Docker image settings :

Dockerfile

```
FROM nvidia/cuda:12.0.0-runtime-ubuntu22.04

WORKDIR /yolov8

ADD https://ultralytics.com/assets/Arial.ttf /root/.config/Ultralytics/

RUN apt-get update \
   && apt-get install --no-install-recommends -y  \
      libgl1-mesa-glx libglib2.0-0 python3 python3-pip \
   && pip3 install ultralytics \
   && rm -rf /var/lib/apt/lists/*

ENTRYPOINT ["yolo"]
```

build the image

docker build -t yolov8 .

make directory where we will put our input to the model

```
mkdir -p ~/yolov8/inputs
```

download the needed video and put it in that directory exemple text.mp4

to test it for the first time ::

```
docker run -it --rm \
    -v ~/yolov8:/yolov8 \
    yolov8 detect predict save model=yolov8s.pt source=inputs/test.mp4
```

create a small container to test

```
docker run -it --rm \
    --cpus="2.0" \
    --memory="2g" \
    -v ~/yolov8:/yolov8 \
    yolov8 detect predict save model=yolov8s.pt
source=inputs/test.mp4
```

create a large container to test

```
docker run -it --rm \
    --cpus="5.0" \
    --memory="5g" \
    -v ~/yolov8:/yolov8 \
    yolov8 detect predict save model=yolov8s.pt
source=inputs/test.mp4
```

the small container ::

```
docker run -d --rm \
    --cpus="1.0" \
    --memory="1g" \
    -v ~/yolov8:/yolov8 \
    yolov8 detect predict save model=yolov8s.pt
source=inputs/test.mp4
```

—-comparaison
//result with the large container
Speed: 4.0ms preprocess, 290.7ms inference, 1.7ms postprocess per image at shape (1, 3, 384, 640)

//result with medium container

/// result if i test with one small container in this case
Speed: 9.6ms preprocess, 1238.1ms inference, 6.2ms postprocess per image at shape (1, 3, 384, 640)

→ in the case of 5 container
idea::

- use 5 small container , give each one a ⅕ partr of the video and at the end regroupe all of videos
- replication with k8s , prob how to do load balance in this case ?