Heatmap, Attention- time/count/average
Documentation for Linux

Documentation Overview :

1. **Installations and env setup**
2. **Structure Overview with example**
3. **Project Configurations**
4. **Project Executions**

Note: If something through the doc mentioned within '< >', then it should be replaced by your own prefered data which you think correct for your instance.
Eg: ***/home/<user>/Desktop*** *=== **/home/hemanth/Desktop*** *# in my case user name is hemanth*

1. Installations and env setup:

[warning] : Go through **'Structure Overview with example'** section and where in **How get to know paths..** try to find better idea where to place 'heatmap_attspan' folder in the right directory**('base_project_path')** and then do below installations.

Three steps:

(a) **Anaconda installation**
(b) **Pytorch env setup**
(c) **Tensorflow env setup** with TF Object-Detection API v0.2

(a) Anaconda installation:

[instruction] : open up terminal and run below commands..
$ sudo apt-get install curl
$ curl -O https://repo.anaconda.com/archive/Anaconda3-2019.03-Linux-x86_64.sh
[instruction] : while installing by the below command.. please type 'y' for everything and accept the anaconda3 path as '/home/<user>/anaconda3'. Hence agree and accept for every thing.
$ bash Anaconda3-2019.03-Linux-x86_64.sh
$ source ~/.bashrc
NOTE : after this you can notice **'(base)' anaconda env** indicator before in the command line.

[Verification]:
$ *conda –version* ([outputs] : **conda** <**version number**> like 'conda 4.8.5')

(b) Pytorch env setup:

[instruction] : open up terminal(anaconda installed and in base env) and run below commands..
$ conda create --name pytor python=3.7
$ conda activate pytor
NOTE : after this you can notice **'(pytor)' anaconda env** indicator before in the command line.

[instruction] : To install pytorch with GPU support via conda and other packages, run below commands..
$ conda install pytorch torchvision cudatoolkit=10.2 -c pytorch
$ conda install -c conda-forge tqdm opencv matplotlib
$ conda install requests
$ conda install -c anaconda seaborn
$ conda deactivate
NOTE : after this you can again notice **'(base)' anaconda env** indicator before in the command line.

[Verification]:
$ conda activate pytor
$ python
NOTE : enters into **python scripting** and through which check pytorch installed by,
>>>import torch ([outputs] : if **pytorch installed** // not installed properly === **imports torch and outputs nothing** // throws error as "no module name torch")
NOTE : check pytorch version by,
>>>print(torch.__version__) ([outputs] : <**version number**> like ' 1.6.0 ')
NOTE : check pytorch has GPU support by,
>>>print(torch.cuda.is_available()) ([outputs] : if **GPU** // no GPU === **True** // False)

>>>exit()
NOTE : returns to **shell scripting**
$ conda deactivate

(c) Tensorflow env setup:

- **tensorflow-2 GPU installation**
- **tensorflow-2 Object Detection API v0.2 installtion**

Tensorflow-2 GPU installation:

[instruction] : open up terminal(anaconda installed and in base env) and run below commands..
$ conda create --name tf-gpu python=3.7
$ conda activate tf-gpu
NOTE : after this you can notice **'(tf-gpu)' anaconda env** indicator before in the command line.
[instruction] : To install tensorflow-2.2+ with GPU support via conda and other packages, run below commands..
$ conda install -c anaconda tensorflow-gpu
$ conda install -c conda-forge opencv matplotlib tqdm

[Verification]:
$ conda activate tf-gpu
$ python
NOTE : check tensorflow installed by,
>>>import tensorflow as tf ([outputs] : if **tensorflow installed** // not installed properly === **imports tensorflow and outputs nothing** // throws error as "no module name tensorflow")
NOTE : check pytorch version by,
>>>print(tf.__version__) ([outputs] : <**version number**> like ' 2.2.0 ')
NOTE : check pytorch has GPU support by,
>>>print(tf.test.is_gpu_available()) ([outputs] : if **GPU** // no GPU === **True** // False)

>>>exit()
NOTE : returns to **shell scripting**
$ conda deactivate

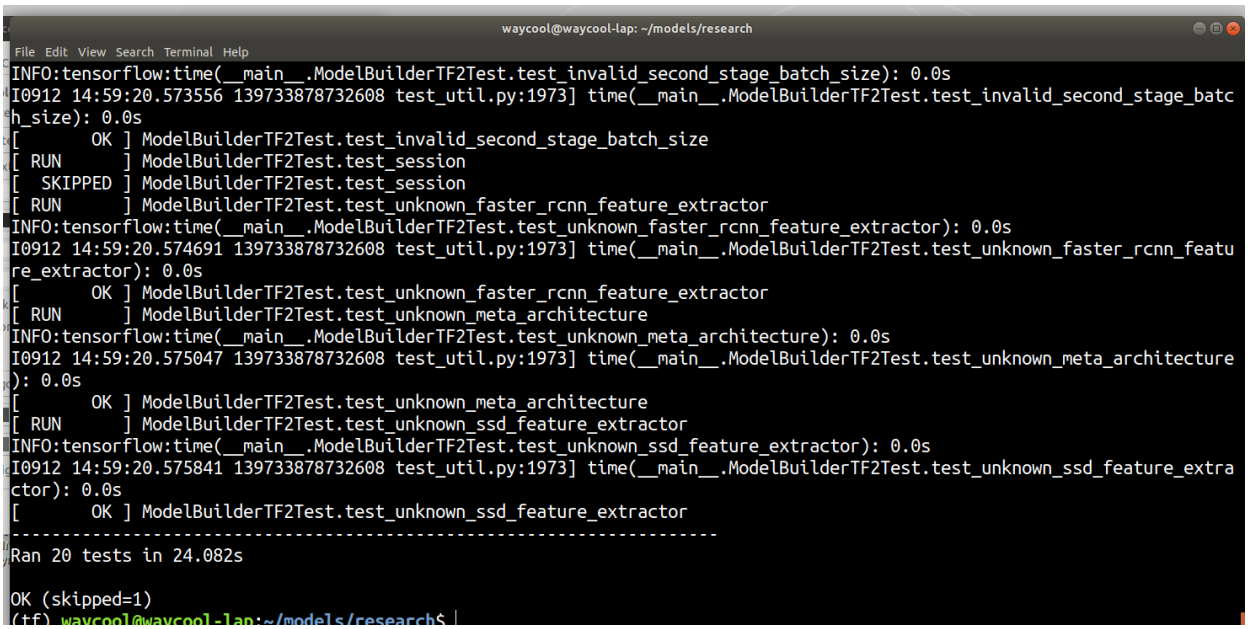
Tensorflow-2 Object Detection API v0.2 installtion:

*[instruction] : To install tensorflow Object Detection API v-0.2, navigate(by **cd <folder_name>**) to directory holding our **project folder(heatmap_attspan)** and then run.*
$ cd heatmap_attspan/EDET7/models/research
NOTE : while running below command.. check there is no error in building wheels or any other things.. So there should not be any error. If any shows up try resolve it and then run again.
$ python -m pip install --use-feature=2020-resolver .

[Verification]:

$ conda activate tf-gpu
$ python object_detection/builders/model_builder_tf2_test.py

NOTE : Installation is done properly when you see similarly below.



---

2) Structure Overview with example:

[warning] : The names of three folders(heatmap_attspan, heatmap_attspan_video, heatmap_attspan_res**) are fixed.. if names were changed, project wont work and while running throws error ‘ no such file or directory’**

Three folders

- **project folder: (heatmap_attspan)** #[just need to be placed mannually], which I shared and where code and other files were placed...
- **video folder: (heatmap_attspan_video)** #[need to create and place mannually], where videos are placed under date-wise sub-directory.

- **result folder: (heatmap_attspan_res)** #[no need to do anything, everything will be made by system itself] where results are placed as a date and video name wise sub-directories.

HOW GET TO KNOW PATHS?

There folders path can be placed anywhere you prefered..

For instance, just in my case,

project folder(heatmap_attspan) in Documents, i.e. /*home*/hemanth/Documents/heatmap_attspan/
=== i.e /<*base_project_path*/heatmap_attspan/

Video folder(heatmap_attspan_video) in Videos, i.e. /*home*/hemanth/Videos/heatmap_attspan_video/
=== i.e /<*base_input_path*/heatmap_attspan_video/

Result folder(heatmap_attspan_res) in Desktop, i.e. /*home*/hemanth/Desktop/heatmap_attspan_res/
=== i.e /<*base_input_path*/heatmap_attspan_res/

Hence,

- base_project_path='/home/<user>/Documents/'
- base_input_path='/home/<user>/Videos/'
- base_output_path='/home/<user>/Desktop/'

Note : The above path is just for my case.. you can set you own path which you prefered.. even all three folders can be set in Documents or any other one directory.. only caution is need to set where each of these three folders are present.

HOW TO ORGANIZE THESE 3 FOLDERS? *To make successful execution..*

1) project folder(**heatmap_attspan**) :

- must copy paste the whole **heatmap_attspan** folder in <base_project_path>. Thats all..

2) video folder(**heatmap_attspan_res**) :

- must create folder named 'heatmap_attspan_video' in <base_input_path>
- in that created folder, if want to upload video file that need to processed.. then,
- create a folder named with day of the video taken and in that upload the video file with name as time duration.

So, ensure video uploading structure similarly like this below..

```
/<base_input_path>/heatmap_attspan_video/
                                ___/23042020/                   # date format (ddmmyyyy)
                                      ___/06000700.mp4          # 24 time format (hhmmhhmm)
                                      ___/13151500.mp4          # 24 time format (hhmmhhmm)
                                      ___/06000700.mp4
                                ___/24042020/                   # date format (ddmmyyyy)
                                      ___/08001000.mp4          # 24 time format (hhmmhhmm)
                                      ___/.................mp4   # 24 time format (hhmmhhmm)
                                      ___/.................mp4
                                      ___/.................mp4
                                ___/25042020/
                                      ___/.....mp4
                                      ___/.....mp4
                                      ....
```

3) result folder(**heatmap_attspan_res**) :

- nothing to do with by user.. will automatically created by system itself in <base_output_path>..

- Just go and find the results by its given date and time named sub-folders recursively..

3) Project Configurations:

[instruction] : To configure, mannually edit the file **'~/heatmap_attspan/configure.py'** with **TextEditor**..

NOTE : You can see **four parameters** need to be set..

########## user's configuration part begins #########

1. base_project_path='' # directory where base project folder named = **'heatmap_attspan'** placed..
2. base_input_path='' # directory where base Video folder named = **'heatmap_attspan_video'** placed..
3. base_output_path='' # directory where base results folder named = **'heatmap_attspan_res'** will be placed..
4. store_id='' #Set store id like **store_id='1001'**

########## user's configuration part ends #########

[instruction] : Set all those four parameters.. and save file.. thats all configuration over..

[warning]: edit only within quotations**(eg: base_input_path='** to **base_X_path='/home/<user>/Desktop/** like that**'),** and don't change any other things(like leaving space and lines inbetween and etc..).

---

4) Project Executions:

- **Mannually process** today's and past two day's videos if any there
- **Schedule Automatic process** today's and past two day's videos if any there
- **Process a Specific Video file**

Mannually process:

Mannually execute on terminal to check video for today and previous two days from **/<base_input_path>/heatmap_attspan_video/ ,** then process and give results in **/<base_output_path>/heatmap_attspan_res/**...
run below commands,

[instruction] : open up terminal and navigate into '**heatmap_attspan**' by,
$ cd <base_project_path>/heatmap_attspan/
$ bash run_scheduler.sh

Schedule Automatic process:

Automatically execute by scheduler to check video for today and previous two days from **/<base_input_path>/heatmap_attspan_video/ ,** then process and give results in **/<base_output_path>/heatmap_attspan_res/**...

use '**run_scheduler.sh**'

[instruction] : For Automtic execution by setting scheduler, use '**run_scheduler.sh**' file in **'~/heatmap_attspan/'** folder.

Process a Specific Video file:

Mannually execute on terminal to check video for today and previous two days from **/<base_input_path>/heatmap_attspan_video/ ,** then process and give results in **/<base_output_path>/ heatmap_attspan_res/**...

NOTE: only one video file can be executable.
2-steps:

- **edit video.txt file**
- **run master file**

***e*dit video.txt file:**

- *open video.txt file in TextEditor.. and write the video file path and save. Thats all*

[warning]: *video.txt must contain* ***only one line*** *and in that should contain* ***only one video file path*** *like a below path format.*
i.e heatmap_attspan_video/21102020/18001900.mp4
So format changes only in ***/date/time.mp4*** *values..*

run master file:

- run the below command to process the video file that mention in the video.txt file.
- $ python master.py
