Multi-Video Processing

System used for the project:

1. OS - Windows 10
2. GPU - Nvidia RTX 2070 s
3. Cuda - CUDA 10.1

Installation for running demo on Windows 10:

1.Install GPU Drivers
https://nvidia.custhelp.com/app/answers/detail/a_id/2900/~/installing-nvidia-display-drivers-under-windows-7%2C-windows-8%2C-or-windows-10

2.Install Cuda Toolkit 10.1
https://developer.nvidia.com/cuda-10.1-download-archive-base?target_os=Windows&target_arch=x86_64&target_version=10&target_type=exelocal

3.Install Cudnn
https://developer.nvidia.com/compute/machine-learning/cudnn/secure/8.0.5/10.1_20201106/cudnn-10.1-windows10-x64-v8.0.5.39.zip

4.cd into project directory

5.pip install -r requirements.txt

6.Install pytorch- pip3 install torch==1.7.1+cu101 torchvision==0.8.1+cu101 -f https://download.pytorch.org/whl/torch_stable.html

Installation for running demo on Linux:

1.Install GPU Drivers
https://www.cyberciti.biz/faq/ubuntu-linux-install-nvidia-driver-latest-proprietary-driver/

2.Install Cuda Toolkit 10.1
https://docs.nvidia.com/cuda/cuda-installation-guide-linux/index.html

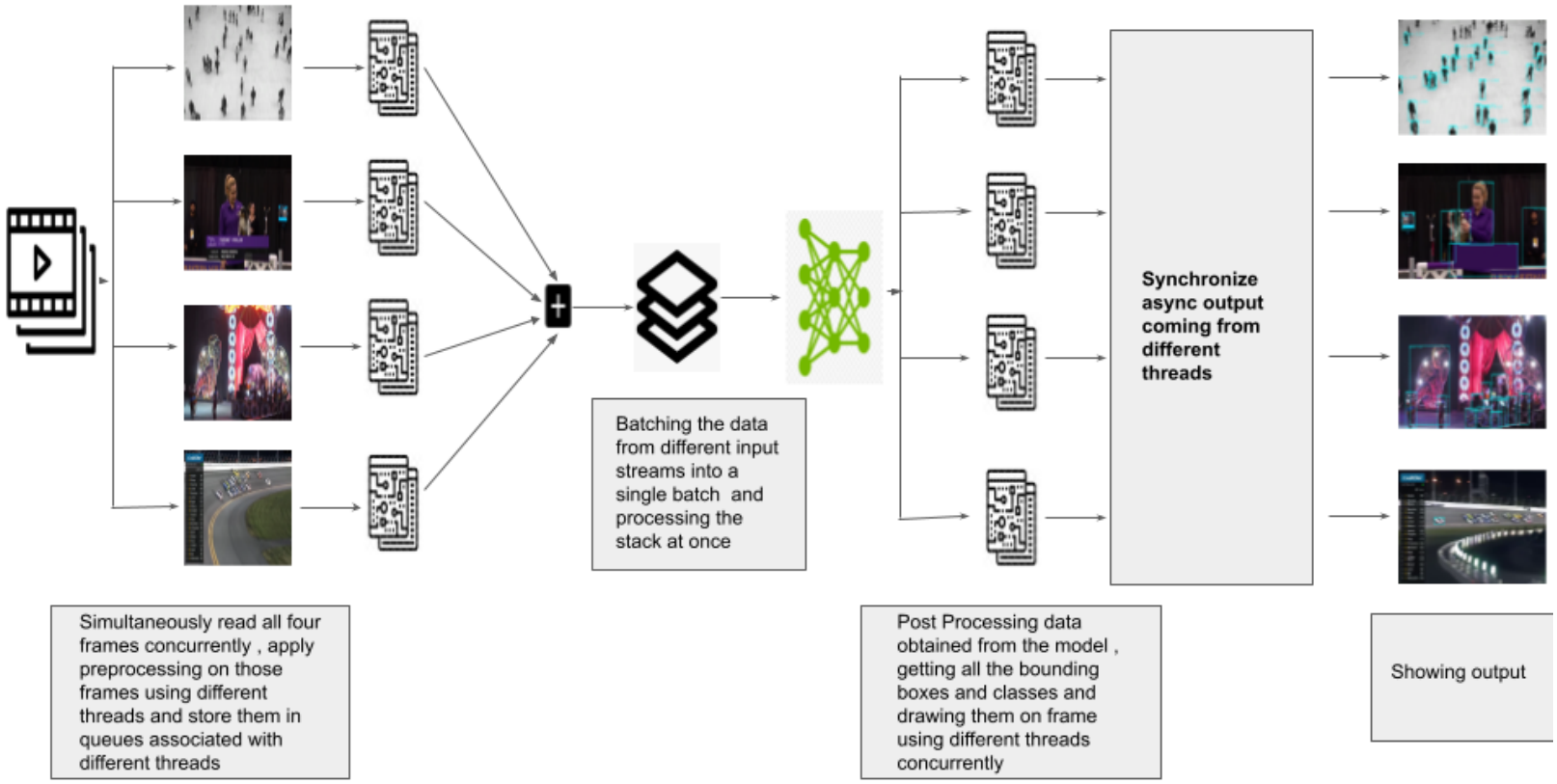
3.Install Cudnn
https://developer.nvidia.com/rdp/cudnn-archive

4.cd into project directory

5.pip3 install -r requirements.txt

6.Install pytorch - pip3 install torch==1.7.1+cu101 torchvision==0.8.1+cu101 -f https://download.pytorch.org/whl/torch_stable.html

PIPELINE AND APPROACH FOR THE PROCESS:



Pipeline for video processing

Some insights:

1.Run python “demo(4 video).py” for running the demo for 4 videos simultaneously, the demo has Inference Request of 1 and Batch of 4 images from different input streams .(Performed best with average of 8 FPS)

2.Run python “demo(1 video).py” for running the demo for 1 video with Inference Request of 1 and Batch of 1 image (Performed with only upto 7 FPS) , you can run this just for comparison with the main pipeline mentioned above.

3.Run python “demo(IR=2).py” for running the demo with 4 videos simultaneously. The demo has Inference Request of 2 and a batch of 2 images from different input streams at once.(Performed poorly with only average of upto 6 FPS)

NOTE: All four videos should be in the project directory

DIFFICULTIES FACED DURING THE PROJECT:

1.The model is not real time as it did not perform well even on a single video.

Solution:The model can be optimized for faster inference with tools like Openvino , TensorRT etc , this project is just an example to show how we can optimize our solutions for end to end video analytics , for future use we can use IR models as well for faster inference.

2.There was a problem running 2 models at the same time in parallel , I tested the code with Batch=2, Inference Request=2 and the results were not better than Batch =4 , Inference Request=1.

Solution:Running separate inferences in separate docker containers asynchronously and then receiving output per frame and synchronizing them all at once could be a valid choice , but for the current application with 4 video streams the IR=1 is sufficient.

3.Linux based OS could have made the project a little easier as i was not able to run a deepstream version of the project or dockerize the inference as mentioned in problem no. 2.

Reference Repository:

1.https://github.com/zylo117/Yet-Another-EfficientDet-Pytorch