**What can you learn from this sample code?**

Through this sample code, you can learn:

1. Set detection results into admetadata.

**Essential knowledge**

1. Plugin sample code, "video filtr", in C/C++.
2. The difference between application code and plugin/element code.
3. ADLINK metadata structure. The metadata structure could be find in Edge Vision Analytics SDK Programming Guide: How to Use ADLINK Metadata in Chapter 5.
4. Basic C/C++ programming.

**About this sample code**

This sample shows how to set the adlink metadata inside the element. The target created pipeline command is:

$ gst-launch-1.0 videotestsrc ! video/x-raw, width=640, height=480 ! adsetobjectdetection ! admetadrawer ! videoconvert ! ximagesink

This command is used to display the simulated inferenced object detection result and the element(this sample code), adsetobjectdetection, set the simulated detection data into admetadata.

**Description**

This sample shows how to set the adlink metadata inside the element. First, include the gstadmeta.h:

#include "gstadmeta.h"

Second, in the virtual method, ad\_set\_object\_detection\_transform\_frame\_ip, of this sample has an extra code block, setObjectDetectionData, which is going to set the detection metadata:

gpointer state = NULL;

GstAdBatchMeta\* meta;

const GstMetaInfo\* info = GST\_AD\_BATCH\_META\_INFO;

meta = (GstAdBatchMeta \*)gst\_buffer\_add\_meta(buffer, info, &state);

bool frame\_exist = meta->batch.frames.size() > 0 ? true : false;

if(!frame\_exist)

{

VideoFrameData frame\_info;

std::vector<adlink::ai::DetectionBoxResult> arr;

std::vector<std::string> labels = {"water bottle", "camera", "chair", "person", "slipper"};

std::vector<adlink::ai::DetectionBoxResult> random\_boxes;

srand( time(NULL) );

// Generate 5 random dummy boxes here

for ( int i = 0 ; i < 5 ; i++ )

{

adlink::ai::DetectionBoxResult temp\_box;

temp\_box.obj\_id = i+1;

temp\_box.obj\_label = labels[i];

temp\_box.prob = (double)( rand() % 1000 )/1000;

temp\_box.x1 = (double)( rand() % 3 + 1 )/10; // 0.1~0.3

temp\_box.x2 = (double)( rand() % 3 + 7 )/10; // 0.7~0.9

temp\_box.y1 = (double)( rand() % 3 + 1 )/10; // 0.1~0.3

temp\_box.y2 = (double)( rand() % 3 + 7 )/10; // 0.7~0.9

random\_boxes.push\_back(temp\_box);

}

frame\_info.stream\_id = " ";

frame\_info.width = 640;

frame\_info.height = 480;

frame\_info.depth = 0;

frame\_info.channels = 3;

frame\_info.device\_idx = 0;

frame\_info.detection\_results.push\_back(random\_boxes[rand()%5]);

meta->batch.frames.push\_back(frame\_info);

}

The metadata structure could be find in Edge Vision Analytics SDK Programming Guide : How to Use ADLINK Metadata in Chapter 5. Or can be found in the files:

* EVA\_INSTALL\_ROOT/include/gstadmeta.h
* EVA\_INSTALL\_ROOT/include/libs/ai/structure.h

Based on the structure, the frame in vector and the inferenced data are stored in each frame need to be set up. When data in this element is empty, we’ll just randomly generate the inference data and set data into the detection result. The simulated result will be set into adbatchmetadata with frame information.

**Run this sample code**

Copy the built plugin libadsetobjectdetection.so file to the plugin folder EVA installed, here used EVA\_ROOT to preset the installed path of EVASDK. Then run the pipeline command for testing:

$ gst-launch-1.0 videotestsrc ! video/x-raw, width=640, height=480 ! adsetobjectdetection ! admetadrawer ! videoconvert ! ximagesink

and you will see the inference result displayed frame by frame in the window:

(video)