

Homework #4: Live streaming

To create live streaming with MPEG-DASH or APPLE HLS and perform object segmentation or tracking using deep learning models. Directly using twitch, livehouse, or youtube for streaming is not allowed.

To create live streaming with MPEG-DASH, you will need to

- Use webcam to capture/encode live video
- Use your deep learning network model to do segmentation or object tracking
- Use ffmpeg to transcode the video into .mp4
- Use MP4Box with a live profile (there are two profiles supported: live and onDemand) to generate fragmented mp4 and a .mpd file
- Create HTML file to play the video (or with controls such as play, pause)
- View the results in a browser that supports HTML5.

To create live streaming with HTTP Live Streaming (HLS), you will need to

- Use webcam to capture/encode live video
- Use your deep learning network model to do segmentation or object tracking
- Use ffmpeg to transcode the video into .ts
- Use Stream segmenter or file segmenter to generate a series of small media files (.ts) and an index file (.m3u8)
- To support various bitrates, you can use VariantPlaylistCreator to generate master index file (.m3u8)
- Create HTML file to play the video (or with controls such as play, pause)
- View the results in a browser that supports HTML5

To support for object segmentation or object tracking, you can simply choose a deep learning model to use. No need to do any improvement.

Teamwork:

- You can make a team with at most 4 people for doing this homework

Grading:

1. Report of your work and answer to TA questions during demonstration (20%)
(Please carry your report when doing demonstration)
(Put contribution of each member on your report)
2. Demonstrate the functionality of your system on **1/11~1/15** (80%)
The grading of your system
 - ✧ Base Model (40%): video/audio live steaming
 - ✧ Function 1 (30%): video live streaming with object tracking
 - ✧ Function 2 (10%): Providing a user interface to specify which object to track
 - ✧ *Bonus (10%): supporting multiple bitrates OR trick mode (users can perform pause/resume/rewind)*

■ Reference for video controls and trick model support

✧ To create a simple web page that display DASH video, you will need to:

- Create an HTML page (add the video tag)
- Add the dash.js player

// Create an HTML page

```
<video src="my.mpd" type="application/dash+xml"> </video>
```

// add the dash player

```
<script src="js/dash.all.js"></script>
```

✧ To create a web page that displays a DASH video player with controls such as play, pause, rewind etc., you will need to:

- Create an HTML page (add the video tag, and controls)
- Add the dash.js player
- Initialize the player
- Add some CSS style

// Create an HTML page

```
<!DOCTYPE html>
<html>
  <head><title>Adaptive Streaming in HTML5</title></head>
  <body>
    <h1>Adaptive Streaming with HTML5</h1>
    <video id="videoplayer" controls></video>
  </body>
</html>
```

// add the dash player

```
<script src="js/dash.all.js"></script>
```

//initialize the player

```
// setup the video element and attach it to the Dash player

function setupVideo() {

  var url = "http://wams.edgesuite.net/media/MPTExpressionData02/BigBuckBunny_1080p24_
  IYUV_2ch.ism/manifest(format=mpd-time-csf)";
```

```
var context = new Dash.di.DashContext();

var player = new MediaPlayer(context);

    player.startup();

    player.attachView(document.querySelector("#videoplayer"));

    player.attachSource(url);

}

</script>
```

Change your <body> element to as following. (to make sure that the function setupVideo() will be executed once the page has fully loaded.)

```
<body onload="setupVideo()">
```

Set the video size using CSS

```
<style>

video {

    width: 80%;

    height: 80%;

}

</style>
```