# DeepStream Inference Setup & Guide

## 1. Folder Structure

/workspace

├── deepstream-app/

│ ├── configs/

│ │ ├── deepstream\_app\_config.txt

│ │ ├── config\_infer\_primary.txt

│ ├── custom\_parser/

│ │ ├── nvdsparsebbox\_mmyolo.cpp

│ │ └── libnvdsinfer\_custom\_impl\_mmyolo.so

│ └── ...

├── training-output/

│ └── end2end\_fp16.engine

├── labels.txt

# Open existing container

docker start -ai c4f933c8e4fb

**Open in another terminal**

docker exec -it -u root c4f933c8e4fb bash

Fresh container start

xhost +local:root  
docker run -it --name deepstream\_container \

--gpus all \

-v /home/prakash/deepstream6.4/:/workspace/ \

-e DISPLAY=$DISPLAY \

-v /tmp/.X11-unix:/tmp/.X11-unix \

--privileged \

deepstream\_trt\_ready:v1 \

/bin/bash

## 2. Custom Parser Code

// Save as nvdsparsebbox\_mmyolo.cpp inside custom\_parser

#include "nvdsinfer\_custom\_impl.h"

#include <cassert>

#include <iostream>

#include <vector>

extern "C" bool NvDsInferParseCustomMMYOLO(

std::vector<NvDsInferLayerInfo> const& outputLayersInfo,

NvDsInferNetworkInfo const& networkInfo,

NvDsInferParseDetectionParams const& detectionParams,

std::vector<NvDsInferParseObjectInfo>& objectList);

static \_\_inline\_\_ float clamp(float val, float min, float max) {

return val > min ? (val < max ? val : max) : min;

}

static std::vector<NvDsInferParseObjectInfo> decodeMMYoloTensor(

const float\* bboxes,

const float\* scores,

const int\* labels,

const unsigned int& num\_dets,

const float& conf\_thres,

const unsigned int& img\_w,

const unsigned int& img\_h)

{

std::vector<NvDsInferParseObjectInfo> bboxInfo;

for (unsigned int i = 0; i < num\_dets; ++i) {

float score = scores[i];

if (score < conf\_thres) continue;

float x0 = bboxes[i \* 4];

float y0 = bboxes[i \* 4 + 1];

float x1 = bboxes[i \* 4 + 2];

float y1 = bboxes[i \* 4 + 3];

float width = x1 - x0;

float height = y1 - y0;

if ((width < 0.01f \* img\_w || height < 0.01f \* img\_h) && score < 0.7f) continue;

if (width > img\_w || height > img\_h) continue;

x0 = clamp(x0, 0.f, img\_w);

y0 = clamp(y0, 0.f, img\_h);

x1 = clamp(x1, 0.f, img\_w);

y1 = clamp(y1, 0.f, img\_h);

NvDsInferParseObjectInfo obj;

obj.left = x0;

obj.top = y0;

obj.width = width;

obj.height = height;

obj.detectionConfidence = score;

obj.classId = labels[i];

bboxInfo.push\_back(obj);

}

return bboxInfo;

}

extern "C" bool NvDsInferParseCustomMMYOLO(

std::vector<NvDsInferLayerInfo> const& outputLayersInfo,

NvDsInferNetworkInfo const& networkInfo,

NvDsInferParseDetectionParams const& detectionParams,

std::vector<NvDsInferParseObjectInfo>& objectList)

{

if (outputLayersInfo.size() < 2) return false;

const float\* bboxes = static\_cast<const float\*>(outputLayersInfo[0].buffer);

const int\* labels = static\_cast<const int\*>(outputLayersInfo[1].buffer);

unsigned int num\_dets = outputLayersInfo[0].inferDims.d[0];

float conf\_thres = detectionParams.perClassThreshold[0];

std::vector<float> scores(num\_dets, 1.0f);

objectList = decodeMMYoloTensor(bboxes, scores.data(), labels, num\_dets, conf\_thres, networkInfo.width, networkInfo.height);

return true;

}

CHECK\_CUSTOM\_PARSE\_FUNC\_PROTOTYPE(NvDsInferParseCustomMMYOLO);

## 3. Compile Shared Library

# Run inside container with correct includes

## g++ -Wall -std=c++14 -shared -fPIC -o libnvdsinfer\_custom\_impl\_mmyolo.so nvdsparsebbox\_mmyolo.cpp \

## `pkg-config --cflags --libs gstreamer-1.0 gstreamer-base-1.0` \

## -I/opt/nvidia/deepstream/deepstream/sources/includes \

## -I/opt/nvidia/deepstream/deepstream/sources/libs/nvdsinfer \

## -I/opt/nvidia/deepstream/deepstream/sources/includes/nvdsinfer\_custom\_impl \

## -I/usr/local/cuda/include \

## -L/opt/nvidia/deepstream/deepstream/lib \

## -L/usr/local/cuda/lib64 \

## -lnvds\_inferutils -lcudart for modified CPP g++ -Wall -std=c++14 -shared -fPIC -o libnvdsinfer\_custom\_impl\_mmyolo\_copy.so nvdsparsebbox\_mmyolo\_modified.cpp `pkg-config --cflags --libs gstreamer-1.0 gstreamer-base-1.0` -I/opt/nvidia/deepstream/deepstream/sources/includes -I/opt/nvidia/deepstream/deepstream/sources/libs/nvdsinfer -I/opt/nvidia/deepstream/deepstream/sources/includes/nvdsinfer\_custom\_impl -I/usr/local/cuda/include -L/opt/nvidia/deepstream/deepstream/lib -L/usr/local/cuda/lib64 -lnvds\_inferutils -lcudart

## 4. DeepStream Config Setup

# deepstream\_app\_config.txt

[application]

enable-perf-measurement=1

[tiled-display]

enable=1

[source0]

uri=file:///path/to/your/input.mp4

[sink0]

type=2

[primary-gie]

config-file=config\_infer\_primary.txt

# config\_infer\_primary.txt

[property]

onnx-file=model.onnx

model-engine-file=end2end\_fp16.engine

labelfile-path=labels.txt

num-detected-classes=3

custom-lib-path=custom\_parser/libnvdsinfer\_custom\_impl\_mmyolo.so

parse-bbox-func-name=NvDsInferParseCustomMMYOLO

## 5. Run Inference

export LD\_PRELOAD=/workspace/mmdeploy/build/lib/libmmdeploy\_tensorrt\_ops.so

## deepstream-app -c /workspace/deepstream-app/configs/deepstream\_app\_config.txt

## 6. Common Errors & Fixes

- Cannot open shared object file:

➜ Check custom-lib-path and run `ldd libnvdsinfer\_custom\_impl\_mmyolo.so`

- cuda\_runtime\_api.h not found:

➜ Ensure CUDA is installed and `-I/usr/local/cuda/include` is added

- libmpeg2 or libmpg123 not found:

➜ These are optional. You can ignore those plugin load warnings

- All boxes showing score=1:

➜ Add confidence filtering logic or extract score from engine if present