ 114.062252545881549, 33.884823424589996 ],
[ 114.063734485974663, 33.885029393359083 ],..........
    :return:
```

Shpfile

```
    invert the Shapefile to labelme_Json
    Flow I  : Read Tif & Shapefile -> Dataset & DataSource
    Flow II : Init Null labelme_json dict
    Flow III: Get Cords(x,y) & Feature and invert longtitude,latitude to
image cols,rows
```

```
    Flow IV : Append image Cords list to dict
    Flow V  : Output Labelme Json
    Flow VI : Use System Command labelme_json_to_dataset / labelme2voc.py
            # It generates:
            #   - data_dataset_voc/JPEGImages
            #   - data_dataset_voc/SegmentationClass
            #   - data_dataset_voc/SegmentationClassVisualization
    :param shp_path:
    :param tif_path:
    :param Labelme_Json_path: Output Json&jpg path
    :param invert2Dataset: invert labelme json to Dataset
    :param  TYPE like (['Storge yard', 'Container', 'Oil Tank', 'Berth'])
    :return:
```