Suposant que estic a $HOME al començar.

1. Descarregar el repositori i accedir al directori

git clone <https://github.com/JDAI-CV/fast-reid.git>

cd fast-reid

1. Preparar l’entorn de python3 (no funciona amb torch>=2.0.0)

Aquest FastREID no es compatible amb ultralytics (YOLOv8)

module load cuda/11.3 cudnn/8.1 gcc/7 cmake/3.23.0  
  
virtualenv --python=python3.6 ~/venv/fastreid  
source ~/venv/fastreid/bin/activate  
  
pip install -U torch==1.10.1+cu113 torchvision==0.11.2+cu113 -f https://download.pytorch.org/whl/cu113/torch\_stable.html  
pip install yacs  
pip install tensorboard  
pip install gdown  
pip3 install -U scikit-learn  
pip install termcolor  
pip install tabulate  
pip install swig  
pip install faiss-gpu  
pip install Cython  
  
cd ~/fast-reid/  
srun --gres=gpu:1 --mem 16G --time=10:00:00 pip install --no-cache-dir -r docs/requirements.txt  
  
cd fastreid/evaluation/rank\_cylib; make all

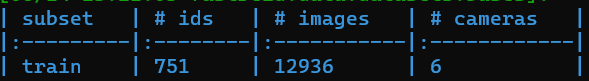
1. Market1501 installation test

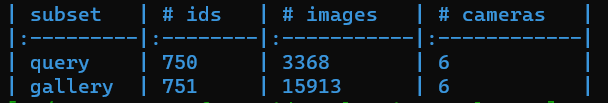
Descarregar <https://drive.google.com/file/d/0B8-rUzbwVRk0c054eEozWG9COHM/view>

Descomprimir (unzip) en ~/fast-reid/datasets/

srun -c 8 --gres=gpu:4 --mem 32G --time=10:00:00 python tools/train\_net.py --config-file ./configs/Market1501/bagtricks\_R50.yml --num-gpus 4

Tardar 50 minuts aproximadament, es pot ometre si les primeres 31 èpoques funcionen.



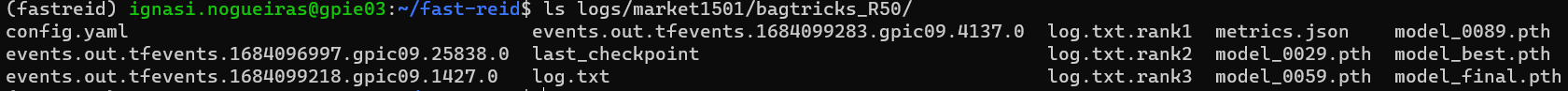


Els conjunts query i gallery (carpetes query i bounding\_box\_test) serveixen per avaluar:

* Si es buiden, l’entrenament falla.
* Si la id de la càmera és la mateixa, també falla (en realitat es un problema de mateixa id en càmera i identitat); al generar els conjunts, forçar càmera 1 o càmera 2 tota la carpeta:

{identitat}\_c{camara}s{X}\_{imgref}.jpg

* I si no hi ah suficients imatges, també falla.



1. Per crear un dataset equivalent automàticament a partir d’un vídeo i anotacions, he fet un programa (el programa descarta identitats amb pocs representants i oclusions):

python3 video\_to\_apparences.py DATA/output\_4\_gt.mp4 DATA/output\_4\_gt.txt

El resultat es un arxiu Market-1501-v15.09.15.zip que conté una carpeta Market-1501-v15.09.15 per sobreescriure-la en ~/fast-reid/datasets/

Es la millor solució en termes de cost-utilitat en un TFM.

srun -c 1 --mem 32G --time=10:00:00 python video\_to\_apparences.py DATA/output\_4\_gt.mp4 DATA/output\_4\_gt.txt

1. Sobreescriure el nou “Market-1501-v15.09.15” en ~/fast-reid/datasets/
2. Preferim png a jpg (per evitar pèrdues):

Modifiquem la linea 73 de fastreid/data/datasets/market1501.py de:

        img\_paths = glob.glob(osp.join(dir\_path, '\*.jpg'))

a

        img\_paths = glob.glob(osp.join(dir\_path, '\*.png'))

Modificar les mides del input en:

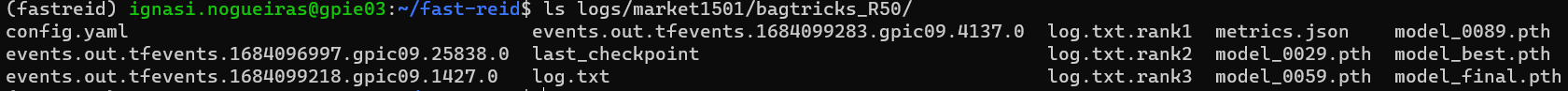
configs/Base-bagtricks.yml

Configuració addicional de l’entrenament, I/O, etc editable en base a arxus YAML.

1. Executar la comanda d’entrenament anterior:

srun -c 8 --gres=gpu:4 --mem 32G --time=10:00:00 python tools/train\_net.py --config-file ./configs/Market1501/bagtricks\_R50.yml --num-gpus 4

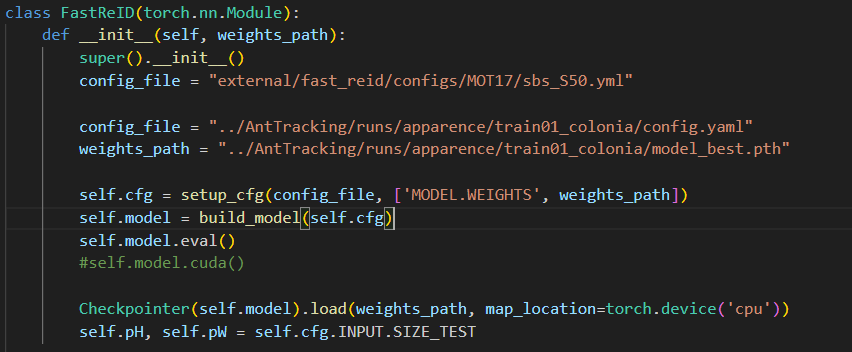
Pesos obtinguts (model\_final.pth) i logs:



1. El model resultant es pot usar amb el codi d’adaptació de Deep OC-SORT i els pesos obtinguts.

L’evolució de l’entrenament es guarda a logs/market1501/bagtricks\_R50/metrics.json (tant pel conjunt traint com pel conjunt test).

1. Per utilitzar els pesos, canviar l’arxiu de configuració i de pesos “hardwired” a “external/adators/fastreid\_adaptor.py”:



1. Els resultats estan en format json, tinc una notebook (to\_mot.ipynb) per passar-ho al format MOTChallenge