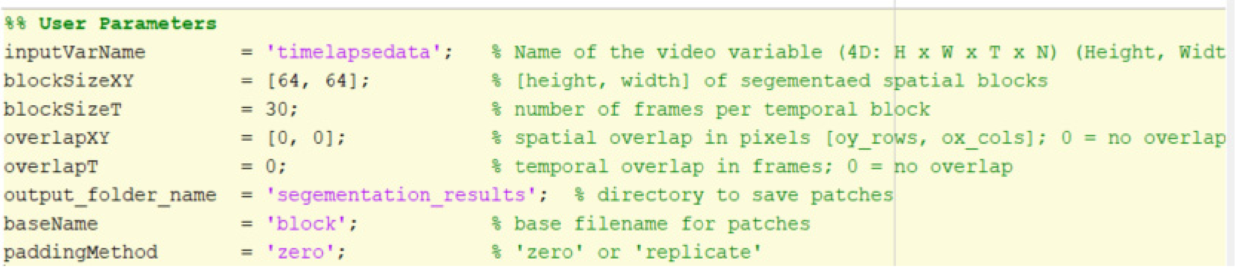
Example workflow – 1 (process large video through segmentations)

1. Run the MATLAB function “Step1_segementation.m”

a. Select a .mat video file you want to segment. The data must be a 4-D array [H × W × T × N] (Height, Width, Time, Number of videos).

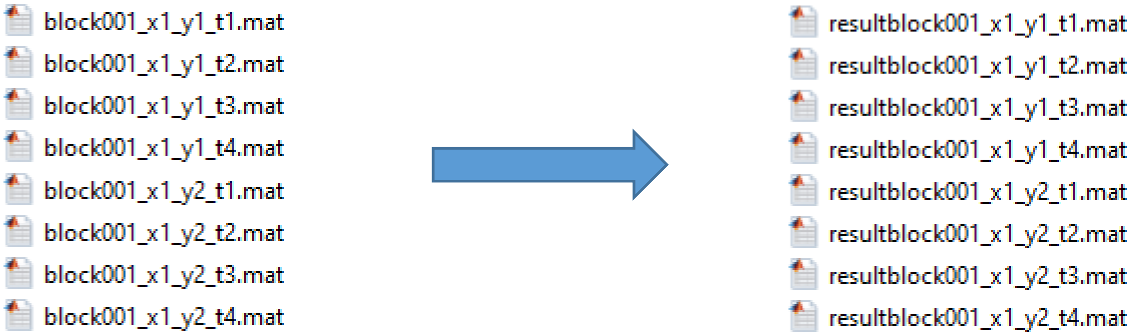
b. Parameters including the saved variable name, block size and stride size can be changed at the beginning of the script (see below).



```
%% User Parameters
inputVarName        = 'timelapsedata';   % Name of the video variable (4D: H x W x T x N) (Height, Widt
blockSizeXY         = [64, 64];          % [height, width] of segementaed spatial blocks
blockSizeT          = 30;                % number of frames per temporal block
overlapXY           = [0, 0];            % spatial overlap in pixels [oy_rows, ox_cols]; 0 = no overlap
overlapT            = 0;                 % temporal overlap in frames; 0 = no overlap
output_folder_name  = 'segementation_results';  % directory to save patches
baseName            = 'block';           % base filename for patches
paddingMethod       = 'zero';            % 'zero' or 'replicate'
```

2. Run “Step2_SPTnet_inference.py”

Use SPTnet estimate each blocks generated in step 1



block001_x1_y1_t1.mat
block001_x1_y1_t2.mat
block001_x1_y1_t3.mat
block001_x1_y1_t4.mat
block001_x1_y2_t1.mat
block001_x1_y2_t2.mat
block001_x1_y2_t3.mat
block001_x1_y2_t4.mat

resultblock001_x1_y1_t1.mat
resultblock001_x1_y1_t2.mat
resultblock001_x1_y1_t3.mat
resultblock001_x1_y1_t4.mat
resultblock001_x1_y2_t1.mat
resultblock001_x1_y2_t2.mat
resultblock001_x1_y2_t3.mat
resultblock001_x1_y2_t4.mat

3. Run the MATLAB function “Step3_stitch_and_display.m”

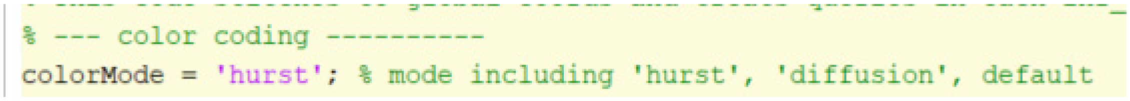
a. Select all the raw blocks, and their corresponding inference results.

b. Enter the stride values you used when segmenting the original video.

c. Stitched result will be saved in file “stitched_results.mat“ and the variable called “DetG” with 11 columns, representing:

1=tg (frame, local), 2=xg (local x), 3=yg (local y), 4=obj (detection probablity), 5=q (query id, local), 6=x0 (x shift), 7=y0 (y shift), 8=t0 (time shift), 9=D (generalized diffusion coefficient), 10=H (Hurst), 11=inference_id.

d. The output video can be color coded differently (default is Hurst exponent)



```
% --- color coding ----------
colorMode = 'hurst'; % mode including 'hurst', 'diffusion', default
```

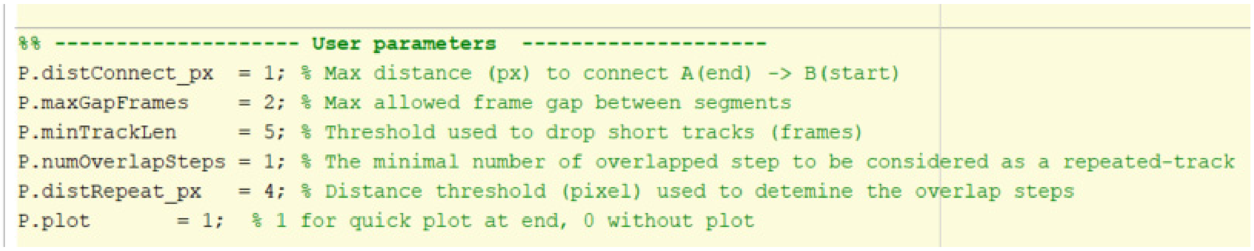
4. Run the MATLAB function “Step4_chunk_connection_and_repeat_reduction.m”

(Optional)

a. load the “stitched_results.mat” generated from step 3

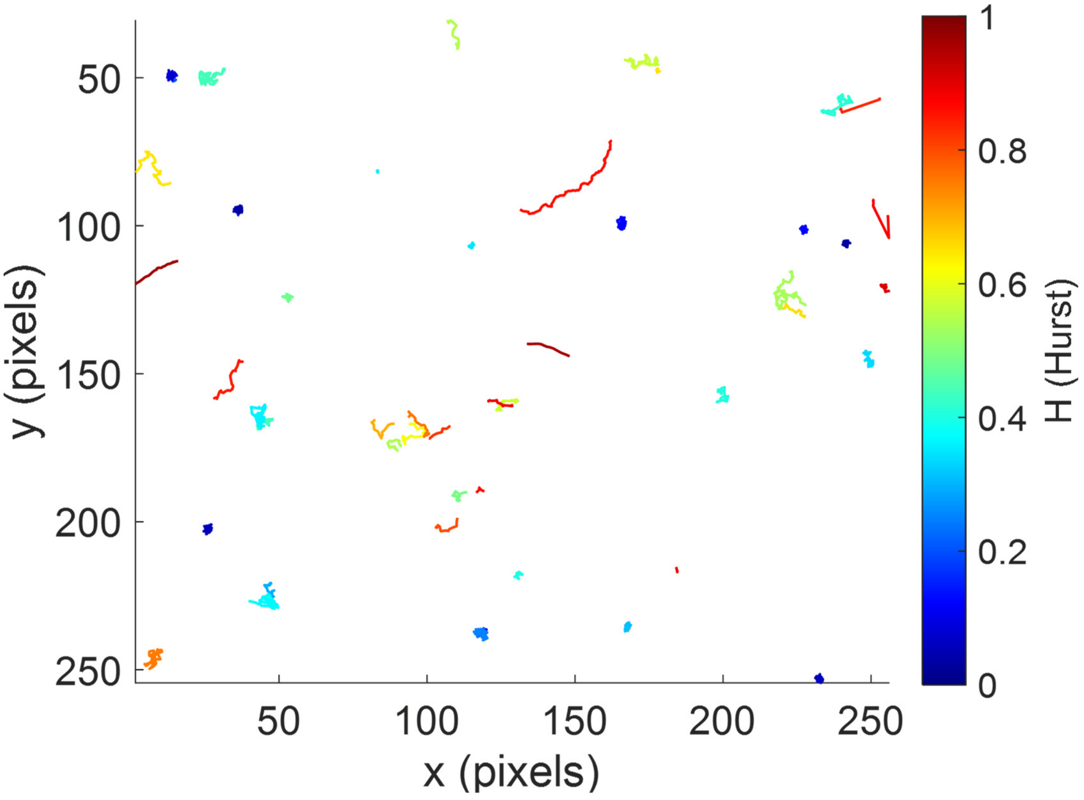
b. Run the script to remove duplicate tracks in overlapping regions and connect trajectories across temporal blocks.

c. This section can be modified and improved as needed, the connection of tracks between different blocks is simply based on distance threshold. Parameters can be changed at the beginning of the script (see blow).



```
%% -------------------- User parameters  --------------------
P.distConnect_px  = 1; % Max distance (px) to connect A(end) -> B(start)
P.maxGapFrames    = 2; % Max allowed frame gap between segments
P.minTrackLen     = 5; % Threshold used to drop short tracks (frames)
P.numOverlapSteps = 1; % The minimal number of overlapped step to be considered as a repeated-track
P.distRepeat_px   = 4; % Distance threshold (pixel) used to detemine the overlap steps
P.plot       = 1;  % 1 for quick plot at end, 0 without plot
```

d. Output is saved as “stitched_results_connected_remove_repeat.mat” and a .tiff plot will be saved as “connected_tracks_h.tif” (see below)


y (pixels)
x (pixels)
H (Hurst)