

AI-Assisted Video Encoding with AMD Alveo MA35D

Jakub Adam

GStreamer Conference 2024



COLLABORA

Open First

About me

- Working at Collabora on multimedia projects since 2018
- Based in Czech Republic, Europe



COLLABORA

Open First

What is AMD Alveo MA35D?

- media accelerator for application demanding many concurrent streams
- supports accelerating full video pipeline (decoding, adaptive bitrate scaling, encoding)
- H.264, H.265 and AV1
- up to 4x 4k@60fps (H264, H265, AV1) + 4x 4k@60fps AV1 encodes per card



AI features

- Dedicated AI processor performing inference on video frames
- Can potentially execute multiple machine learning models
- Currently enabled ML model is Region of Interest, detecting faces and text (more to come)
- ML engine output can help MA35D video encoders achieve better video quality during compression



GStreamer support

MA35D SDK 1.2 comes with a set of GST elements that expose capabilities of the hardware:

Encoders and decoders

ama_av1dec: AMA AV1 decoder
ama_av1enc: AMA AV1 encoder
ama_h264dec: AMA H264 decoder
ama_h264enc: AMA H264 encoder
ama_h265dec: AMA H265 decoder
ama_h265enc: AMA H265 encoder
ama_jpegdec: AMA JPEG decoder
ama_jpegenc: AMA JPEG encoder
ama_ljpegenc: AMA Lossless JPEG encoder
ama_vp9dec: AMA VP9 decoder

Video effects

ama_rotate: AMA rotate
ama_scaler: AMA ABR scaler
ama_tile: AMA tile
ama_videoconvert: AMA videoconvert
ama_videocrop: AMA videocrop
ama_videopad: AMA videopad
ama_overlay: AMA overlay
ama_blend: AMA blend
ama_compositor: AMA compositor
ama_drawbox: AMA draw box

System ↔ AMA memory transfers

ama_upload: AMA uploader
ama_download: AMA downloader



GStreamer support

AI features coming with the next SDK release

Encoders and decoders

ama_av1dec: AMA AV1 decoder
ama_av1enc: AMA AV1 encoder
ama_h264dec: AMA H264 decoder
ama_h264enc: AMA H264 encoder
ama_h265dec: AMA H265 decoder
ama_h265enc: AMA H265 encoder
ama_jpegdec: AMA JPEG decoder
ama_jpegenc: AMA JPEG encoder
ama_ljpegenc: AMA Lossless JPEG encoder
ama_vp9dec: AMA VP9 decoder

System ↔ AMA memory transfers

ama_upload: AMA uploader
ama_download: AMA downloader

Video effects

ama_rotate: AMA rotate
ama_scaler: AMA ABR scaler
ama_tile: AMA tile
ama_videoconvert: AMA videoconvert
ama_videocrop: AMA videocrop
ama_videopad: AMA videopad
ama_overlay: AMA overlay
ama_blend: AMA blend
ama_compositor: AMA compositor
ama_drawbox: AMA draw box

AI features

ama_ml_roi: Xilinx ML ROI filter
ama_roiscale: Xilinx ROI Scale filter



COLLABORA

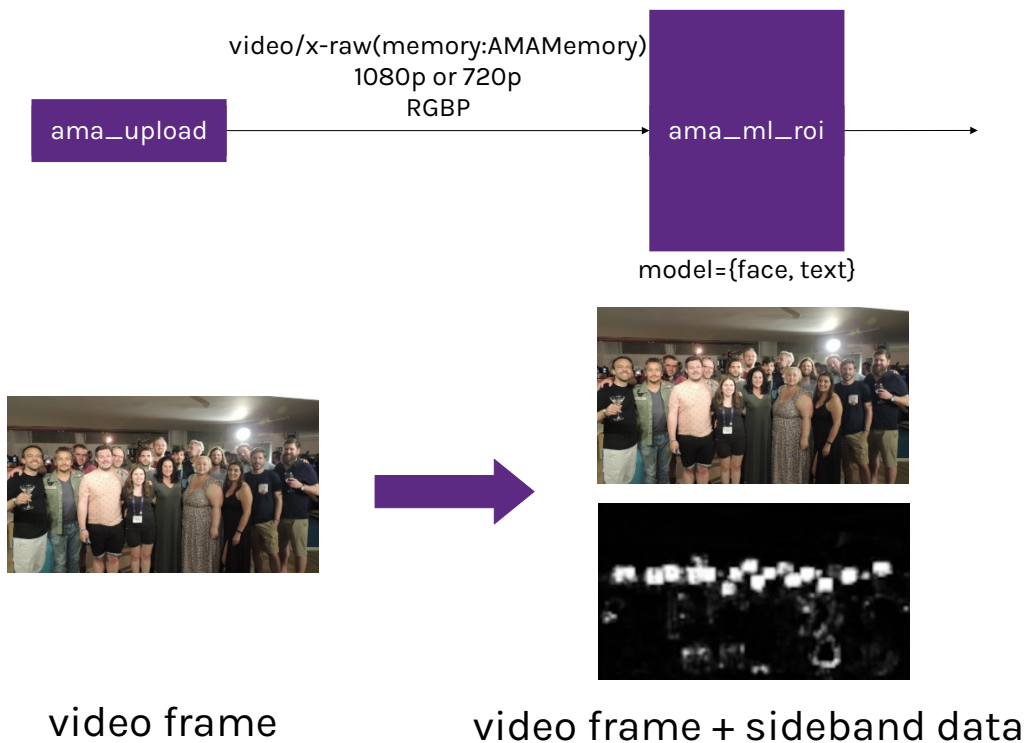
Open First



COLLABORA

AMA ML GStreamer elements

ama_ml_roi



- Input:

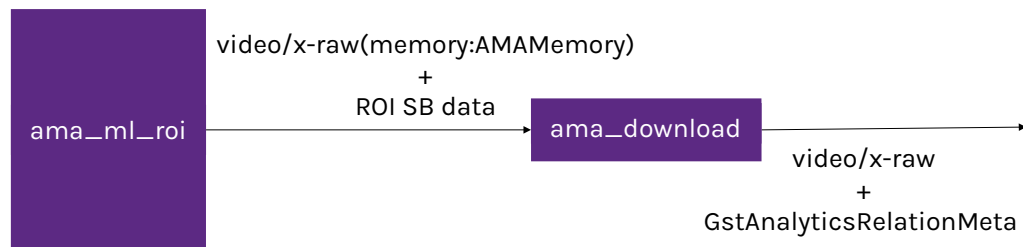
- frame in RGBP format
- Resolutions: 1080p, 720p

- Output:

- Identical buffer with attached 8-bit grayscale map, each pixel representing 16x16 region
- Map contains values 0-255
- Higher value indicates ROI region



When you want to inspect filter output yourself...



- ama_download converts HW-specific SB data to GstAnalyticsRelationMeta
- Not required when SB is consumed by another AMA element.



Reading out ROI data

```
GstAnalyticsRelationMeta *meta = gst_buffer_get_analytics_relation_meta (outbuf);

if (meta) {
    GstAnalyticsMtdType od_type = gst_analytics_od_mtd_get_mtd_type ();
    GstAnalyticsMtd mtd;
    gpointer state = NULL;

    while (gst_analytics_relation_meta_iterate (meta, &state, od_type, &mtd)) {
        /* Each instance of GstAnalyticsODMtd represents one 16x16 region (pixel). Regions with confidence level 0.0 are omitted. */
        GstAnalyticsODMtd *od_mtd = &mtd;

        /* If the frame was passed through multiple ML filters, output of each filter will have different obj_type: roi-0, roi-1, etc. */
        const gchar *obj_type = g_quark_to_string (gst_analytics_od_mtd_get_obj_type (od_mtd));

        gint x, y, w, h;
        gfloat confidence_lvl;

        gst_analytics_od_mtd_get_location (od_mtd, &x, &y, &w, &h, &confidence_lvl);
        gst_println ("Found object detection region from %s x:%d y:%d w:%d h:%d confidence lvl %f", obj_type, x, y, w, h, confidence_lvl);
    }
}
```



Better representation for ML filter output – GstTensorMeta

ROI SB follows standard NHWC layout (N=1, C=1)

```
typedef struct _GstTensor
{
    GQuark id;
    gint num_dims;
    int64_t *dims;
    GstTensorDataType data_type;
    GstBuffer *data;
} GstTensor;
```

Simple memcpy of ROI SB data into a GstBuffer.

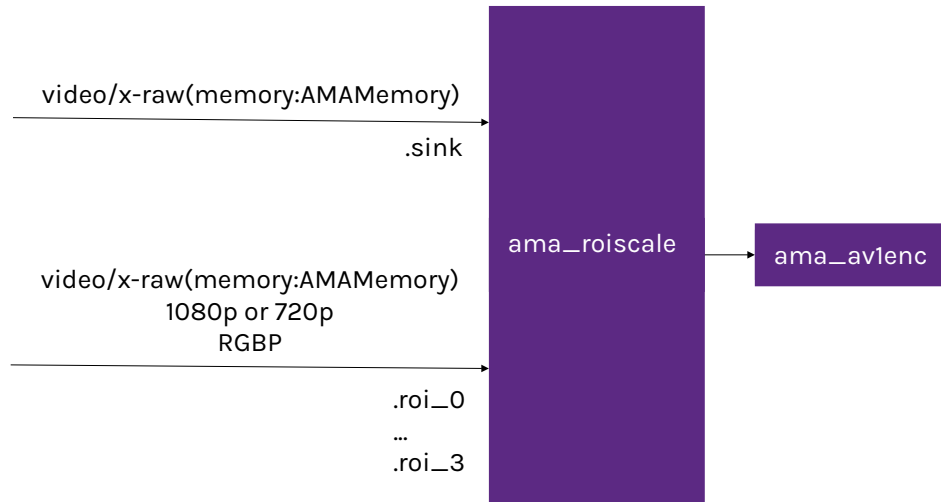
data_type = UINT8, num_dims=2

Currently private to ONNX plugin, will become public in Gst Analytics:

https://gitlab.freedesktop.org/gstreamer/gstreamer/-/merge_requests/6000



ama_roiscale



- Input:
 - Original video up to 4K resolution
 - Up to 4 ama_ml_roi outputs
- Element
 - Merges ROI inputs
 - Scales it to the size of input frame
 - Converts to delta QP map to feed the encoder
- Output:
 - Video frame + delta QP map as SB data



ama_roiscale

- AMA video encoders use the delta QP map to enhance video resolution in detected areas of interest



original encoding



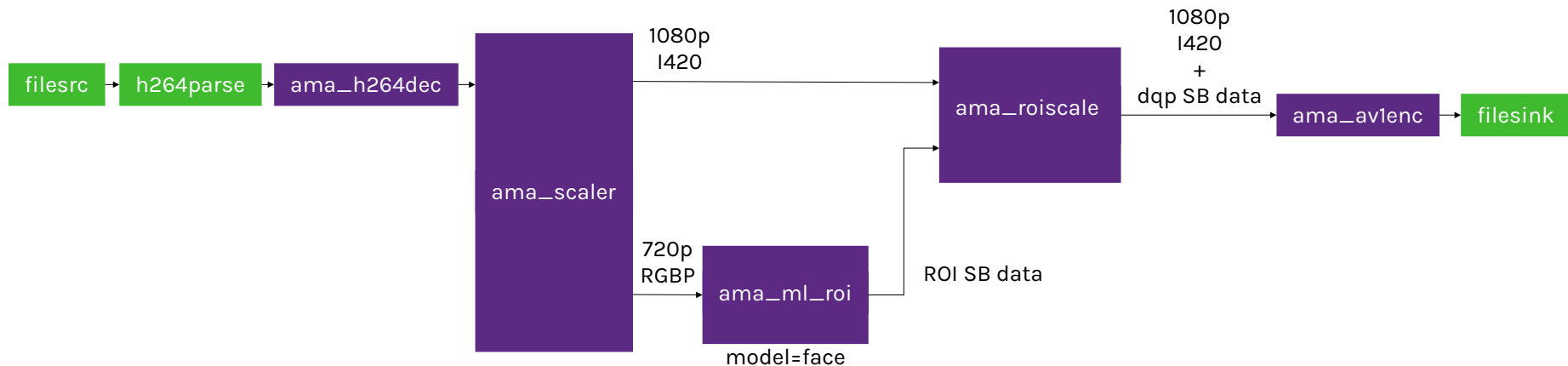
with ROI enhancements





Pipeline examples

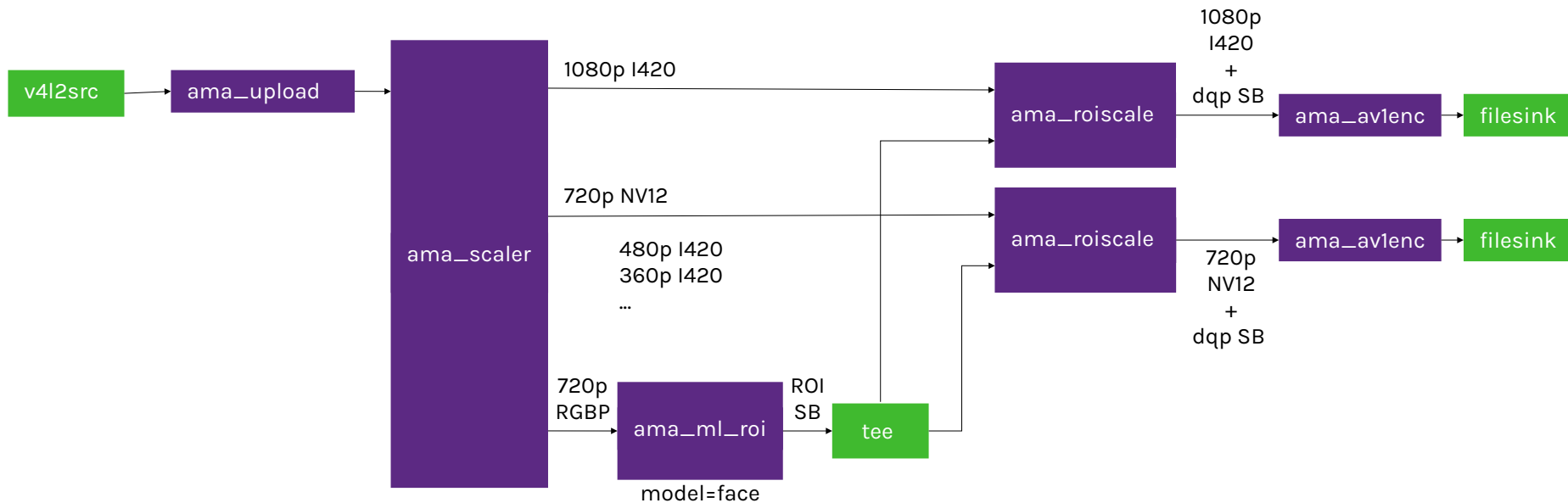
Basic transcoding scenario



```
filesrc location=input.h264 ! h264parse ! ama_h264dec ! ama_scaler name=s  
s.src_0 ! 'video/x-raw(memory:AMAMemory),width=1280,height=720' ! roiscale.sink  
s.src_1 ! 'video/x-raw(memory:AMAMemory),width=1280,height=720' !  
ama_ml_roi model=face ! roiscale.roi_0  
ama_roiscale name=roiscale ! ama_av1enc ! filesink location=output.av1
```



ABR Ladder



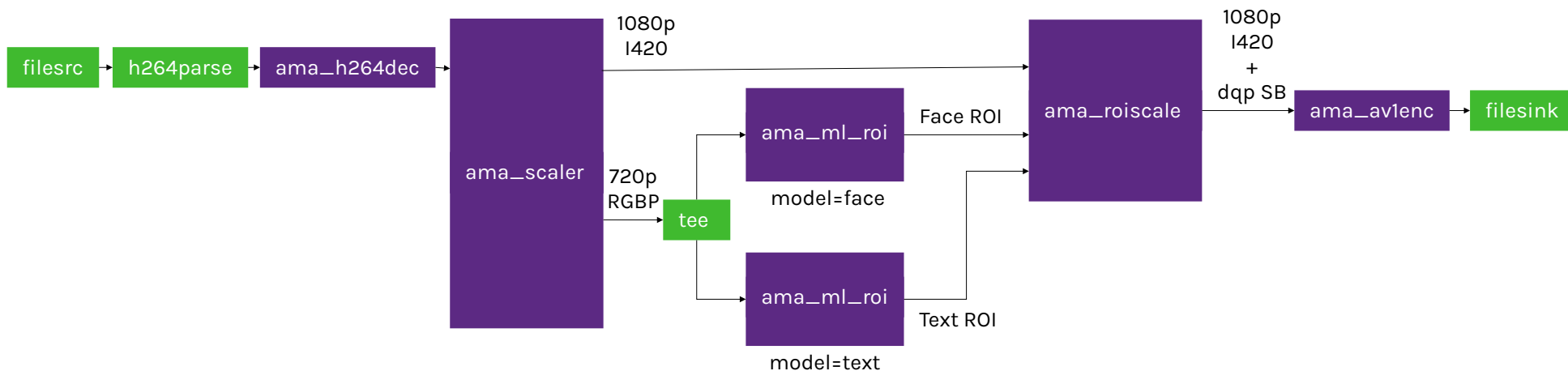
ABR Ladder

```
v4l2src ! ama_upload ! ama_scaler name=s
  s.src_0 ! 'video/x-raw(memory:AMAMemory),width=1920,height=1080,format=I420'
    ! roi1080.sink
  s.src_1 ! 'video/x-raw(memory:AMAMemory),width=1280,height=720,format=NV12'
    ! roi720.sink
  s.src_2 ! 'video/x-raw(memory:AMAMemory),width=1280,height=720,format=RGBP'
    ! ama_ml_roi model=face ! tee name=t
      t.src_0 ! ama_roiscale name=roi1080 ! ama_av1enc ! filesink location=out1080.av1
      t.src_1 ! ama_roiscale name=roi720 ! ama_av1enc ! filesink location=out720.av1
```



Merging ROI outputs – parallel

ama_roiscale can aggregate outputs of up to 4 ML filters.



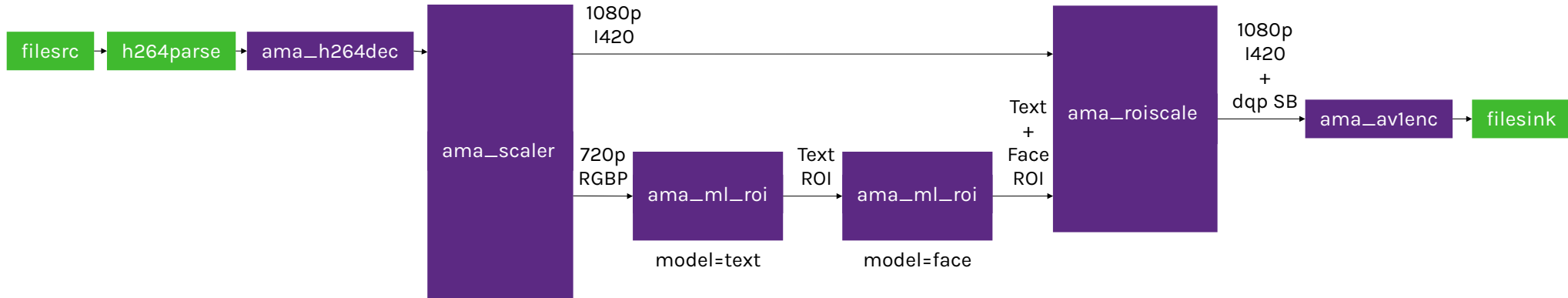
Merging ROI outputs – parallel

```
filesrc location=input.h264 ! h264parse ! ama_h264dec ! ama_scaler name=s  
s.src_0 ! 'video/x-raw(memory:AMAMemory),width=1920,height=1080' ! roiscale.sink  
s.src_1 ! 'video/x-raw(memory:AMAMemory),width=1280,height=720' ! tee name=t  
t.src_0 ! ama_ml_roi model=face ! roiscale.roi_0  
t.src_1 ! ama_ml_roi model=text ! roiscale.roi_1  
ama_roiscale name=roiscale ! ama_av1enc ! filesink location=output.av1
```



Merging ROI outputs – serial

ML filters append their output to sideband data of the frame in device memory, so a frame can collect outputs of several ML filters one by one.



Merging ROI outputs – serial

```
filesrc location=input.h264 ! h264parse ! ama_h264dec ! ama_scaler name=s  
s.src_0 ! 'video/x-raw(memory:AMAMemory),width=1920,height=1080' ! roiscale.sink  
s.src_1 ! 'video/x-raw(memory:AMAMemory),width=1280,height=720'  
! ama_ml_roi model=face ! ama_ml_roi model=text ! roiscale.roi_0  
ama_roiscale name=roiscale ! ama_av1enc ! filesink location=output.av1
```



Source code

GStreamer integration is open source under a dual license, but depends on some proprietary middleware libraries.

```
/* SPDX-License-Identifier: LGPL-3.0-or-later OR Apache-2.0 */
```

- https://github.com/amd/ama-sdk/blob/v1.2/sources/source_access.md
- https://www.xilinx.com/content/dam/xilinx/guest-resources/2024/sdk/ma35d_sdk_v1.2.0_gstreamer.zip

ama_ml_roi and ama_ml_roidscale elements coming soon.





Thank you!



COLLABORA

Open First