README.md

Video Analytics Serving Object Identification

This sample demonstrates how to run and configure an object identification pipeline based on Video Analytics Serving and DL Streamer. The sample uses the face-detection-adas-0001 model for detection and the face-recognition-resnet100-arcface model for recognition but can be customized to use any detection and recognition model.

Overview

Object Identification

Object identification is the task of detecting and matching items against a set of known objects. It is a combination of object detection and object recognition. Object detection models detect and label regions of interest. Object recognition models take regions of interest and embed them in a tensor space where two embeddings of the same object have a cosine similarity close to 1. Object identification applications typically keep a gallery of known objects each with a set of representative embeddings. To identify an object the application calculates its embedding and then calculates its similarity to all known objects. If an existing object matches with a cosine similarity greater than the threshold it is considered the same object and labeled. If no objects match within the threshold the object is labeled as "UNKNOWN".

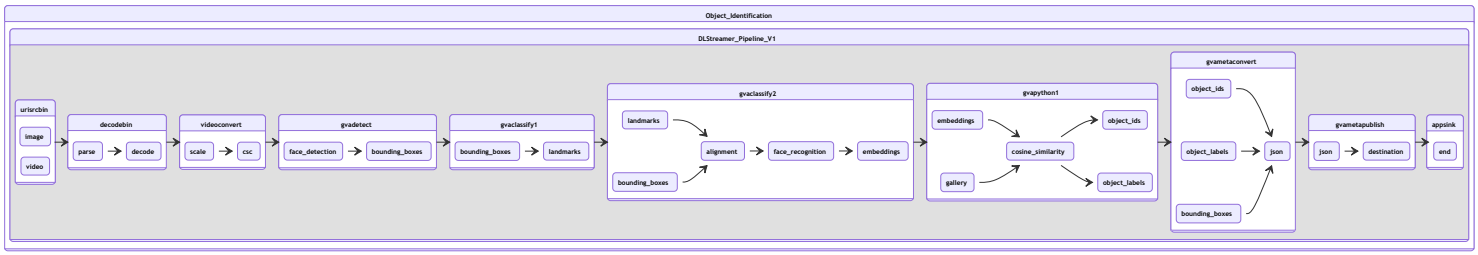
Pipelines

The object identification sample uses a DL Streamer based pipeline definition with two versions.

Pipeline Version 1

Pipeline version 1 uses standard gstreamer elements for parsing, decoding, and converting incoming media files, gvadetect to detect objects in each frame, gvaclassify to generate landmarks for each object, gvaclassify to generate embeddings for each object, gvapython to match incoming embeddings against the known gallery and finally gvametaconvert and gvametapublish to publish results.

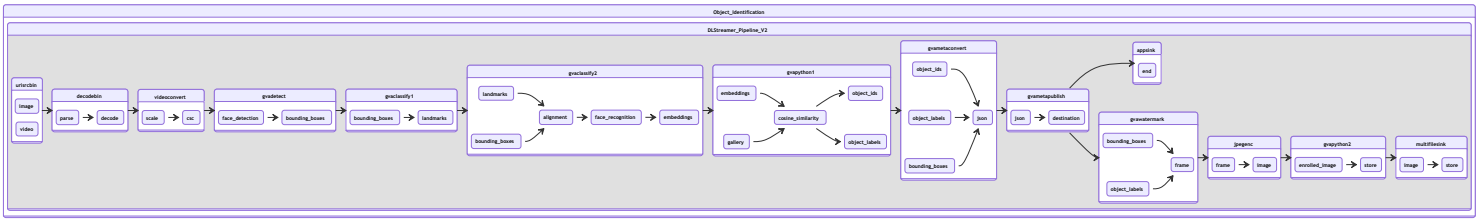
```
        "urisourcebin name=source ! decodebin ! videoconvert name=videoconvert ! video/x-raw,format=BGRx",
        " ! gvadetect model-instance-id=detection model={models[face_detection_adas][1][network]} name=detection",
        " ! gvaclassify model={models[landmarks_regression][1][network]} model-proc={models[landmarks_regression][1][proc]}
name=landmarks model-instance-id=landmarks",
        " ! gvaclassify model={models[face_recognition][1][network]} model-proc={models[face_recognition][1][proc]}
name=recognition model-instance-id=recognition",
        " ! gvapython name=identify module=/home/video-analytics-serving/extensions/vas_identify.py class=Identify",
        " ! gvametaconvert name=metaconvert ! queue ! gvametapublish name=destination",
        " ! appsink name=appsink"
```



Pipeline Version 2

Pipeline version 2 extends pipeline version 1 with gvawatermark to draw bounding boxes and labels on frames, jpegenc to encode frames as images, gvapython to store frames enrolled in the gallery and finally multifilesink to store a rolling window of watermarked images.

```
        "urisourcebin name=source ! decodebin ! videoconvert name=videoconvert ! video/x-raw,format=BGRx",
        " ! gvadetect model-instance-id=detection model={models[face_detection_adas][1][network]} name=detection",
        " ! gvaclassify model={models[landmarks_regression][1][network]} model-proc={models[landmarks_regression][1][proc]}
name=landmarks model-instance-id=landmarks",
        " ! gvaclassify model={models[face_recognition][1][network]} model-proc={models[face_recognition][1][proc]}
name=recognition model-instance-id=recognition",
        " ! gvapython name=identify module=/home/video-analytics-serving/extensions/vas_identify.py class=Identify",
        " ! gvametaconvert name=metaconvert ! queue ! gvametapublish name=destination",
        " ! tee name = t ! queue ! gvawatermark ! videoconvert ! jpegenc",
        " ! gvapython name=capture module=/home/video-analytics-serving/extensions/vas_identify.py class=Capture",
        " ! multifilesink name=filesink t. ! queue",
        " ! appsink name=appsink"
```



Nearest Neighbor Matching Algorithm

The Video Analytics Serving gvapython extension performs a simple lookup to match incoming objects. It calculates the pairwise similarity between the object with all embeddings in the gallery and chooses the known object with the highest similarity. If no known objects are within the specified threshold then the object is labeled as UNKNOWN. If the pipeline parameters have been configured to enroll new objects - the tensor for the object is saved and the gallery updated.

Installation

1. Clone repository:

```
git clone -b v0.3.1.1-alpha https://github.com/intel/video-analytics-serving.git
```

2. Apply patch

```
cd video-analytics-serving
git apply ../object-identification-sample.patch
```

3. Download and convert face-recognition-resnet100-arcface (takes several minutes)

```
./samples/object_identification/download_recognition_model.sh
```

4. Build video analytics serving docker image

```
./docker/build.sh --base openvisualcloud/xeon-ubuntu1804-analytics-gst
```

Tutorial

Create Gallery from classroom.mp4

1. Launch video analytics serving dev environment

```
./docker/run.sh --dev
```

2. Process classroom.mp4 and enroll detected objects

   By default the sample processes the file: `video-analytics-serving/samples/classroom.mp4` which contains four individuals.

```
python3 ./samples/object_identification/object_identification.py --enroll --max-enrolled 4 --dump-frames --stop-on-max
```

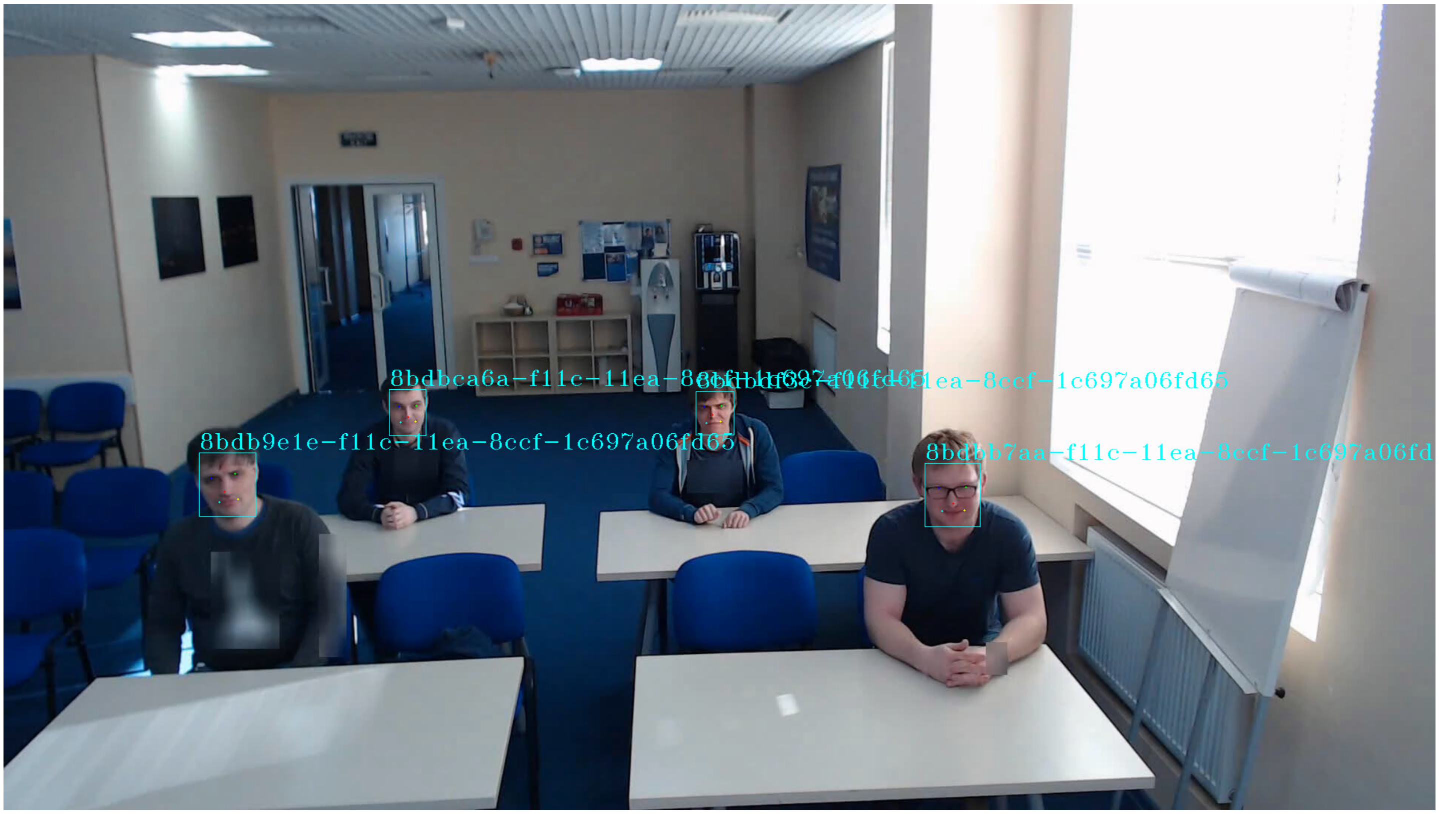
   Expected Output:

```
Max Enrollment Reached

----------------
Original Gallery
----------------



---------------
Updated Gallery
---------------

 Object: 8bdb9e1e-f11c-11ea-8ccf-1c697a06fd65 Tensors: 10
 Object: 8bdbb7aa-f11c-11ea-8ccf-1c697a06fd65 Tensors: 10
 Object: 8bdbca6a-f11c-11ea-8ccf-1c697a06fd65 Tensors: 10
 Object: 8bdbdf3c-f11c-11ea-8ccf-1c697a06fd65 Tensors: 10


-------
Results
-------

 classroom.mp4

 Total Objects: 40
 Matched Objects: 40
 Unknown Objects: 0

 Object: 8bdb9e1e-f11c-11ea-8ccf-1c697a06fd65 Matches: 10

 Object: 8bdbb7aa-f11c-11ea-8ccf-1c697a06fd65 Matches: 10

 Object: 8bdbca6a-f11c-11ea-8ccf-1c697a06fd65 Matches: 10

 Object: 8bdbdf3c-f11c-11ea-8ccf-1c697a06fd65 Matches: 10

 Object: UNKNOWN Matches: 0
```

3. View Enrolled Frames

   Enrolled frames are stored in the host at:

```
video-analytics-serving/pipelines/gstreamer/object_identification/gallery/features/<object-id>/<object-id>.<index>.jpeg
```



Objects with blue bounding boxes have been used for enrollment. Objects with green bounding boxes have been detected but not enrolled.

4. Rename Objects in Gallery

   By default new objects are enrolled with UUIDs. For convenience we can rename the labels to simpler values. The mapping of name to enrolled tensors is stored in: `video-analytics-serving/pipelines/gstreamer/object_identification/gallery/gallery.json` and can be edited with any text editor.

   Change the `name` field of each json object to a name of your choice.

   Before:

```
    {
     "features": [
         "/home/video-analytics-serving/pipelines/object_identification/gallery/features/8bdb9e1e-f11c-11ea-8ccf-1c697a06fd65
         "/home/video-analytics-serving/pipelines/object_identification/gallery/features/8bdb9e1e-f11c-11ea-8ccf-1c697a06fd65
         "/home/video-analytics-serving/pipelines/object_identification/gallery/features/8bdb9e1e-f11c-11ea-8ccf-1c697a06fd65
         "/home/video-analytics-serving/pipelines/object_identification/gallery/features/8bdb9e1e-f11c-11ea-8ccf-1c697a06fd65
         "/home/video-analytics-serving/pipelines/object_identification/gallery/features/8bdb9e1e-f11c-11ea-8ccf-1c697a06fd65
         "/home/video-analytics-serving/pipelines/object_identification/gallery/features/8bdb9e1e-f11c-11ea-8ccf-1c697a06fd65
         "/home/video-analytics-serving/pipelines/object_identification/gallery/features/8bdb9e1e-f11c-11ea-8ccf-1c697a06fd65
         "/home/video-analytics-serving/pipelines/object_identification/gallery/features/8bdb9e1e-f11c-11ea-8ccf-1c697a06fd65
         "/home/video-analytics-serving/pipelines/object_identification/gallery/features/8bdb9e1e-f11c-11ea-8ccf-1c697a06fd65
         "/home/video-analytics-serving/pipelines/object_identification/gallery/features/8bdb9e1e-f11c-11ea-8ccf-1c697a06fd65
     ],
     "name": "8bdb9e1e-f11c-11ea-8ccf-1c697a06fd65"
 }
```

   After:

```
    {
     "features": [
```

```
        "/home/video-analytics-serving/pipelines/object_identification/gallery/features/8bdb9e1e-f11c-11ea-8ccf-1c697a06fd65
        "/home/video-analytics-serving/pipelines/object_identification/gallery/features/8bdb9e1e-f11c-11ea-8ccf-1c697a06fd65
        "/home/video-analytics-serving/pipelines/object_identification/gallery/features/8bdb9e1e-f11c-11ea-8ccf-1c697a06fd65
        "/home/video-analytics-serving/pipelines/object_identification/gallery/features/8bdb9e1e-f11c-11ea-8ccf-1c697a06fd65
        "/home/video-analytics-serving/pipelines/object_identification/gallery/features/8bdb9e1e-f11c-11ea-8ccf-1c697a06fd65
        "/home/video-analytics-serving/pipelines/object_identification/gallery/features/8bdb9e1e-f11c-11ea-8ccf-1c697a06fd65
        "/home/video-analytics-serving/pipelines/object_identification/gallery/features/8bdb9e1e-f11c-11ea-8ccf-1c697a06fd65
        "/home/video-analytics-serving/pipelines/object_identification/gallery/features/8bdb9e1e-f11c-11ea-8ccf-1c697a06fd65
        "/home/video-analytics-serving/pipelines/object_identification/gallery/features/8bdb9e1e-f11c-11ea-8ccf-1c697a06fd65
        "/home/video-analytics-serving/pipelines/object_identification/gallery/features/8bdb9e1e-f11c-11ea-8ccf-1c697a06fd65
    ],
    "name": "person-1"
}
```


Update Gallery from head-pose-face-detection Sample Videos

1. Enroll person from head-pose-face-detection-male.mp4

   Since the sample video includes a single person we update the command to allow 5 max enrollments (4 from the existing gallery, one from the new video).

   ```
   python3 ./samples/object_identification/object_identification.py --source https://github.com/intel-iot-devkit/sample-videos/b
   ```

   Expected Output:

   ```
   Max Enrollment Reached

   ----------------
   Original Gallery
   ----------------

    Object: person-1 Tensors: 10
    Object: person-2 Tensors: 10
    Object: person-3 Tensors: 10
    Object: person-4 Tensors: 10


   ---------------
   Updated Gallery
   ---------------

    Object: person-1 Tensors: 10
    Object: person-2 Tensors: 10
    Object: person-3 Tensors: 10
    Object: person-4 Tensors: 10
    Object: 40919eee-f122-11ea-be67-1c697a06fd65 Tensors: 10


   -------
   Results
   -------

    head-pose-face-detection-male.mp4?raw=true

     Total Objects: 15
     Matched Objects: 15
     Unknown Objects: 0

     Object: 40919eee-f122-11ea-be67-1c697a06fd65 Matches: 15

     Object: UNKNOWN Matches: 0
   ```

```
classroom.mp4

 Total Objects: 40
 Matched Objects: 40
 Unknown Objects: 0

 Object: 8bdb9e1e-f11c-11ea-8ccf-1c697a06fd65 Matches: 10

 Object: 8bdbb7aa-f11c-11ea-8ccf-1c697a06fd65 Matches: 10

 Object: 8bdbca6a-f11c-11ea-8ccf-1c697a06fd65 Matches: 10

 Object: 8bdbdf3c-f11c-11ea-8ccf-1c697a06fd65 Matches: 10

 Object: UNKNOWN Matches: 0
```

2. Enroll person from head-pose-face-detection-female.mp4

   Since the sample video includes a single person we update the command to allow 6 max enrollments (5 from the existing gallery, one from the new video).

```
python3 ./samples/object_identification/object_identification.py --source https://github.com/intel-iot-devkit/sample-videos/b
```

Expected Output:

```
Max Enrollment Reached

----------------
Original Gallery
----------------

 Object: person-1 Tensors: 10
 Object: person-2 Tensors: 10
 Object: person-3 Tensors: 10
 Object: person-4 Tensors: 10
 Object: 40919eee-f122-11ea-be67-1c697a06fd65 Tensors: 10


---------------
Updated Gallery
---------------

 Object: person-1 Tensors: 10
 Object: person-2 Tensors: 10
 Object: person-3 Tensors: 10
 Object: person-4 Tensors: 10
 Object: 40919eee-f122-11ea-be67-1c697a06fd65 Tensors: 10
 Object: 8809c8f4-f123-11ea-8678-1c697a06fd65 Tensors: 10


-------
Results
-------

 head-pose-face-detection-male.mp4?raw=true

  Total Objects: 15
  Matched Objects: 15
  Unknown Objects: 0

  Object: 40919eee-f122-11ea-be67-1c697a06fd65 Matches: 15
```

```
 Object: UNKNOWN Matches: 0

classroom.mp4

 Total Objects: 40
 Matched Objects: 40
 Unknown Objects: 0

 Object: 8bdb9e1e-f11c-11ea-8ccf-1c697a06fd65 Matches: 10

 Object: 8bdbb7aa-f11c-11ea-8ccf-1c697a06fd65 Matches: 10

 Object: 8bdbca6a-f11c-11ea-8ccf-1c697a06fd65 Matches: 10

 Object: 8bdbdf3c-f11c-11ea-8ccf-1c697a06fd65 Matches: 10

 Object: UNKNOWN Matches: 0

head-pose-face-detection-female.mp4?raw=true

 Total Objects: 22
 Matched Objects: 22
 Unknown Objects: 0

 Object: 8809c8f4-f123-11ea-8678-1c697a06fd65 Matches: 22

 Object: UNKNOWN Matches: 0
```

3. Rename Objects in Gallery

   As above, rename the new objects as `person-5` and `person-6`

Identify Objects in head-pose-face-detection Sample Video

With the created the gallery we can identify objects in videos and images.

1. Identify objects in head-pose-face-detection-female-and-male without enrolling

```
python3 ./samples/object_identification/object_identification.py --source https://github.com/intel-iot-devkit/sample-videos/blob/master/head-pose-face-detection-female-and-male.mp4?raw=true --dump-frames
```

> Note: as the video is long you can press Ctrl-C to exit the processing at any time to see intermediate results

Expected Output:

```
-------
Gallery
-------

 Object: person-1 Tensors: 10
 Object: person-2 Tensors: 10
 Object: person-3 Tensors: 10
 Object: person-4 Tensors: 10
 Object: person-5 Tensors: 10
 Object: person-6 Tensors: 10


-------
Results
-------

 head-pose-face-detection-male.mp4?raw=true

     Total Objects: 15
```

```
            Matched Objects: 15
            Unknown Objects: 0

            Object: 40919eee-f122-11ea-be67-1c697a06fd65 Matches: 15

            Object: UNKNOWN Matches: 0

        classroom.mp4

            Total Objects: 40
            Matched Objects: 40
            Unknown Objects: 0

            Object: 8bdb9e1e-f11c-11ea-8ccf-1c697a06fd65 Matches: 10

            Object: 8bdbb7aa-f11c-11ea-8ccf-1c697a06fd65 Matches: 10

            Object: 8bdbca6a-f11c-11ea-8ccf-1c697a06fd65 Matches: 10

            Object: 8bdbdf3c-f11c-11ea-8ccf-1c697a06fd65 Matches: 10

            Object: UNKNOWN Matches: 0

        head-pose-face-detection-female.mp4?raw=true

            Total Objects: 22
            Matched Objects: 22
            Unknown Objects: 0

            Object: 8809c8f4-f123-11ea-8678-1c697a06fd65 Matches: 22

            Object: UNKNOWN Matches: 0

        head-pose-face-detection-female-and-male.mp4?raw=true

            Total Objects: 314
            Matched Objects: 75
            Unknown Objects: 239

            Object: person-5 Matches: 70

            Object: person-6 Matches: 5

            Object: UNKNOWN Matches: 239
```

Deploying Object Identification as a Microservice

The created gallery and pipeline can be used with the sample Video Analytics Serving microservice.

1. Rebuild the deployment container

```
./docker/build.sh --base openvisualcloud/xeon-ubuntu1804-analytics-gst
```

2. Run the container as a Microservice

```
./docker/run.sh -e IGNORE_INIT_ERRORS=true -v /tmp:/tmp
```

3. Run a client container

```
./docker/run.sh --dev --name client -v /tmp:/tmp
```

4. Run sample REST client to start pipeline

```
python3 samples/sample.py --pipeline object_identification --source https://github.com/intel-iot-devkit/sample-
videos/blob/master/head-pose-face-detection-female-and-male.mp4?raw=true --destination
/tmp/object_identification_results.txt > /dev/null 2>&1 &
```

5. Tail output file to see incremental results

```
tail -f /tmp/object_identification_results.txt
```

Expected Output:


```
{"objects":[{"detection":{"bounding_box":
{"x_max":0.823763370513916,"x_min":0.7017346620559692,"y_max":0.5149155259132385,"y_min":0.19162707030773163},"confidence":0.
{"label":"person-5","model":{"name":"face-recognition-resnet100-arcface"}},"h":140,"w":94,"x":539,"y":83},{"detection":
{"bounding_box":
{"x_max":0.3044722080230713,"x_min":0.17744578421115875,"y_max":0.547193169593811,"y_min":0.23166534304618835},"confidence":0
{"label":"UNKNOWN","model":{"name":"face-recognition-resnet100-arcface"}},"h":136,"w":98,"x":136,"y":100}],"resolution":
{"height":432,"width":768},"source":"https://github.com/intel-iot-devkit/sample-videos/blob/master/head-pose-face-
detection-female-and-male.mp4?raw=true","timestamp":4500000000}
```


Script Arguments

```
usage: object_identification.py [-h] [--source SOURCE] [--gallery GALLERY]
                                [--enroll] [--dump-frames]
                                [--stop-on-max-enrolled] [--clear-results]
                                [--clear-gallery] [--threshold THRESHOLD]
                                [--max-enrolled MAX_ENROLLED]
                                [--max-time-between-enrollments MAX_TIME_BETWEEN_ENROLLMENTS]
                                [--max-enrolled-tensors MAX_ENROLLED_TENSORS]
                                [--max-dumped-frames MAX_DUMP_FRAMES]
                                [--label-from-path]
                                [--destination DESTINATION]
                                [--extensions [EXTENSIONS [EXTENSIONS ...]]]

optional arguments:
  -h, --help            show this help message and exit
  --source SOURCE       uri or directory to process. Directory is searched for
                        media files having supported extensions
                        (.jpg,.jpeg,.mp4) (default: file:///home/video-
                        analytics-serving/samples/classroom.mp4)
  --gallery GALLERY     location to store gallery and features (default:
                        /home/video-analytics-
                        serving/pipelines/object_identification/gallery)
  --enroll              enroll detected objects into gallery (default: False)
  --dump-frames         dump frames with bounding boxes for debugging
                        (default: False)
  --stop-on-max-enrolled
                        stop pipeline when max enrollment is reached.
                        (default: False)
  --clear-results       clear results (default: False)
  --clear-gallery       clear gallery (also clears results) (default: False)
  --threshold THRESHOLD
                        similarity threshold for identification. Objects whose
                        cosine similaritgy > threshold are matched (default: 0.6)
  --max-enrolled MAX_ENROLLED
                        max number of objects to enroll (default: 100)
  --max-time-between-enrollments MAX_TIME_BETWEEN_ENROLLMENTS
                        maximum time in seconds between new enrollments
```

```
                      (default: 60)
  --max-enrolled-tensors MAX_ENROLLED_TENSORS
                      max number of tensors to enroll per object (default:
                      10)
  --max-dumped-frames MAX_DUMP_FRAMES
                      max frames to dump with bounding boxes for debugging
                      (default: 10)
  --label-from-path   label objects based on path or filename (default:
                      False)
  --destination DESTINATION
                      directory to store results and dumped frames (default:
                      /home/video-analytics-
                      serving/samples/object_identification/results)
  --extensions [EXTENSIONS [EXTENSIONS ...]]
                      supported media files (default: ['mp4', 'jpg',
                      'jpeg'])
```