

# COT 5930 DIGITAL IMAGE PROCESSING

DR. HARI KALVA

SPRING 2024

IAN MORGAN-GRAHAM

## MATLAB ASSIGNMENT 2 - VIDEO LABELING, OBJECT DETECTION, OBJECT TRACKING USING MATLAB VIDEOLABELING APPLICATION

### TABLE OF CONTENTS

#### SECTION 1: INTRODUCTION AND SET UP

#### SECTION 2: PROCEDURE

##### -- SCREEN CAPTURE RAW VIDEO DATA

##### -- CONSTRUCT INPUT AND TEST .MP4 FILES

##### -- MAKING GROUND TRUTH .MAT TABLES

#### SECTION 3: RESULTS

#### SECTION 4: CODE

#### SECTION 5: CODE: FUNCTIONS

#### SECTION 6: CONCLUSION

#### SECTION 7: REFERENCES

### SECTION 1: INTRODUCTION AND SET UP

[RETURN TO TABLE OF CONTENTS](#)

#### -- INTRODUCTION

The goal of this project is to label video files using the Matlab VidoLabeler application. In order to explore the video labeling concepts, video files were selected to cover different object detection/tracking situations. Video clips of tiger barb aquarium fish (*Puntigrus tetrazona*) were used to evaluate object detection/tracking of objects with random movement in an aqueous environment. Video clips of foil surfers and wind surfers were used to evaluate object detection/tracking of objects moving across a flat surface in mostly linear patterns. Video clips of snowboarders were used to evaluate object detection/tracking of objects moving down a gradient under the influence of gravity. Video clips of toy tops spinning in a parabolic basin were used to evaluate object detection/tracking of objects moving in a roughly circular pattern with occasional collisions.

#### -- SET UP

1.) Load needed files into the same contents folder and make a note of the directory path leading to this contents folder.

|                          |  |
|--------------------------|--|
| <b>Main Matlab file:</b> |  |
| -- Assignment2.mlx       |  |

| <b>Input .mp4 video files used with Matlab VideoLabeler application:</b>                                      | <b>Ground truth .mat files created with Matlab VideoLabeler application:</b>   |
|---|--|
| -- barb_tetra__1.mp4<br>-- foilsurf__1.mp4<br>-- snowboard__1.mp4<br>-- top_spin__1.mp4<br>-- windsurf__1.mp4 | -- gTruth_barb.mat<br>-- gTruth_foil_surf.mat<br>-- gTruth_snowboarder.mat<br>-- gTruth_top_spin.mat<br>-- gTruth_windsurfer.mat |
| <b>Test .mp4 video files that will be labeled using upon ground truth .mat files:</b>                         | <b>Output .avi files ** which are object detected and labeled versions of the test .mp4 files (these files will be created):</b> |
| -- barb_tetra__2.mp4<br>-- foilsurf__2.mp4<br>-- snowboard__2.mp4<br>-- top_spin__2.mp4<br>-- windsurf__2.mp4 | -- tiger_barb.avi<br>-- foilsurfer.avi<br>-- snowboarder.avi<br>-- top_spin.avi<br>-- windsurfer.avi                             |
| <b>Output folder containing created output .avi files **:</b>   |  |
| -- OUTPUT   |  |

Table 1. Files and folder loaded in contents folder

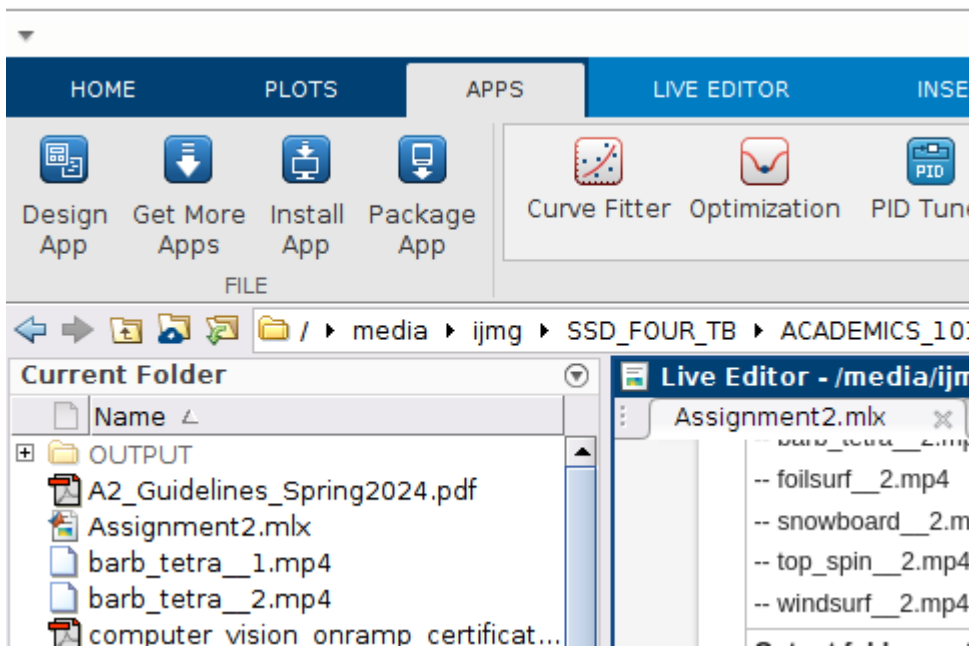
- 2.) Determine the present working directory with the "pwd" command in Matlab Command Window. Note the ans = 'result' where 'result' is the current working directory.
- 3.) In the Matlab Command Window, enter "cd" followed by the directory path leading to the contents folder noted in 1.). This will change the current working directory to the directory path leading to the contents folder.
- 4.) Verify the new present working directory change by retyping "pwd" in the Matlab Command Window.

## SECTION 2: PROCEDURE

## [RETURN TO TABLE OF CONTENTS](#)

All work was done on an Ubuntu 22.04.3 LTS operating system.

The Matlab VideoLabeler application was installed directly from within Matlab using the "Install App" button shown below.



## -- SCREEN CAPTURE RAW VIDEO DATA

## RETURN TO TABLE OF CONTENTS

Step 1 in the procedure was identifying suitable video clips on YouTube.com that met the various object detection/tracking situations mentioned in Section 1. Once suitable videos were located, an open source, linux based video screen capture tool, SimpleScreenRecorder version 0.3.11 (shown in figures 1 and 2 below), was used to capture roughly 20 to 30 seconds of raw video footage. The SimpleScreenRecorder software allows for selection of specific sections of the screen and output file format (here .mp4 format was used). Figure 2 shows the raw windsurfer screen video capture.



Figure 1. SimpleScreenRecorder start up dialog box

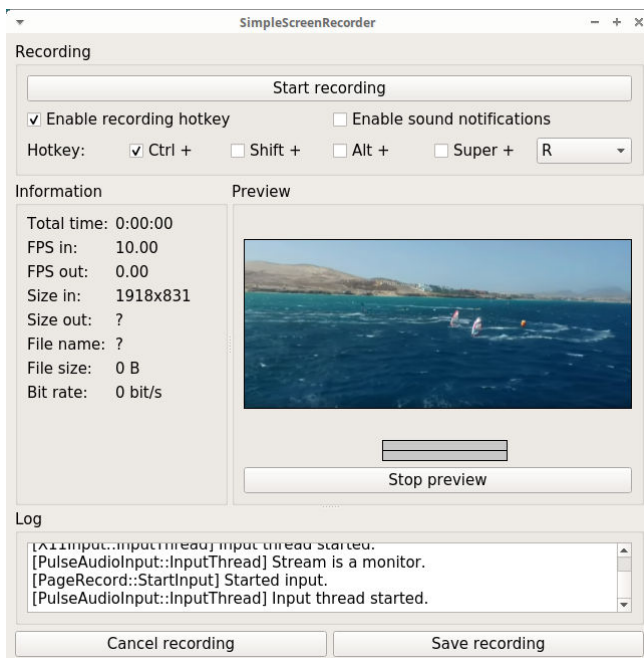


Figure 2. SimpleScreenRecorder video capture dialog box

## -- CONSTRUCT INPUT AND TEST .MP4 FILES

[RETURN TO TABLE OF CONTENTS](#)

Step 2 was breaking the raw screen captured video into two approximately 10 second videos with any audio sections removed. This was done using an open source, linux based video editing tool, Kdenlive version 21.12.3 (shown in figure 3 below). Figure 3 shows a roughly 60 second long raw .mp4 video that was converted into top\_spin\_\_1.mp4 and top\_spin\_\_2.mp4.

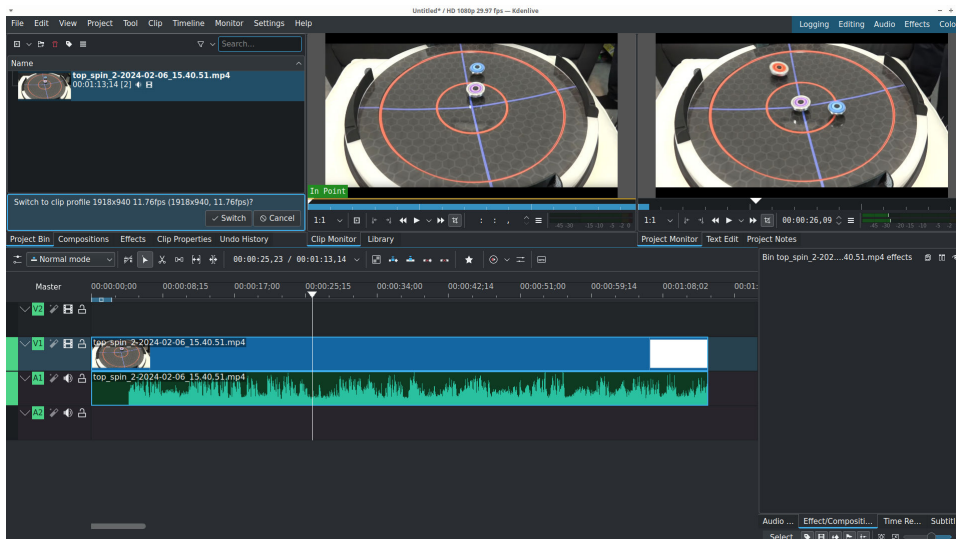


Figure 3. Kdenlive video editing screen.

Step 2 results in the construction of a set of input .mp4 video files that will be used to train the VideoLabeler application and and set of test .mp4 files that will be labeled using the ground truth .mat files generated by the VideoLabeler application.

| Input .mp4 video files used with Matlab VideoLabeler application:   | Test .mp4 video files that will be labeled based upon ground truth .mat files:                                |
|---|---|
| -- barb_tetra__1.mp4<br>-- foilsurf__1.mp4<br>-- snowboard__1.mp4<br>-- top_spin__1.mp4<br>-- windsurf__1.mp4 | -- barb_tetra__2.mp4<br>-- foilsurf__2.mp4<br>-- snowboard__2.mp4<br>-- top_spin__2.mp4<br>-- windsurf__2.mp4 |

Table 2. Input and Test .mp4 files

In step 3, each of the input .mp4 files was fed into the Matlab VideoLabeler application to produce a ground truth .mat file (figures 4 through 8).

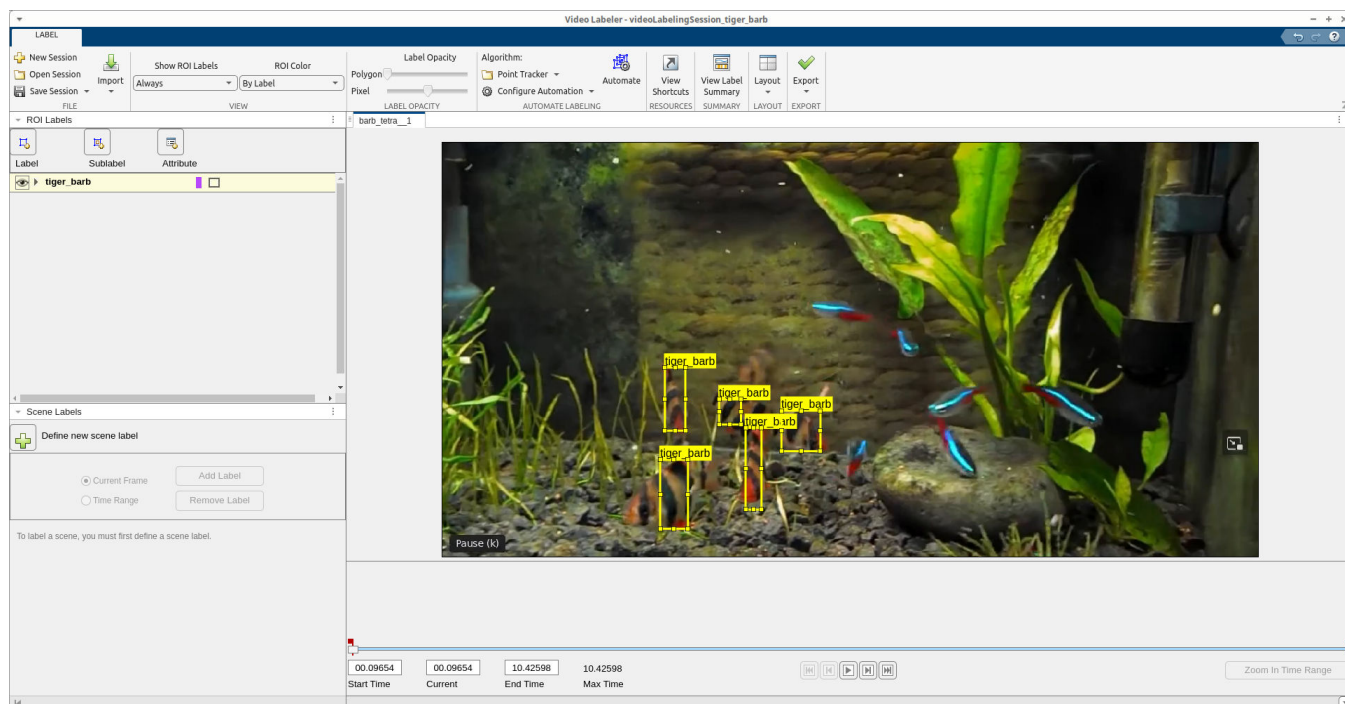


Figure 4. VideoLabeler labeling screen for barb\_tetra\_\_1.mp4 file.

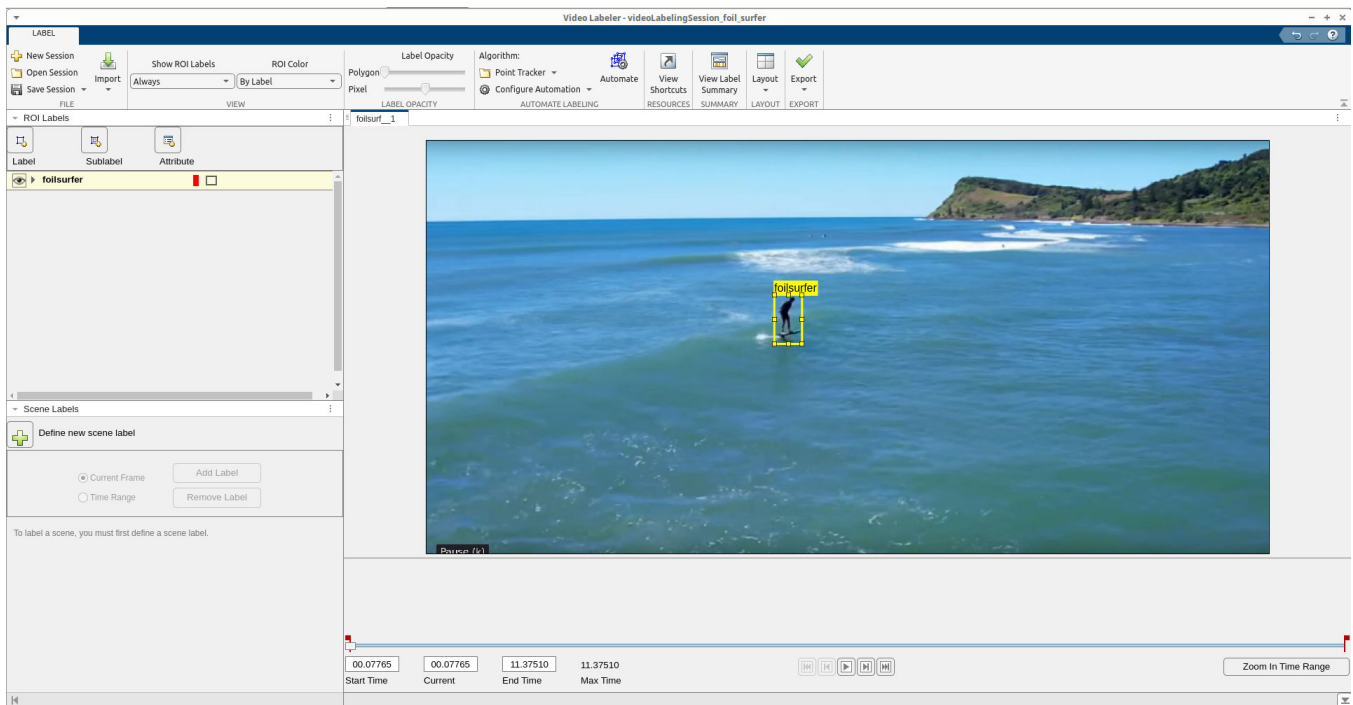


Figure 5. VideoLabeler labeling screen for foilsurf\_\_1.mp4 file.

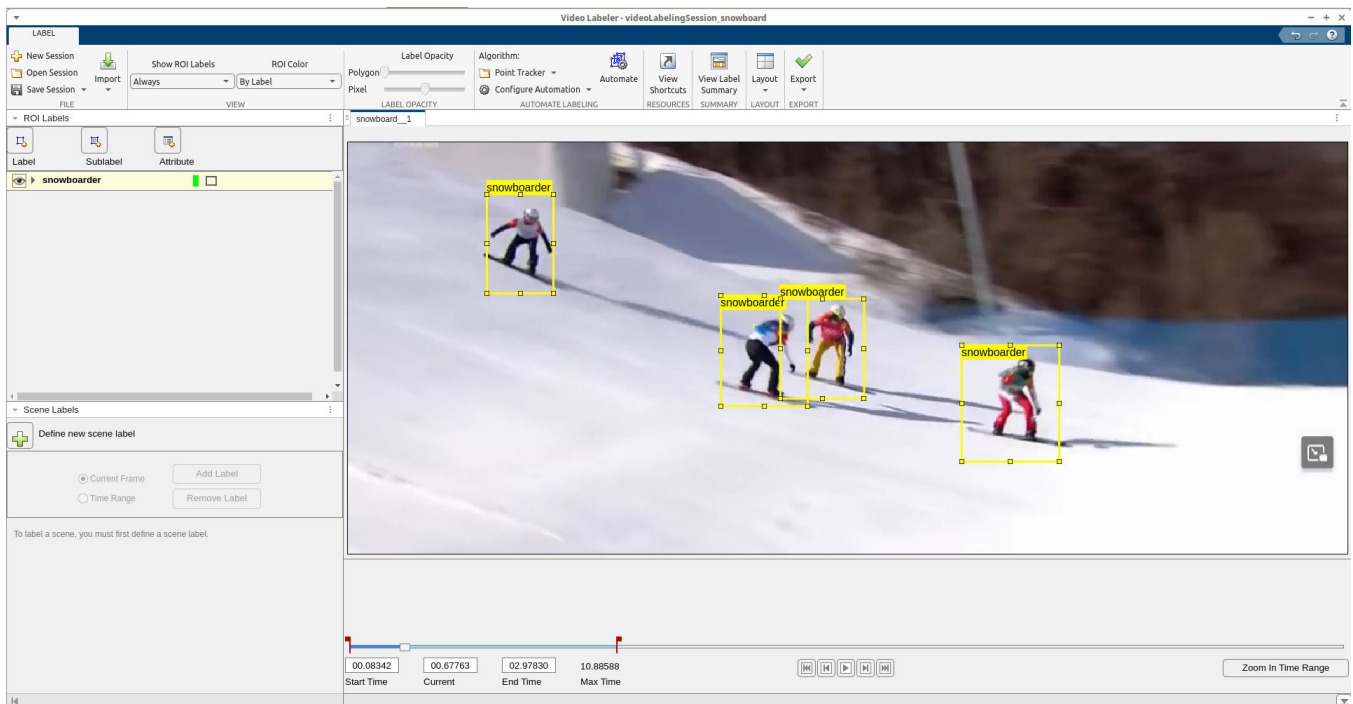


Figure 6. VideoLabeler labeling screen for snowboard\_\_1.mp4 file.



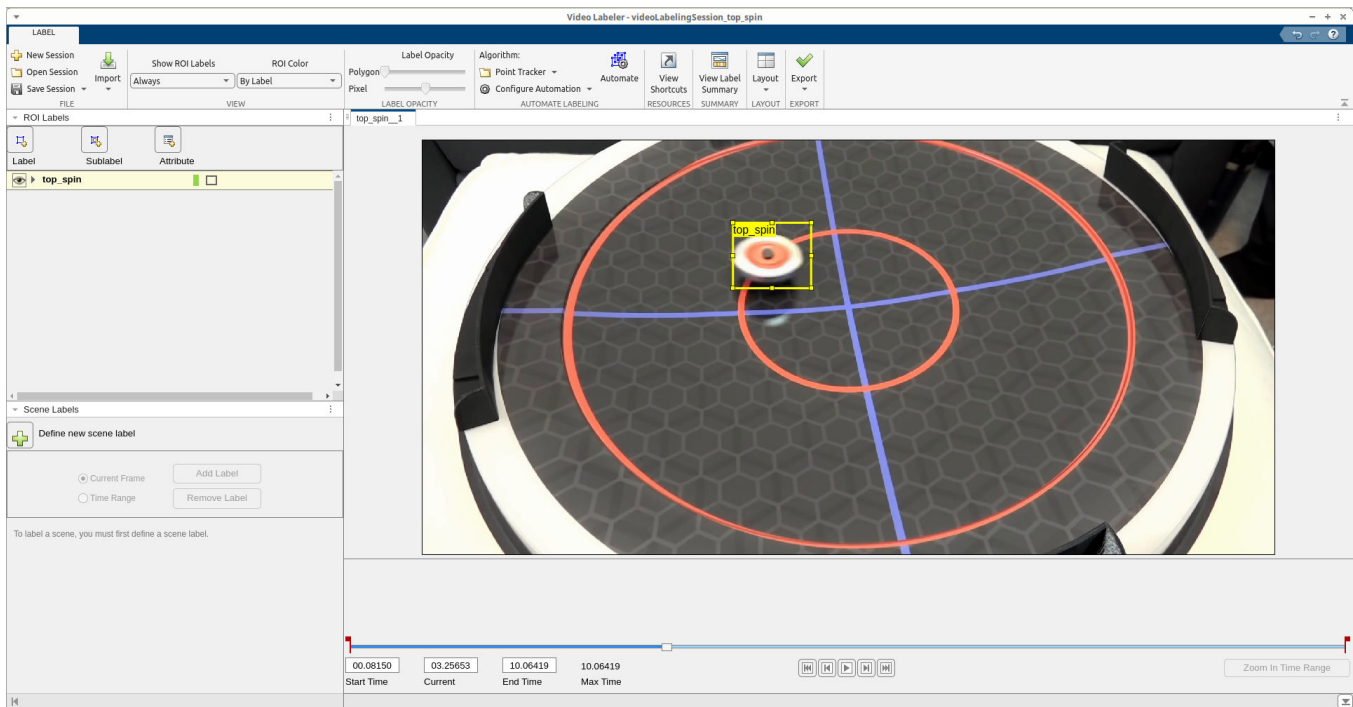


Figure 7. VideoLabeler labeling screen for top\_spin\_\_1.mp4 file.

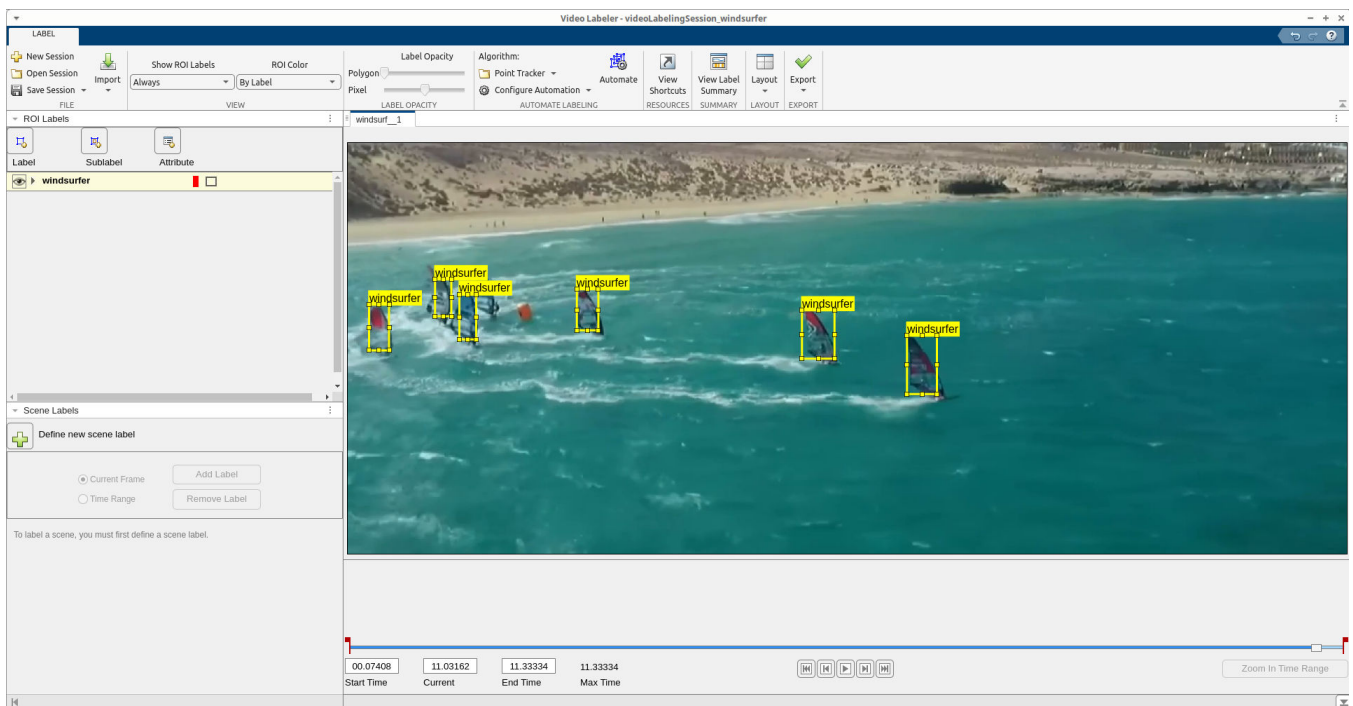


Figure 8. VideoLabeler labeling screen for windsurf\_\_1.mp4 file.

## -- MAKING GROUNG TRUTH .MAT TABLES

[RETURN TO TABLE OF CONTENTS](#)

Step 3 results in the construction of ground truth .mat files.

**Ground truth .mat files created with  
Matlab VideoLabeler application:**

-- gTruth\_barb.mat

```
-- gTruth_foil_surf.mat
-- gTruth_snowboarder.mat
-- gTruth_top_spin.mat
-- gTruth_windsurfer.mat
```

Table 3. Ground truth ,mat files

In the final step, the test .mp4 files are converted into object labeled/tracked .avi files and sent to the OUTPUT folder in the contents folder described in SECTION 1 –SET UP

In summary, each “\*\_1.mp4” file was manipulated using the Matlab VideoLabeler application to produce a corresponding “gTruth\_\*.mat” file. Each “gTruth\_\*.mat” file was then used to label a corresponding “\*\_2.mp4” file which was then converted into a final, labeled “\*.avi” file.

### SECTION 3: RESULTS

[RETURN TO TABLE OF CONTENTS](#)

The final labeled .avi files are contained in the OUTPUT folder. The code below can be used to generate each labeled .avi file separately.

**Output .avi files which are object detected and labeled versionsof the test .mp4 files:**

```
-- tiger_barb.avi
-- foilsurfer.avi
-- snowboarder.avi
-- top_spin.avi
-- windsurfer.avi
```

### SECTION 4: CODE

[RETURN TO TABLE OF CONTENTS](#)

Each cestion below generates a labeled .avi file and loads it into the OUTPUT folder

Push the "PERFORM ... LABELING" button to active the section of code and produce the final labeled .avi file in the OUTPUT folder. Before each labeled .avi file is generated the labeled .mp4 file is shown one frame at a time in a player that then closes.

The final .avi file can then be viewed using any local video player with the needed .avi codecs.

Generation of the final .avi file takes about 90 seconds depending on the local processor speed.

```
% clear previous outputs and workspace data
close all;
clear;
clc;
```



```
% load appropriate ground truth table (gTruth)
load('gTruth_foil_surf.mat');
% call function that will use ground truth table to label objects in a new
video file
% provide "video_annotate_function" function with :
%   ground truth table
%   label name within ground truth table
%   new video file to be labeled with label name
%   cutoff for detected matches between ground truth table and objects in
video file to
%       be labeled
video_annotate_function(gTruth, "foilsurfer", "foilsurf__2.mp4", 25)
```

Write images extracted for training to folder:

/media/ijmg/SSD\_FOUR\_TB/ACADEMICS\_101/a\_Florida Atlantic University/MSOE/COT 5930 Dig Image Process/As

Writing 56 images extracted from foilsurf\_\_1.mp4...Completed.

ACF Object Detector Training

The training will take 5 stages. The model size is 58x35.

Sample positive examples(~100% Completed)

Compute approximation coefficients...Completed.

Compute aggregated channel features...Completed.

-----

Stage 1:

Sample negative examples(~100% Completed)

Compute aggregated channel features...Completed.

Train classifier with 201 positive examples and 1005 negative examples...Completed.

The trained classifier has 46 weak learners.

-----

Stage 2:

Sample negative examples(~100% Completed)

Found 224 new negative examples for training.

Compute aggregated channel features...Completed.

Train classifier with 201 positive examples and 1005 negative examples...Completed.

The trained classifier has 146 weak learners.

-----

Stage 3:

Sample negative examples(~100% Completed)

Found 6 new negative examples for training.

Compute aggregated channel features...Completed.

Train classifier with 201 positive examples and 1005 negative examples...Completed.

The trained classifier has 153 weak learners.

-----

Stage 4:

Sample negative examples(~100% Completed)

Found 1 new negative examples for training.

Compute aggregated channel features...Completed.

Train classifier with 201 positive examples and 1005 negative examples...Completed.

The trained classifier has 153 weak learners.

-----

Stage 5:

Sample negative examples(~100% Completed)

Found 1 new negative examples for training.

Compute aggregated channel features...Completed.

Train classifier with 201 positive examples and 1005 negative examples...Completed.

The trained classifier has 153 weak learners.

-----

ACF object detector training is completed. Elapsed time is 70.8446 seconds.

workingDir =

'./OUTPUT/'

```

close all;
clear;
clc;
load('gTruth_top_spin.mat');
video_annotate_function(gTruth, "top_spin", "top_spin__2.mp4", 70)

```

Write images extracted for training to folder:

/media/ijmg/SSD\_FOUR\_TB/ACADEMICS\_101/a\_Florida Atlantic University/MSOE/COT 5930 Dig Image Process/As

Writing 32 images extracted from top\_spin\_\_1.mp4...Completed.

ACF Object Detector Training

The training will take 5 stages. The model size is 111x148.

Sample positive examples(~100% Completed)

Compute approximation coefficients...Completed.

Compute aggregated channel features...Completed.

-----

Stage 1:

Sample negative examples(~100% Completed)

Compute aggregated channel features...Completed.

Train classifier with 39 positive examples and 195 negative examples...Completed.

The trained classifier has 19 weak learners.

-----

Stage 2:

Sample negative examples(~100% Completed)

Found 195 new negative examples for training.

Compute aggregated channel features...Completed.

Train classifier with 39 positive examples and 195 negative examples...Completed.

The trained classifier has 19 weak learners.

-----

Stage 3:

Sample negative examples(~100% Completed)

Found 195 new negative examples for training.

Compute aggregated channel features...Completed.

Train classifier with 39 positive examples and 195 negative examples...Completed.

The trained classifier has 19 weak learners.

-----

Stage 4:

Sample negative examples(~100% Completed)

Found 195 new negative examples for training.

Compute aggregated channel features...Completed.

Train classifier with 39 positive examples and 195 negative examples...Completed.

The trained classifier has 23 weak learners.

-----

Stage 5:

Sample negative examples(~100% Completed)

Found 155 new negative examples for training.

Compute aggregated channel features...Completed.

Train classifier with 39 positive examples and 195 negative examples...Completed.

The trained classifier has 88 weak learners.

-----

ACF object detector training is completed. Elapsed time is 52.6716 seconds.

workingDir =

'./OUTPUT/'

```

close all;
clear;
clc;
load('gTruth_windsurfer.mat');

```

```
video_annotate_function(gTruth, "windsurfer", "windsurf__2.mp4", 35)
```

Write images extracted for training to folder:

/media/ijmg/SSD\_FOUR\_TB/ACADEMICS\_101/a\_Florida Atlantic University/MSOE/COT 5930 Dig Image Process/As

Writing 58 images extracted from windsurf\_\_1.mp4...Completed.

ACF Object Detector Training

The training will take 5 stages. The model size is 37x21.

Sample positive examples(~100% Completed)

Compute approximation coefficients...Completed.

Compute aggregated channel features...Completed.

-----  
Stage 1:

Sample negative examples(~100% Completed)

Compute aggregated channel features...Completed.

Train classifier with 630 positive examples and 1426 negative examples...Completed.

The trained classifier has 157 weak learners.

-----  
Stage 2:

Sample negative examples(~100% Completed)

Found 288 new negative examples for training.

Compute aggregated channel features...Completed.

Train classifier with 630 positive examples and 1714 negative examples...Completed.

The trained classifier has 256 weak learners.

-----  
Stage 3:

Sample negative examples(~100% Completed)

Found 219 new negative examples for training.

Compute aggregated channel features...Completed.

Train classifier with 630 positive examples and 1933 negative examples...Completed.

The trained classifier has 512 weak learners.

-----  
Stage 4:

Sample negative examples(~100% Completed)

Found 338 new negative examples for training.

Compute aggregated channel features...Completed.

Train classifier with 630 positive examples and 2271 negative examples...Completed.

The trained classifier has 1024 weak learners.

-----  
Stage 5:

Sample negative examples(~100% Completed)

Found 173 new negative examples for training.

Compute aggregated channel features...Completed.

Train classifier with 630 positive examples and 2444 negative examples...Completed.

The trained classifier has 2048 weak learners.

-----  
ACF object detector training is completed. Elapsed time is 107.1218 seconds.

workingDir =

'./OUTPUT/'

```
close all;
clear;
clc;
load('gTruth_barb.mat');
video_annotate_function(gTruth, "tiger_barb", "barb_tetra__2.mp4", 14)
```

Write images extracted for training to folder:

/media/ijmg/SSD\_FOUR\_TB/ACADEMICS\_101/a\_Florida Atlantic University/MSOE/COT 5930 Dig Image Process/As

Writing 18 images extracted from barb\_tetra\_\_1.mp4...Completed.

```

ACF Object Detector Training
The training will take 5 stages. The model size is 53x50.
Sample positive examples(~100% Completed)
Compute approximation coefficients...Completed.
Compute aggregated channel features...Completed.
-----
Stage 1:
Sample negative examples(~100% Completed)
Compute aggregated channel features...Completed.
Train classifier with 63 positive examples and 315 negative examples...Completed.
The trained classifier has 256 weak learners.
-----
Stage 2:
Sample negative examples(~100% Completed)
Found 81 new negative examples for training.
Compute aggregated channel features...Completed.
Train classifier with 63 positive examples and 315 negative examples...Completed.
The trained classifier has 256 weak learners.
-----
Stage 3:
Sample negative examples(~100% Completed)
Found 81 new negative examples for training.
Compute aggregated channel features...Completed.
Train classifier with 63 positive examples and 315 negative examples...Completed.
The trained classifier has 512 weak learners.
-----
Stage 4:
Sample negative examples(~100% Completed)
Found 64 new negative examples for training.
Compute aggregated channel features...Completed.
Train classifier with 63 positive examples and 315 negative examples...Completed.
The trained classifier has 1024 weak learners.
-----
Stage 5:
Sample negative examples(~100% Completed)
Found 77 new negative examples for training.
Compute aggregated channel features...Completed.
Train classifier with 63 positive examples and 315 negative examples...Completed.
The trained classifier has 2048 weak learners.
-----
ACF object detector training is completed. Elapsed time is 55.6936 seconds.
workingDir =
'./OUTPUT/'

```

```

close all;
clear;
clc;
load('gTruth_snowboard.mat');
video_annotate_function(gTruth, "snowboarder", "snowboard__2.mp4", 14)

```

Write images extracted for training to folder:  
/media/ijmg/SSD\_FOUR\_TB/ACADEMICS\_101/a\_Florida Atlantic University/MSOE/COT 5930 Dig Image Process/As

```

Writing 14 images extracted from snowboard__1.mp4...Completed.
ACF Object Detector Training
The training will take 5 stages. The model size is 34x27.
Sample positive examples(~100% Completed)
Compute approximation coefficients...Completed.
Compute aggregated channel features...Completed.
-----

```

```

Stage 1:
Sample negative examples(~100% Completed)
Compute aggregated channel features...Completed.
Train classifier with 54 positive examples and 270 negative examples...Completed.
The trained classifier has 256 weak learners.
-----
Stage 2:
Sample negative examples(~100% Completed)
Found 140 new negative examples for training.
Compute aggregated channel features...Completed.
Train classifier with 54 positive examples and 270 negative examples...Completed.
The trained classifier has 256 weak learners.
-----
Stage 3:
Sample negative examples(~100% Completed)
Found 39 new negative examples for training.
Compute aggregated channel features...Completed.
Train classifier with 54 positive examples and 270 negative examples...Completed.
The trained classifier has 512 weak learners.
-----
Stage 4:
Sample negative examples(~100% Completed)
Found 13 new negative examples for training.
Compute aggregated channel features...Completed.
Train classifier with 54 positive examples and 270 negative examples...Completed.
The trained classifier has 1024 weak learners.
-----
Stage 5:
Sample negative examples(~100% Completed)
Found 18 new negative examples for training.
Compute aggregated channel features...Completed.
Train classifier with 54 positive examples and 270 negative examples...Completed.
The trained classifier has 2048 weak learners.
-----
ACF object detector training is completed. Elapsed time is 20.2911 seconds.
workingDir =
'./OUTPUT/'

```

## SECTION 5: CODE: FUNCTIONS

## [RETURN TO TABLE OF CONTENTS](#)

```

function [] = video_annotate_function(ground_truth, truth_label,
video_to_label, score_cutoff)
    % load ground truth table
    truth = selectLabels(ground_truth, truth_label);
    % extract training data for detector from truth table
    trainingData = objectDetectorTrainingData(truth);
    % construct detector using training data
    detector = trainACFObjectDetector(trainingData, 'Numstages', 5);
    % load input video file to be labeled into video reader
    loaded_video = VideoReader(video_to_label);
    % create video player object
    video_player = vision.DeployableVideoPlayer;

    % relative path to output directory folder, "OUTPUT"
    workingDir = './OUTPUT/'
    % declare output .avi video file and prepare to add video frames to it
    to build

```

```

% final video
outputVideo = VideoWriter(fullfile(workingDir, truth_label + ".avi"));
outputVideo.FrameRate = loaded_video.FrameRate;
open(outputVideo)

while hasFrame(loaded_video)
    % load input video frames one at a time
    frame = readFrame(loaded_video);
    % use detector and ground truth to locate possible matching objects
    [bbox,score] = detect(detector,frame);
    % assign probability of match rank to possible matching objects
    % retain only those match rank probabilities above score cut off
    % provided
    bbox = bbox(score > score_cutoff,:);
    score = score(score > score_cutoff,:);
    [selectedBbox,selectedScore] = selectStrongestBbox(bbox,score,
OverlapThreshold=0.1);

    numBoxes = size(selectedBbox,1);
    % annotate possible matching objects
    str = "OBJECTS DETECTED : " + numBoxes;
    img = insertObjectAnnotation(frame,"rectangle",
selectedBbox,truth_label + " : " + selectedScore);
    img = insertText(img,[250 550],str,TextColor=[1 1 0]);

    % display each annotated video frame in video player object
    step(video_player, img)
    % write each annotated video frame to declared output .avi video file
    writeVideo(outputVideo,img)
end
close(outputVideo)
end

```

## SECTION 6: CONCLUSION

## [RETURN TO TABLE OF CONTENTS](#)

Although the goals of the project were met and videos were labeled, other challenges were left unexplored.

Language -- A search of indeed.com and other job boards suggests the need to learn both Matlab and Python for image/video processing. In light of this, given additional time, this project would have been replicated using Python.

Type of object detection/tracking -- Videos requiring the detection and tracking of projectile objects were not fully explored. This exploration would be a worthwhile investment since most sports involve a projectile (a ball) and most warfare situations would also focus on a projectile (a moving enemy vessel or a launched enemy weapon).

Refinement -- The score\_cutoff variable had a great affect on the number of matching objects detected. Its impact was not fully explored.



Ground Truth Algorithm - Only the Point Tracker algorithm was explored to form ground truth tables. The Temporal Interpolator algorithm was not fully explored. The Temporal Interpolator algorithm might be a good choice in future projectile object detection and tracking.

## SECTION 7: REFERENCES

## RETURN TO TABLE OF CONTENTS

<https://matlabacademy.mathworks.com/details/computer-vision-onramp/orcv>



[https://www.youtube.com/watch?v=ow\\_B\\_30WU1s&t=727s&ab\\_channel=MATLAB](https://www.youtube.com/watch?v=ow_B_30WU1s&t=727s&ab_channel=MATLAB)

[https://www.youtube.com/watch?v=UnXDQmjYvDk&t=414s&ab\\_channel=MATLAB](https://www.youtube.com/watch?v=UnXDQmjYvDk&t=414s&ab_channel=MATLAB)

<https://www.mathworks.com/help/vision/ug/get-started-with-the-video-labeler.html>

[https://www.mathworks.com/help/matlab/import\\_export/convert-between-image-sequences-and-video.html](https://www.mathworks.com/help/matlab/import_export/convert-between-image-sequences-and-video.html)