**text-detection-ctpn**

<https://github.com/eragonruan/text-detection-ctpn>

**ctpn環境建置(For GPU Demo)**

1. **Git Clone檔案**
   1. **註冊SSH**

~$ ssh-keygen -t rsa -b 4096 -C "yourGithub@gmail.com"

* 1. **查看生成的SSH，記得加到自己Github SSH Keys上**

~$ cat ~/.ssh/id\_rsa.pub

Settings > SSH and GPG keys > New SSH key

* 1. **Clone repo**

~$ git clone [git@github.com:eragonruan/text-detection-ctpn.git --depth=1](mailto:git@github.com:eragonruan/text-detection-ctpn.git%20--depth=1)

1. **安裝環境、套件**
   1. **下載、安裝 anaconda/miniconda(擇一)**

~/downloads$ wget <https://repo.anaconda.com/archive/Anaconda3-2024.02-1-Linux-x86_64.sh>

~/downloads$ wget https://repo.anaconda.com/miniconda/Miniconda3-latest-Linux-x86\_64.sh

bash Anaconda3-2024.02-1-Linux-x86\_64.sh

bash Miniconda3-latest-Linux-x86\_64.sh

安裝完成後，會提示需重新啟動終端，接著就前方就會有base字樣

(base)~$

* 1. **手動安裝舊版本套件(opencv-python、tensorflow)**

手動安裝之前，請記得先創建一個虛擬環境，例如py3.6

(base)~/downloads$ conda create -n py3.6 python=3.6

(base)~/downloads$ conda activate py3.6

前方的base就會切換道新創的虛擬環境py3.6，如下

(py3.6)~/downloads$

若想退出此環境 : conda deactivate

接著才安裝opencv-python、tensorflow

(py3.6)~/downloads$ wget https://files.pythonhosted.org/packages/50/f9/5c454f0f52788a913979877e6ed9b2454a9c7676581a0ee3a2d81db784a6/opencv\_python-3.4.0.12-cp36-cp36m-manylinux1\_x86\_64.whl

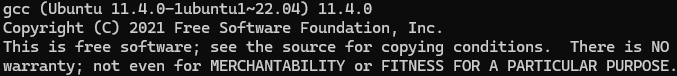
(py3.6)~/downloads$ wget https://files.pythonhosted.org/packages/b8/16/14fbe414f2b3c39c5d01521e38e7aaec384f8fbf26eec021d247345260c3/tensorflow\_gpu-1.3.0-cp36-cp36m-manylinux1\_x86\_64.whl

(py3.6)~/downloads$ python3 -m pip install opencv\_python-3.4.0.12-cp36-cp36m-manylinux1\_x86\_64.whl

(py3.6)~/downloads$ python3 -m pip install tensorflow\_gpu-1.3.0-cp36-cp36m-manylinux1\_x86\_64.whl

* 1. **檢查/安裝gcc編碼器**

(py3.6)~/text-detection-ctpn$ gcc –version



有的話跳過，沒有的話請先詢問是否可安裝

(py3.6)~/text-detection-ctpn$ sudo apt-get install build-essential

* 1. **檢查/安裝nvidia-cudatool-kit**

(py3.6)~/text-detection-ctpn$ nvcc --version

**一張含有 文字, 字型, 螢幕擷取畫面 的圖片

自動產生的描述**

有的話跳過，沒有的話請先詢問是否可安裝

(py3.6)~/text-detection-ctpn$ sudo apt-get install nvidia-cuda-toolkit

* 1. **從第三方來源安裝舊版cuda、cudnn**

因為conda裡沒有cuda 8、cudnn 6這種舊版本，

所以需要增加第三方來源協助安裝，

(py3.6)~/text-detection-ctpn$ conda config --append channels <https://mirrors.tuna.tsinghua.edu.cn/anaconda/pkgs/free/>

(py3.6)~/text-detection-ctpn$ conda search cudatoolkit

(py3.6)~/text-detection-ctpn$ conda search cudnn

尋找支援tensorflow版本的cuda、cudnn，並安裝他

(py3.6)~/text-detection-ctpn$ conda install cudatoolkit=8.0 cudnn=6.0.21

* 1. **安裝需要用到的所有套件**

(py3.6)~/text-detection-ctpn$ python3 -m pip install -r requirements.txt

1. **環境建置與Demo**
   1. **在build資料夾中生成兩個.so檔**

(py3.6)~/text-detection-ctpn$ cd ./lib/utils/

(py3.6)~/text-detection-ctpn/lib/utils$ chmod +x make.sh

(py3.6)~/text-detection-ctpn/lib/utils$ ./make.sh

**檔名應該是對應py版本:**

bbox.cpython-36m-x86\_64-linux-gnu.so、

cython\_nms.cpython-36m-x86\_64-linux-gnu.so

* 1. **測試numpy且匯出numpy include檔案位置**

(py3.6)~$ python

Python 3.6.13 |Anaconda, Inc.| (default, Jun 4 2021, 14:25:59)

[GCC 7.5.0] on linux

Type "help", "copyright", "credits" or "license" for more information.

>>> import numpy as np

>>> print(np.get\_include())

/home/g0983230881/miniconda3/envs/tf1.3\_py3.6/lib/python3.6/site-packages/numpy/core/include

**Ctrl+D退出**

* 1. **設置環境變量CFLAGS以指定numpy的include路徑**

(py3.6)~/text-detection-ctpn$ export CFLAGS=-I/home/g0983230881/miniconda3/envs/tf1.3\_py3.6/lib/python3.6/site-packages/numpy/core/include

(以我的例子是在這個位置)

* 1. **取消註解**

$ cd~/text-detection-ctpn/lib/rpn\_msr

proposal\_layer\_tf.py 第45行取消註解，如下

#cfg\_key=cfg\_key.decode('ascii')

* cfg\_key=cfg\_key.decode('ascii')
  1. **下載並配置release檔案**

從repo下載 ctpn.pb、checkpoints.zip

ctpn.pb 放在 ~/text-detection-ctpn/data/

checkpoints.zip 解壓縮後，

在 ~/text-detection-ctpn/ 新建 checkpoints資料夾，

再把那四個檔案放入其中

* 1. **Demo**

(py3.6)~/text-detection-ctpn/lib/utils$ python setup.py build

(py3.6)~/text-detection-ctpn$ python ./ctpn/demo.py

Demo到這邊就可以了。

1. **配置訓練資料集**
   1. **下載並配置檔案**

下載作者提供的VGG\_imagenet.npy、VOCdevkit.zip

把VGG\_imagenet.npy放到 data/pretrain/

解壓縮VOCdevkit.zip後，資料夾放在lib/prepare\_training\_data/

* 1. **撰寫xml轉txt檔案**

因為資料集裡split\_label.py需要匯入.txt，但作者提供的是.xml，所以要寫一支xml2txt.py的程式，如下:

-----------------------

import os

import xml.etree.ElementTree as ET

def convert\_xml2txt(xml\_folder, txt\_folder):

if not os.path.exists(txt\_folder):

os.makedirs(txt\_folder)

for xml\_file in os.listdir(xml\_folder):

if not xml\_file.endswith('.xml'):

continue

xml\_path = os.path.join(xml\_folder, xml\_file)

tree = ET.parse(xml\_path)

root = tree.getroot()

txt\_lines = []

for obj in root.findall('object'):

bndbox = obj.find('bndbox')

xmin = bndbox.find('xmin').text

ymin = bndbox.find('ymin').text

xmax = bndbox.find('xmax').text

ymax = bndbox.find('ymax').text

# 根據split\_label.py生成座標string

coords = f"{xmin},{ymin},{xmax},{ymin},{xmax},{ymax},{xmin},{ymax}"

txt\_lines.append(coords)

txt\_filename = 'gt\_' + os.path.splitext(xml\_file)[0] + '.txt'

txt\_path = os.path.join(txt\_folder, txt\_filename)

with open(txt\_path, 'w') as txt\_file:

for line in txt\_lines:

txt\_file.write(line + '\n')

directory = '/home/g0983230881/CTPN/text-detection-ctpn-master'

xml\_folder = directory + '/lib/prepare\_training\_data/xmlFolder' # XML Folder

txt\_folder = directory + '/lib/prepare\_training\_data/txtFolder' # store txt\_folder

convert\_xml2txt(xml\_folder, txt\_folder)

-----------------------

程式中的xml\_folder、txt\_folder記得要改路徑，指到對應的資料夾

接著也一起修改splite\_label.py裡path、gt\_path的路徑，

-----------------------

directory = '/home/g0983230881/CTPN/text-detection-ctpn-master'

path = directory + '/lib/prepare\_training\_data/imageFolder'

gt\_path = directory + '/lib/prepare\_training\_data/txtFolder'

-----------------------

* 1. **生成prepared data**

接著產生在當前資料夾prepared data

(py3.6)~/text-detection-ctpn$ cd lib/prepare\_training\_data

(py3.6)~/text-detection-ctpn/lib/ prepare\_training\_data$ python xml2txt.py

把.xml轉成.txt再按照repo上的步驟走

(py3.6)~/text-detection-ctpn/lib/ prepare\_training\_data$ python split\_label.py

* 1. **轉換prepared training data into VOC.format**

(py3.6)~/text-detection-ctpn/lib/ prepare\_training\_data$ python ToVoc.py

執行完後會產生一個TEXTVOC資料夾，把它移到data/目錄下，

並到data/執行

cd ~/text-detection-ctpn-master/data/

(py3.6)~/text-detection-ctpn/data$ ln -s TEXTVOC VOCdevkit2007

生成一個VOCdevkit2007檔案，連結到 ~/data資料夾底下

* 1. **修改檔案 /lib/datasets/imdb.py**

這部分有兩個地方要改，分別是第3、81行

origin: import PIL

after: from PIL import Image

-----------------------

origin: PIL.Image.open(self.image\_path\_at(i)).size[0]

after: Image.open(self.image\_path\_at(i)).size[0]

* 1. **訓練資料集**

cd ~/text-detection-ctpn-master/

python ./ctpn/train\_net.py

**遇到問題與解決方法**

1. **unable to execute 'gcc': No such file or directory**

**error: command 'gcc' failed with exit status 1**

若遇到此問題，檢查是否有安裝gcc，安裝前請詢問是否可安裝

$ gcc -v

$ sudo apt-get install build-essential

1. **error: no commands supplied**

若遇到此問題，請檢查是否缺少或打錯關鍵字。

1. **IndentationError: expected an indented block**

若遇到這問題，請檢查程式是否正確縮排。

1. **Traceback (most recent call last):**

**File "./ctpn/demo.py", line 13, in <module>**

**from lib.networks.factory import get\_network**

**ModuleNotFoundError: No module named 'lib.networks'**

若遇到這問題，手動指定PYTHONPATH到自己專案位置。

$ export PYTHONPATH='/home/g0983230881/text-detection-ctpn/'

$ cd ~/text-detection-ctpn/ 在這個路徑下執行demo.py

$ python ./ctpn/demo.py

Solution\_link: [https://github.com/eragonruan/text-detection-ctpn/issues/26#issuecomment-342794327](https://github.com/eragonruan/text-detection-ctpn/issues/26%23issuecomment-342794327)

1. **Traceback (most recent call last):**

**File "./ctpn/demo.py", line 93, in <module>**

**raise 'Check your pretrained {:s}'.format(ckpt.model\_checkpoint\_path)**

**AttributeError: 'NoneType' object has no attribute 'model\_checkpoint\_path'**

若遇到這個問題，

$ cd ~/text-detection-ctpn/

從release下載的checkpoints.zip解壓縮後，

把checkpoints資料夾放在 ~/text-detection-ctpn/，檔案路徑如下

一張含有 文字, 螢幕擷取畫面, 字型 的圖片

自動產生的描述

若還是不行，就用絕對路徑指定該資料夾，

(以checkpoints資料夾放在 ~/text-detection-ctpn/ctpn為例)

$ cd ~/text-detection-ctpn/ctpn/ 編輯demo.py第88行，如下

origin :

ckpt = tf.train.get\_checkpoint\_state(cfg.TEST.checkpoints\_path)

after :

checkpoint\_path = '/home/g0983230881/text-detection-ctpn/ctpn/checkpoints'

ckpt = tf.train.get\_checkpoint\_state(checkpoint\_path)

1. **TabError: inconsistent use of tabs and spaces in indentation**

請確認程式內是否有Tab鍵與空白鍵混用，請統一用空白鍵或Tab縮排

1. **UnknownError (see above for traceback): KeyError: b'TEST'**

**[[Node: rois/PyFunc = PyFunc[Tin=[DT\_FLOAT, DT\_FLOAT, DT\_FLOAT, DT\_STRING, DT\_INT32, DT\_INT32], Tout=[DT\_FLOAT, DT\_FLOAT], token="pyfunc\_0", \_device="/job:localhost/replica:0/task:0/cpu:0"](Reshape\_2, rpn\_bbox\_pred/Reshape\_1, \_arg\_Placeholder\_1\_0\_1, rois/PyFunc/input\_3, rois/PyFunc/input\_4, rois/PyFunc/input\_5)]]**

若遇到這個問題，

$ cd ~/text-detection-ctpn/lib/rpn\_msr/

proposal\_layer\_tf.py 第45行取消註解，如下

# cfg\_key=cfg\_key.decode('ascii')

* cfg\_key=cfg\_key.decode('ascii')

$ cd ~/text-detection-ctpn

python ./ctpn/demo.py

Solution\_link: [https://github.com/eragonruan/text-detection-ctpn/issues/208#issuecomment-450314488](https://github.com/eragonruan/text-detection-ctpn/issues/208%23issuecomment-450314488)

1. **ValueError: Object arrays cannot be loaded when allow\_pickle=False**

**Traceback (most recent call last):**

**File "/home/g0983230881/text-detection-ctpn/lib/fast\_rcnn/train.py", line 122, in train\_model**

**self.net.load(self.pretrained\_model, sess, True)**

**File "/home/g0983230881/text-detection-ctpn/lib/networks/network.py", line 41, in load**

**data\_dict = np.load(data\_path,encoding='latin1').item()**

**File "/home/g0983230881/miniconda3/envs/tf\_1.15py3.6/lib/python3.6/site-packages/numpy/lib/npyio.py", line 440, in load**

**pickle\_kwargs=pickle\_kwargs)**

**File "/home/g0983230881/miniconda3/envs/tf\_1.15py3.6/lib/python3.6/site-packages/numpy/lib/format.py", line 727, in read\_array**

**raise ValueError("Object arrays cannot be loaded when "**

若遇到這個問題，到 /lib/networks/network.py，

新增參數: allow\_pickle=Ture

origin:

data\_dict = np.load(data\_path,encoding='latin1').item()

after:

data\_dict = np.load(data\_path,encoding='latin1', allow\_pickle=Ture).item()