Final project solution

**Table of Contents**

[Section 1: Enhancing the video 1](#_Toc27704)

[Section 2.1: Isolating the cars with background subtraction 2](#_Toc27705)

[Background subtraction preparation 2](#_Toc27706)

[Sections 2.2 & 3: Segmenting cars and Calculating region properties 2](#_Toc27707)

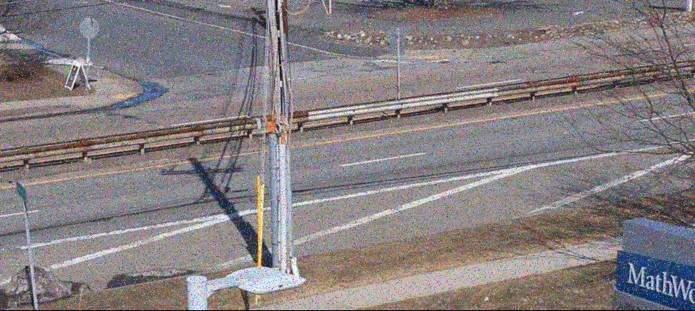
[Segmentation function 7](#_Toc27708)

# 

# Section 1: Enhancing the video

Enhance the video by removing the noise, convert the video to grayscale, then save the result as a new video file.

For an example frame, go from the noisy image (left) to the grayscale image (right).



Initialize the video reader and writer objects.

vid = VideoReader("RoadTraffic.mp4");

vidWr = VideoWriter("RoadTrafficFiltered","MPEG-4");

vidWr.FrameRate = vid.FrameRate;

open(vidWr);

Loop through every frame, apply the filter, convert to grayscale, and write the result to a new video.

|  |
| --- |
| while hasFrame(vid) %Read a frame  frame = readFrame(vid);  % Remove noise using a 2D median filter frame(:,:,1) = medfilt2(frame(:,:,1)); frame(:,:,2) = medfilt2(frame(:,:,2)); frame(:,:,3) = medfilt2(frame(:,:,3));  % Convert to grayscale frame = im2gray(frame); |
| % Write frame to new video writeVideo(vidWr,frame);  end close(vidWr); |
|  |

# Section 2.1: Isolating the cars with background subtraction

Isolate the cars using background subtraction.

The end goal of section 2 for an example frame, is to go from the grayscale image (left) to the BW mask image (right).



This is done using background subtraction to first isolate the moving cars from the stationary background.

## Background subtraction preparation

First, create a background image with no cars from the first frame.

vid = VideoReader("RoadTrafficFiltered.mp4"); backImg = readFrame(vid);

backImg = im2gray(backImg);

backImg = im2double(backImg);

# Sections 2.2 & 3: Segmenting cars and Calculating region properties

Segment the cars and create a table that contains a row for each frame of the video and a column for the following three properties: number of regions, mean region size, and total region size.

Initialize the video reader object.

vid = VideoReader("RoadTrafficFiltered.mp4");

Initialize the table variables.

NumberRegions = [];

MeanRegionSize = [];

TotalRegionSize = [];

Loop through every frame and collect region properties.

|  |
| --- |
| while hasFrame(vid)  % Read a frame  frame = readFrame(vid); frame = im2gray(frame); frame = im2double(frame);    % Perform background subtraction subImg = abs(frame - backImg);  % Segment cars from subtraction result mask = segmentCars(subImg);  % Filter out small regions  mask = bwpropfilt(mask,'Area',[4000 inf]);  % Collect region properties  props = regionprops("table", mask, "Area"); numReg = height(props);  meanRegS = mean(props.Area);  totRegS = sum(props.Area);  % Append results to arrays  NumberRegions = [NumberRegions; numReg];  MeanRegionSize = [MeanRegionSize; meanRegS];  TotalRegionSize = [TotalRegionSize; totRegS]; end |

Convert arrays to a table variable.

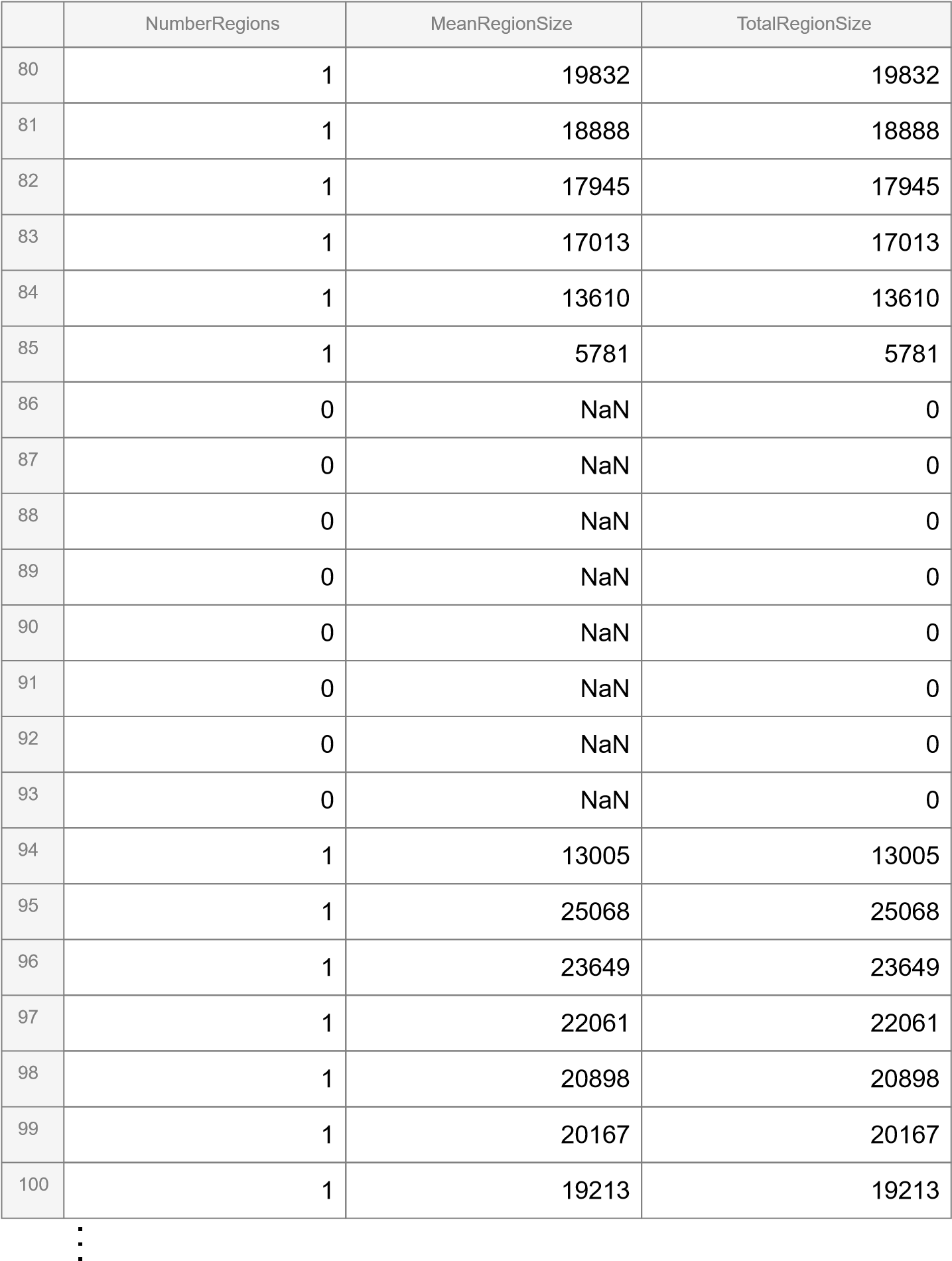
carData = table(NumberRegions, MeanRegionSize, TotalRegionSize)

carData = 240×3 table

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | NumberRegions |  | MeanRegionSize | TotalRegionSize |
| 1 |  | 0 | NaN | 0 |
| 2 |  | 0 | NaN | 0 |
| 3 |  | 0 | NaN | 0 |
| 4 |  | 0 | NaN | 0 |
| 5 |  | 0 | NaN | 0 |
| 6 |  | 1 | 6277 | 6277 |
| 7 |  | 2 | 14378 | 28756 |
| 8 |  | 2 | 1.9762e+04 | 39523 |
| 9 |  | 2 | 1.8682e+04 | 37365 |
| 10 |  | 2 | 17997 | 35994 |
| 11 |  | 2 | 1.8166e+04 | 36333 |
| 12 |  | 2 | 17993 | 35986 |
| 13 |  | 2 | 17383 | 34766 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | NumberRegions |  | MeanRegionSize | TotalRegionSize |
| 14 |  | 2 | 16644 | 33288 |
| 15 |  | 2 | 1.6402e+04 | 32805 |
| 16 |  | 3 | 1.5725e+04 | 47176 |
| 17 |  | 3 | 1.9769e+04 | 59308 |
| 18 |  | 2 | 24630 | 49260 |
| 19 |  | 2 | 22867 | 45734 |
| 20 |  | 2 | 1.8030e+04 | 36059 |
| 21 |  | 1 | 27163 | 27163 |
| 22 |  | 1 | 26315 | 26315 |
| 23 |  | 1 | 25080 | 25080 |
| 24 |  | 1 | 23729 | 23729 |
| 25 |  | 1 | 21947 | 21947 |
| 26 |  | 1 | 16165 | 16165 |
| 27 |  | 1 | 7417 | 7417 |
| 28 |  | 0 | NaN | 0 |
| 29 |  | 0 | NaN | 0 |
| 30 |  | 0 | NaN | 0 |
| 31 |  | 0 | NaN | 0 |
| 32 |  | 0 | NaN | 0 |
| 33 |  | 0 | NaN | 0 |
| 34 |  | 0 | NaN | 0 |
| 35 |  | 0 | NaN | 0 |
| 36 |  | 0 | NaN | 0 |
| 37 |  | 0 | NaN | 0 |
| 38 |  | 0 | NaN | 0 |
| 39 |  | 0 | NaN | 0 |
| 40 |  | 0 | NaN | 0 |
| 41 |  | 0 | NaN | 0 |
| 42 |  | 0 | NaN | 0 |
| 43 |  | 0 | NaN | 0 |
| 44 |  | 0 | NaN | 0 |
| 45 |  | 0 | NaN | 0 |
| 46 |  | 0 | NaN | 0 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | NumberRegions |  | MeanRegionSize | TotalRegionSize |
| 47 |  | 0 | NaN | 0 |
| 48 |  | 0 | NaN | 0 |
| 49 |  | 0 | NaN | 0 |
| 50 |  | 0 | NaN | 0 |
| 51 |  | 0 | NaN | 0 |
| 52 |  | 0 | NaN | 0 |
| 53 |  | 0 | NaN | 0 |
| 54 |  | 0 | NaN | 0 |
| 55 |  | 0 | NaN | 0 |
| 56 |  | 0 | NaN | 0 |
| 57 |  | 0 | NaN | 0 |
| 58 |  | 0 | NaN | 0 |
| 59 |  | 0 | NaN | 0 |
| 60 |  | 0 | NaN | 0 |
| 61 |  | 0 | NaN | 0 |
| 62 |  | 1 | 7930 | 7930 |
| 63 |  | 1 | 8557 | 8557 |
| 64 |  | 1 | 9166 | 9166 |
| 65 |  | 1 | 9498 | 9498 |
| 66 |  | 1 | 9880 | 9880 |
| 67 |  | 1 | 10215 | 10215 |
| 68 |  | 1 | 10155 | 10155 |
| 69 |  | 1 | 10635 | 10635 |
| 70 |  | 1 | 11282 | 11282 |
| 71 |  | 1 | 12326 | 12326 |
| 72 |  | 1 | 7925 | 7925 |
| 73 |  | 0 | NaN | 0 |
| 74 |  | 1 | 12756 | 12756 |
| 75 |  | 1 | 26270 | 26270 |
| 76 |  | 1 | 24131 | 24131 |
| 77 |  | 1 | 22916 | 22916 |
| 78 |  | 1 | 21659 | 21659 |
| 79 |  | 1 | 20915 | 20915 |



# Segmentation function

function [BW,maskedImage] = segmentCars(X)

|  |
| --- |
| % Threshold image - manual threshold  BW = X > 0.1;  % Close mask with disk  radius = 3; decomposition = 0;  se = strel('disk', radius, decomposition);  BW = imclose(BW, se);  % Fill holes  BW = imfill(BW, 'holes');  % Open mask with disk  radius = 5; decomposition = 0;  se = strel('disk', radius, decomposition);  BW = imopen(BW, se);  % Close mask with rectangle dimensions = [1 39];  se = strel('rectangle', dimensions);  BW = imclose(BW, se);  % Fill holes  BW = imfill(BW, 'holes');  % Create masked image. maskedImage = X; maskedImage(~BW) = 0;  end |