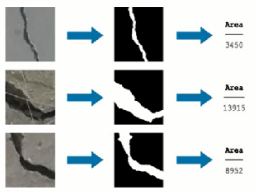
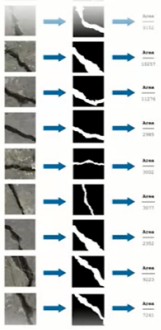
Using the Image Batch Processor App





Automation Steps

1. Test and refine the segmentation function

2. Combine the segmentation and analysis functions

```
img = imread("00001.jpg");
Bw = segmentCrackVersion1(img);
imshow(Bw)
```

segmentImageFinal + findProperties

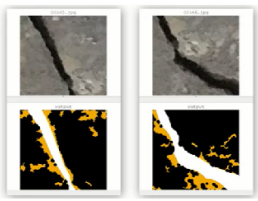
CREATE AND REFINE PROCESSING STEPS

EVALUATE ON A BATCH OF IMAGES

EXPORT RESULTS

Structure
A container for multiple variables

Iterative Process



Batch Processing with Image Datastores

```
imagesFolder = fullfile(toolboxdir("nnet"),"nndemos","nndatasets","DigitDataset");
imds = imageDatastore(imagesFolder,"IncludeSubfolders",true)
```

```
idx = 4242;
img = readimage(imds,idx);
imshow(img)
```

```
imds = imageDatastore(imagesFolder,"IncludeSubfolders",true,"LabelSource","foldernames")
idx = 6400 ;
[img, imgInfo] = readimage(imds,idx);
imgInfo
```

```
imgInfo = struct with fields:
    Filename: 'C:\Program Files\MATLAB\R2021b\toolbox\nnet\nndemos\nndatasets\DigitDataset\6\image5400.png'
    FileSize: 421
       Label: 6
```

```
imshow(img)
```

Detecting Moving Objects

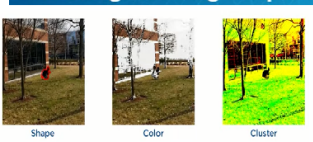
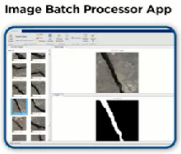
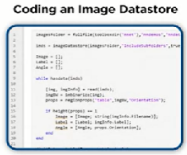


Image Batch Processor App

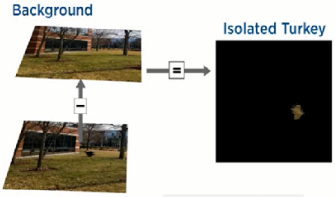


Coding an Image Datastore



```
Image = [];
Label = [];
Angle = [];

while hasdata(imds)

    [img, imgInfo] = read(imds);
    imgBW = imbinarize(img);
    props = regionprops("table",imgBW,"Orientation");

    if height(props) == 1
        Image = [Image; string(imgInfo.Filename)];
        Label = [Label; imgInfo.Label];
        Angle = [Angle; props.Orientation];
    end
end

digitData = table(Image, Label, Angle)
```

Background



Isolated Turkey

How do you get a background image to calculate this difference?

Whole Image

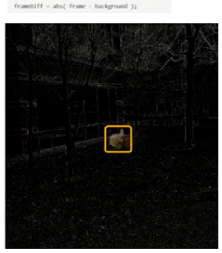
```
frame = read(v,290);
frame = im2double(frame);
background = aveFrame;
```

An average of all frames will *approximate* the background

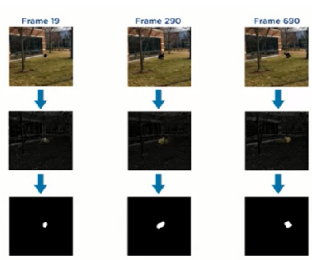
```
v = VideoReader("Turkey Video.mp4");

frameSum = im2double(read(v,1));
for idx = 2:v.NumFrames
    frameSum = frameSum + im2double(read(v,idx));
end
aveFrame = frameSum/v.NumFrames;
```

```
imshow(aveFrame)
```

```
bw = segmentTurkey(frameDiff);
imshow(bw)
```



Artifacts



```
props = regionprops("table",bw,"BoundingBox");
turkeyBoxed = insertShape(frame,"Rectangle",props.BoundingBox,"LineWidth",3,"Color","red");
imshow(turkeyBoxed)
```

```
v2 = VideoWriter("Turkey Boxed.mp4","MPEG-4");
open(v2);

for idx = 1:v.NumFrames

frame = read(v,idx);

frame = im2double(frame);

background = aveFrame;

frameDiff = abs( frame - background );

bw = segmentTurkey(frameDiff);

props = regionprops("table",bw,"BoundingBox");

turkeyBoxed = insertShape(frame,"Rectangle",props.BoundingBox,"LineWidth",3,"Color","red");

writeVideo(v2,turkeyBoxed);
end
close(v2)
```

In Summary:

- Convert images to double for calculations
- Extract or create a background frame
- Calculate the absolute value difference | frame – background |
- Segment the results

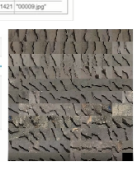
- Categorize cracks as severe vs. mild risk
- Investigate outliers

```
load CrackData.mat
crackData
```

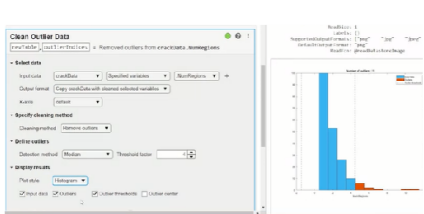
	NumRegions	Area	MaxWidth	fileName
1	2	2461	17.8885	"00001.jpg"
2	2	2125	17.8885	"00002.jpg"
3	2	2417	16	"00003.jpg"
4	1	2438	14.1421	"00004.jpg"
5	4	1479	18.4391	"00005.jpg"
6	1	2356	14	"00006.jpg"
7	1	2926	18.9737	"00007.jpg"
8	1	2573	15.2315	"00008.jpg"
9	2	1861	14.1421	"00009.jpg"

```
histogram(crackData.Area,30)
```

```
cutoffArea = 4000;
crackData.Risk = discretize(crackData.Area,[0,cutoffArea,inf], ...
    "categorical",["Mild","Severe"]);
montage(crackData.fileName(crackData.Risk == "Severe"))
dsSevere = imageDatastore(crackData.fileName(crackData.Risk == "Severe"))
```

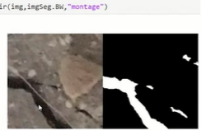


```
outlierImages = crackData.fileName(outlierIndices);
montage(outlierImages,"BorderSize",1)
```




```
imgName = outlierImages(1);
img = imread(imgName);
imgSeg = crackAnalysis(img);

imshowpair(img,imgSeg.Bw,"montage")
```

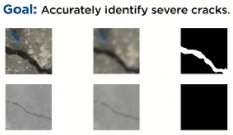
```
imgName = 00183.jpg ;
img = imread(imgName);
imgSeg = crackAnalysis(img);

imshowpair(img,imgSeg.Bw,"montage")
```



Live Tasks
Helps with common tasks like smoothing data or combining variables

Goal: Accurately measure the crack area.

Goal: Accurately identify severe cracks.



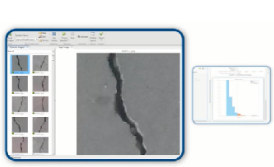
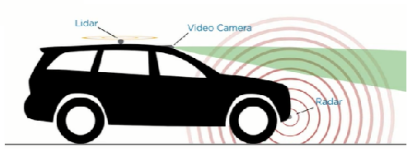


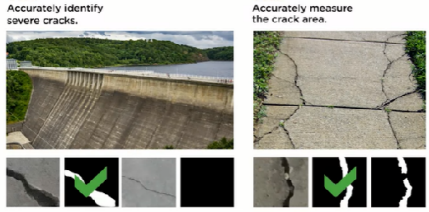
Image Batch Processor App

Final Project Steps:

1. Preprocess the video
2. Isolate the cars
3. Calculate region properties



Goal:
Accurately identify severe cracks.



Goal:
Accurately measure the crack area.

Congratulations! You passed!

Grade received 100%
Latest Submission Grade 100%
To pass 80% or higher

Go to next item

1. You've deployed your algorithm for classifying concrete cracks as either severe or mild risk. You are now in charge of directing workers to repair any severe cracks found. **1 / 1 point**

   Previously in this module, you found that outliers in the number of regions detected indicated that the automated segmentation function performed poorly. For example, here is the segmentation for image 00148.jpg.

   Knowing that it is expensive to send workers to repair cracks, but you also don't want to miss any truly severe ones, how would you deal with these outlier images you've found using the **NumRegions** variable?

   - ( ) Remove the outlier images from the dataset
   - ( ) Do nothing and treat the outlier images the same as the rest of the dataset
   - (•) Evaluate the outlier images to manually label them as severe or mild
   - ( ) Prioritize the cracks in these outlier images for repair

   Correct
   It should require a minimum amount of work to manually inspect only a few images, and is much more efficient than sending a worker out to repair a mild crack that might not need repair.

2. The following questions use the `crackData` table that is found in the CrackData.mat file. To load the `crackData` table variable, run this command: `load CrackData.mat`. **1 / 1 point**

   This table is generated using the `crackAnalysis` function and contains four variables for each of the images in the concrete dataset:

   1. **NumRegions**: The number of regions found.
   2. **Area**: The total number of all true pixels.
   3. **MaxWidth**: The width of a crack at its widest point, measured in number of pixels.
   4. **fileName**: The name of the image file.

   Then create a new variable, "Risk" using the code below, which classifies images with a total crack area below 4000 pixels as "mild," and above 4000 pixels as "severe."

```
load CrackData.mat
cutoffArea = 4000;
crackData.Risk = discretize(crackData.Area,[0,cutoffArea,inf], ...
    "categorical",["Mild","Severe"]);
```

   Question:

   Assume a single pixel in every image has the dimensions of 0.5 mm by 0.5 mm. What is the median **MaxWidth** of just the cracks labeled "Severe"? Provide your answer in millimeters.

   17.4642

   Correct

3. Identify all outlier entries in the **MaxWidth** variable using the median method with a threshold factor of 3. You are encouraged to use the Clean Outlier Data Live Task. **1 / 1 point**

   Think about how these outliers may be different than the previous outliers from question 1.

   Which of the following statements is true?

   - ( ) The outliers images all have a **lower** MaxWidth values than the non-outlier images.
   - (•) The outliers images all have a **higher** MaxWidth values than the non-outlier images.
   - ( ) The outliers images all have **both lower and higher** MaxWidth values than the non-outlier images.

   Correct

4. Inspect a few of the images marked as **MaxWidth** outliers. Again, knowing that it is expensive to send workers to repair cracks, how would you deal with these new outlier images you've found using the **MaxWidth** variable? **1 / 1 point**

   - ( ) Remove the outlier images from the dataset
   - ( ) Do nothing and treat the outlier images the same as the rest of the dataset
   - ( ) Evaluate the outlier images to manually label them as severe or mild
   - ( ) Fire your team and rewrite a new segmentation and analysis function that will be 100% perfect.
   - (•) Prioritize the cracks in these outlier images for repair

   Correct
   In this case, these outliers are likely the result of the cracks actually being very wide. Quick, fix them now!

5. For the next two questions, consider the earlier video of liquid filling a container (frame 92 from the video is shown below). **1 / 1 point**

   Which image processing methods could be reasonable to segment the liquid (the dark liquid with or without the light foam on top) from the background in each frame of the video?

   Choose ALL that apply.

   - [x] Perform background subtraction.

     Correct
     The background is stationary, and the liquid changes every frame. So this is a good option.

   - [x] Convert image to grayscale and perform an intensity-based threshold.

     Correct
     The dark liquid contrasts with the foam and background and should easily be segmented using an intensity threshold.

   - [x] Perform color threshold.

     Correct
     All three regions have unique colors and intensities, and can be easily segmented in one of the available color spaces.

6. Now consider that your application requires you to measure the area of just the light foam on top. **1 / 1 point**

   Which of the three methods would you choose to perform the initial segmentation?

   - ( ) Background Subtraction
   - ( ) Grayscale Thresholding
   - (•) Color Thresholding

   Correct
   In this case, color thresholding would probably be the best choice because the light foam has a similar intensity to the background, but it is a very different color. In our own attempts, we were able to get a result like this:

Congratulations! You passed!

Grade received 100% **To pass** 80% or higher

Go to next item

1. Assume you just developed a segmentation function using a single image, and you have a large set of similar images. How can you use the Image Batch Processor App? **1 / 1 point**

- [x] To export the segmented images

Correct
You can export the segmented images either to the workspace or to files.

- [ ] To refine the segmentation function directly inside of the Image Batch Processor App.
- [x] To check if the segmentation function works well on all the images

Correct
Use the Image Batch Processor app to see the results of the segmentation function alongside the original images.

2. Select all of the correct statements regarding datastores. **1 / 1 point**

- [x] The "readimage" function can extract metadata when reading an image from an image datastore.

Correct
Correct! If you ask for a second output from the "readimage" function, it will return the image's metadata. This includes the image label if you created the datastore using the "LabelSource" option.

- [x] You can use the folder name of an image file to label the image.

Correct
Correct! Set the "LabelSource" option to "foldernames" to extract the subfolder name as labels.

- [ ] A datastore loads an entire set of images when initialized.

3. After initializing a datastore "imds", which of the following lines of code will cycle through all the images? **1 / 1 point**

- (•)
```
while hasdata(imds)
    im=read(imds)
end
```
- ( )
```
while isempty(imds)
    im=read(imds)
end
```
- ( )
```
while ~isempty(imds)
    im=read(imds)
end
```
- ( )
```
while hasinfo(imds)
    im=read(imds)
end
```

Correct

Congratulations! You passed!

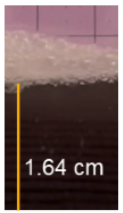
Grade received 100% | Latest Submission Grade 100% | To pass 80% or higher

Go to next item

1. Quiz setup

1 / 1 point

Your goal for this quiz is to process the "liquidVideo.mp4" and calculate the height of the liquid in each frame, like this:



In a previous video, the liquid was isolated using color thresholding. However, for this quiz you will use the background subtraction technique covered previously. The advantage of this method over color thresholding is that it should provide more consistent results if used on new videos with liquids of different colors.

The following questions will guide you through the process. We suggest checking your answer to a question before proceeding to the next one. You can take this quiz as many times as needed.

Question

Your first task is to isolate the liquid using the background subtraction method. Therefore, you'll need to define a background frame to use. Which option provides the best background frame for this video?

- ○ All of the frames averaged together.
- ◉ The first frame of the video.
- ○ The middle frame of the video (number 120).

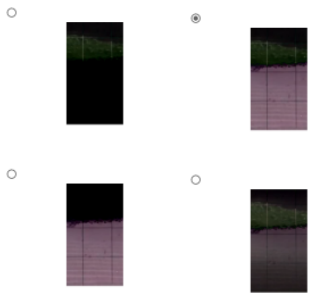
Correct
If your video has a frame with a stationary background and no foreground object present, then this is a great choice for a background frame.

2. Now that you have chosen what to use as the background frame, your next task is to test out the background subtraction method on some example frames.

1 / 1 point

Question

What is the correct result after performing background subtraction on frame 175?



3. Now your task is to isolate the liquid from the images corresponding to the difference between the background and your chosen example frames. We encourage you to experiment around with a few options. After completing this task, answer the following question.

1 / 1 point

Question

What approach is the best method to segment the liquid?

- ○ Grayscale binarization with Otsu's method
- ○ Grayscale binarization with adaptive thresholding
- ◉ Grayscale binarization with a manual threshold value
- ○ Clustering

Correct
This is the ideal method because it will properly isolate the liquid even when it is not present or occupies the entire frame.

4. Now that you have segmented out the liquid from the background subtraction image, your next task is to use morphological operations to improve the mask by removing unwanted true pixels (such as from the grid markings on the background) and unwanted false pixels (such as from the foam). Answer the following question after finding a suitable method.

1 / 1 point

Question

After performing background subtraction, grayscale thresholding, and morphological operations on frame number 175, what is the percentage of true pixels (corresponding to the liquid) in your final mask?

60.8912

Correct
Your final result may vary depending on your methods. The result we got was 60.42.

5. Now apply the entire workflow you've developed in this quiz to the entire video, such that for each frame you:

1 / 1 point

1. Perform background subtraction
2. Segment out the liquid from the resulting image
3. Improve the mask using morphological operations
4. Use the mask to calculate how full the container is

Once completed, answer the following question.

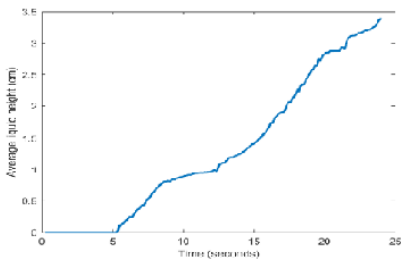
Question

Assume that each frame of the video is 3.4 cm in height. What is the average height of the liquid 16.7 seconds into the video?

1.8886

Correct
For our own algorithm, we got a height of 1.86 cm.

Here is our result for the entire video.



Congratulations! You passed!

Grade received 100% **Latest Submission Grade** 100% **To pass** 80% or higher

Go to next item

1. The function `processPuzzle` was given to you in the previous reading. Use it to analyze the image "Puzzle_01.jpg". **1 / 1 point**

What is the percentage of true pixels in the BW mask produced by this function?

4.5848

Correct

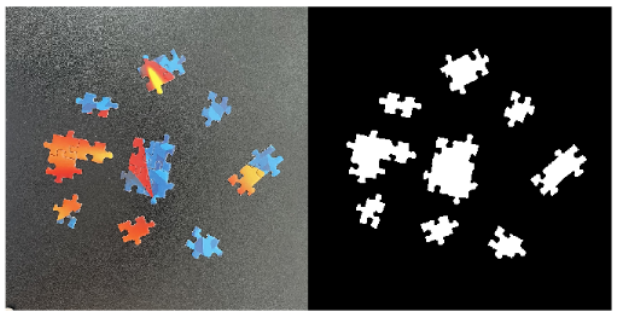
2. Use the Image Batch Processor App to apply the `processPuzzle` function to all 50 images in the puzzle piece collection. **1 / 1 point**

Below are listed four images that we have chosen to be a representative sample of the entire collection. Visually inspect the images and their resulting BW masks in the app. Which ones are fairly accurately segmented using the `processPuzzle` function?

Check all that apply.

- [x] Puzzle_13.jpg

Correct
Although not "perfect", the segmentation of this image is close enough to be "accurate".



- [ ] Puzzle_27.jpg
- [ ] Puzzle_42.jpg
- [ ] Puzzle_49.jpg

3. The previous question should have revealed some images in which the `processPuzzle` function didn't accurately segment the puzzle pieces. Your next task is to modify the function to improve your results. Specifically, you will adjust the Sensitivity parameter found on line 19, which starts with an initial value of 0.60. **1 / 1 point**

Use the Image Batch Processor App to test the function with different sensitivity values until you find one that accurately segments the images from the previous question. (Hint: the initial value is relatively close.)

What is the Sensitivity value that accurately segments the puzzle pieces?

0.57

Correct

4. Using your new value for Sensitivity, apply the function to every image in the collection. The second output of the `processPuzzle` function is a list of areas for each true region in the image. Use this output to create a table with the following properties for each image: **1 / 1 point**

- The number of regions
- The total area of true pixels

You can use either the image datastore or Image Batch Processor App workflows. Once completed, analyze your results to answer the following two questions.

What is the total number of true regions in all 50 images?

628

Correct

5. What is the average size of a true region in all 50 images (in terms of the number of pixels)? **1 / 1 point**

15754

Correct

Congratulations! You passed!

Grade received 100%

Latest Submission Grade 100%

To pass 100% or higher

Go to next item

1. If your segmentation function is not producing adequate results, you can attempt this quiz as many times as needed. 1 / 1 point

How many frames of the video does your algorithm detect at least one car?

155

Correct
This is an acceptable result. Depending on your image processing methods, your answer may vary. For example, our sample solution yields the answer of 149.

2. What is the total region area (in number of pixels) for frame 152? 1 / 1 point

65054

Correct
This is an acceptable result. Depending on your image processing methods, your answer may vary. For example, our sample solution yields the answer of 67211.

3. Looking at the whole video, what is the mode value for the number of cars detected in a frame? 1 / 1 point

- ○ 0
- ◉ 1
- ○ 2
- ○ 3

Correct

4. During the entire video, what is the average size of a region in terms of the number of pixels? 1 / 1 point

16451

Correct
This is an acceptable result. Depending on your image processing methods, your answer may vary. For example, our sample solution yields the answer of 17931.

5. Take some time to compare your results for the number of cars detected in each frame to what you would say is the result when watching the video yourself. No algorithm is going to yield absolutely perfect results. 1 / 1 point

Which frame would you say your segmentation process gave an incorrect result and why? How could you change or add to your algorithm to fix this issue?

It seems to work okay now, except where the overlapping car regions are merged.

Correct
Good job taking the time to think critically about your results. Rarely does image processing always produce perfect results, and it is up to you to find out when and how your algorithms fail.

Congratulations! You passed!

Grade received 100% **To pass** 100% or higher

Go to next item

1. **The purpose of this quiz is to familiarize yourself with the video used in this project. Answer the following questions by viewing and analyzing the video using MATLAB.** 1 / 1 point

What is the total number of frames in the video?

240

Correct

2. What is the frame rate of the video? 1 / 1 point

10

Correct

3. What is the total number of pixels in a single video frame? 1 / 1 point

599844

Correct

Congratulations! You passed!

Grade received 100%
Latest Submission Grade 100%
To pass 100% or higher

Go to next item

1. 1 / 1 point

- **Here you will use your segmentation function that from section 2 to isolate cars in various video frames. This quiz acts as a checkpoint to make sure that you can adequately isolate cars from the video.**
- **You can answer the following questions by either looking at the BW images your segmentation function produces or analyzing the region properties for the frames in question.**
- **If your segmentation function is not producing adequate results, you can attempt this quiz as many times as needed.**

Question

Frame 153 has both white and black cars present in the image. How many regions does your segmentation function produce for this frame?

3

Correct

2. How many regions does your segmentation function produce for frame 201? 1 / 1 point

0

Correct

3. How many regions does your segmentation function produce for frame 148? 1 / 1 point

2

Correct

4. What is the total number of true region pixels detected in frame 22? 1 / 1 point

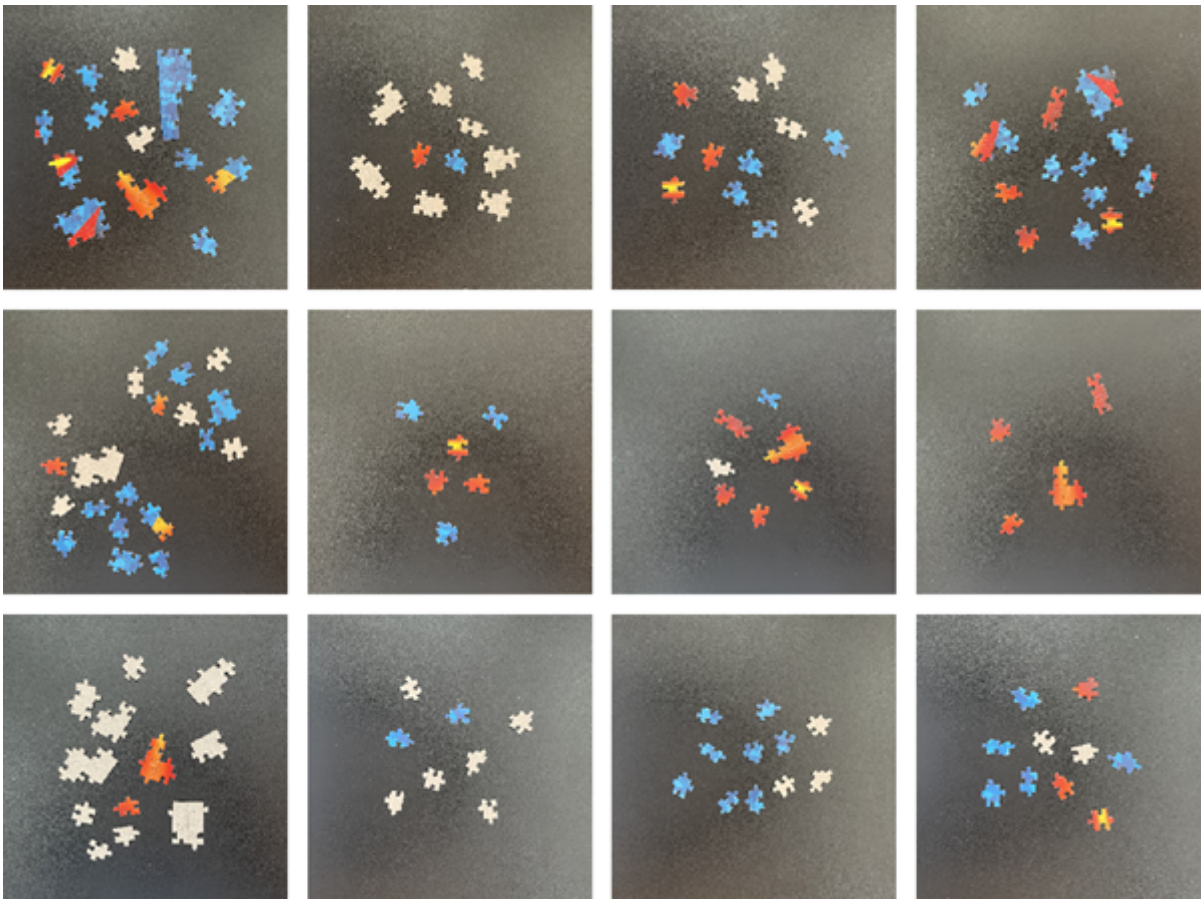
24673

Correct
This is an acceptable result. Depending on your image processing methods, your answer may vary. For example, our sample solution yields the answer of 26306.

Automated Region Counting Project

Project summary

In the folder named "MathWorks Puzzle" in the course files, there is a collection of 50 images of puzzle pieces. A few of the images are shown below.



Your goal for this project is to use automated image processing to analyze this image collection, such as counting the total number of regions in all 50 images.

Puzzle piece region analysis function

We've provided you with a function that segments and analyzes the puzzle pieces.

Copy this function code into your own function file in MATLAB, and use it to complete this project.

```
4   %  Morphological operations and region filtering are then applied to improve the ir
5   %  final segmentation mask is returned in the variable BW, and a list of region are
```

Grades

You have completed all of the assignments that are currently due.

You passed this course! Your grade is 100%.

Item	Status	Due	Weight
Automated Region Counting Project Assessment Quiz	Passed	Aug 14 11:59 PM IST	15%
Analyzing Videos Quiz	Passed	Aug 21 11:59 PM IST	15%
Module 3 Quiz Quiz	Passed	Aug 28 11:59 PM IST	15%
Final Project Assessment 1: Removing noise from video frames Graded External Tool	Passed	Sep 4 11:59 PM IST	15%
Final Project Assessment 2: Car segmentation results Quiz	Passed	Sep 4 11:59 PM IST	20%
Final Project Assessment 3: Analyzing the car region properties Quiz	Passed	Sep 4 11:59 PM IST	20%

Drawing bounding box onto an image without imshow and hold on ...

Unfollow | 10 views (last 30 days) | Show older comments

Alfian Abdul Halin on 9 Dec 2015

Edited: Sandipan Dey less than a minute ago

Accepted Answer: Walter Roberson LEVEL 10 MVP

Vote | 0 | Link | Flag

Hi there. I was wondering if its possible to draw a rectangle, whose coordinates is obtained from regoingprops. I however, am not interested to show the image first. What I intend to do is to get the bounding box coordinates form regionprops, and then just superimpose it onto an image without displaying it. I would then like to save the image (saving is ok... just drawing the rectangle got me a bit tangled up :P). Would appreciate any advice!

The code I'm currently using is as shown above. (It's in a for loop) where 'l' is the counter.

THANKS in advance!!!

Theme | Copy

```
f = figure,imshow(frame, 'Border', 'tight');
hold on;
rectangle('Position',s(l).BoundingBox,'EdgeColor','y','LineWidth',2)
print(f, '-r80', '-djpeg', [dirName 'results' '\' jpegFiles(i).name]);
close all;
```

0 Comments

SD Comment on this question...

Answer this question

Accepted Answer

Walter Roberson LEVEL 10 MVP

on 9 Dec 2015

Vote | 0 | Link | Flag

If you have the Computer Vision Toolbox then you can use shapeInserter

If you do not have that, then you can just assign values to the appropriate rows and columns in the array. Watch out for the possibility that the values for the coordinates are not integral

Theme | Copy

```
bb = s(1).BoundingBox;
from_row = ceil(bb(2));
to_row = from_row + bb(4) - 1;
from_col = ceil(bb(1));
to_col = from_col + bb(3) - 1;
yellow = [255, 255, 0];

%left and right side
frame(from_row:to_row, [from_col, to_col], 1) = yellow(1);
frame(from_row:to_row, [from_col, to_col], 2) = yellow(2);
frame(from_row:to_row, [from_col, to_col], 3) = yellow(3);
%top and bottom
frame([from_row, to_row], from_col+1:to_col-1, 1) = yellow(1);
frame([from_row, to_row], from_col+1:to_col-1, 1) = yellow(2);
frame([from_row, to_row], from_col+1:to_col-1, 1) = yellow(3);
```

3 Comments Show 2 older comments

SD Sandipan Dey less than a minute ago

Edited: Sandipan Dey less than a minute ago

or slightly moifiying the abovr code to change color channels at once and draw a bounding box (e.g., one obtained using `regionprops`) with a given (odd) width (e.g., 3):

Theme Copy

```
function frame = drawBB(bb, frame, width)
    from_row = ceil(bb(2));
    to_row = from_row + bb(4) - 1;
    from_col = ceil(bb(1));
    to_col = from_col + bb(3) - 1;
    yellow = [255, 255, 0];
    for i = -fix(width/2):fix(width/2)
        %left and right side
        frame(from_row:to_row, [from_col+i, to_col+i], :) = ...
            permute(repmat(yellow, to_row-from_row+1, 1, 2), [1,3,2]);
        %top and bottom
        frame([from_row+i, to_row+i], from_col+1:to_col-1, :) = ...
            permute(repmat(yellow, 2, 1, to_col-from_col-1), [1,3,2]);
    end
end
```

The output can be seen in the following montage (the input image is taken from a coursera course by *mathworks*).

SD Comment on this answer...

More Answers (0)

Answer this question

ANNOUNCEMENT
New feature in MATLAB Answers: Moving comments to and from answers
In MATLAB Answers, oftentimes we see good comments that provide solutions in...

See Also

MATLAB Answers

What does the 1 in the row and column in Blight_red = img(row(1), column(1), 1) means?
2 Answers

Local histogram equalization manually
2 Answers

Use mask to crop circle images from a picture with Matlab
1 Answer

Entire Website

Cannot load app with grid that has row height or column width of zero
Bug Reports

xlsrange
File Exchange

MATLAB® Report Generator™ may cause MATLAB to hang when the table component's column widths are not fully specified.
Bug Reports

Categories

Image Processing and Computer Vision › Image Processing Toolbox › Image Segmentation and Analysis › Region and Image Properties

Tags

rectangle regionprops

Poll

How often is your MATLAB code officially or unofficially reviewed?

- ◯ Never, I don't typically share code
- ◯ Never, even when my code is shared
- ◯ Occasionally
- ◯ Sometimes
- ◯ More often than not
- ◯ Always or almost always

557 votes 5 comments 243 views

Vote

Join the conversation

Example final project solution

Table of Contents

This is a Live script of our example solution for the final project. As there are many ways to perform the tasks for this project, your own solution will likely be different. Feel free to compare your own methods and results with our own.

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

	NumberRegions	MeanRegionSize	TotalRegionSize
80	1	19832	19832
81	1	18888	18888
82	1	17945	17945
83	1	17013	17013
84	1	13610	13610
85	1	5781	5781
86	0	NaN	0
87	0	NaN	0
88	0	NaN	0
89	0	NaN	0
90	0	NaN	0
91	0	NaN	0
92	0	NaN	0
93	0	NaN	0
94	1	13005	13005
95	1	25068	25068
96	1	23649	23649
97	1	22061	22061
98	1	20898	20898
99	1	20167	20167
100	1	19213	19213

⋮

Segmentation function

```
function [BW,maskedImage] = segmentCars(X)
%segmentCars Segment image using auto-generated code from imageSegmenter app
%  [BW,MASKEDIMAGE] = segmentCars(X) segments image X using auto-generated
%  code from the imageSegmenter app. The final segmentation is returned in
%  BW, and a masked image is returned in MASKEDIMAGE.

% Auto-generated by imageSegmenter app on 17-Jun-2021
%----------------------------------------------------
```

```
% 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
```

Final Challenge

Final Challenge

As a final challenge, use the region properties to overlay a bounding box on each frame of the video (as shown in the GIF below).

Share your work

You've worked hard on this final project, so show off your image processing skills with your social network. Post an image, video, or GIF from your project with the tag **#MATLAB** and include a link to the specialization https://coursera.org/specializations/image-processing

Good luck!

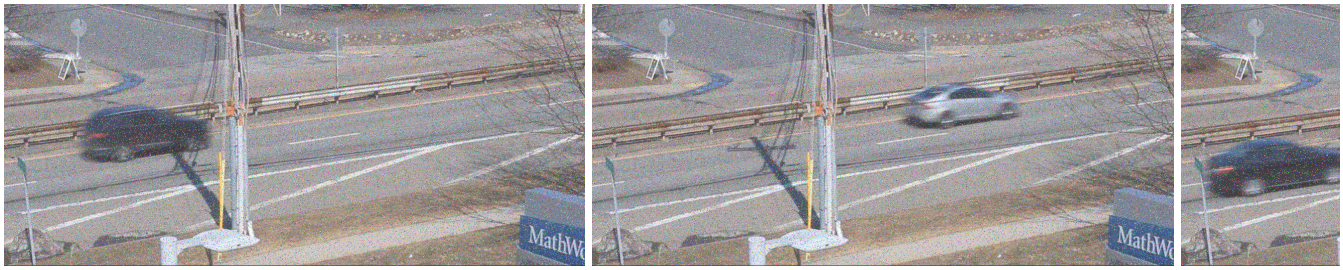
Removing noise from a video frame

My Solutions >

As described in the previous reading, in this assignment, you will write a function called `enhanceImage()` tha

- takes in a single color image
- removes the noise
- converts the image to **grayscale**
- outputs the final result

There are 3 available frames from the video you can use to test your function, named: *Frame96.bmp*, *Frame*



After passing this assignment, you should use your `enhanceImage()` function on every frame of the final prc rest of the project.

Function

Save Reset MATLAB Documentation (https://www.mathworks.com/help/)

```
function img = enhanceImage(img)

% Insert code here to remove noise and convert to grayscale
img = im2gray(medfilt3(img));

end

```

Code to call your function

Reset

```
% See the results of you function on the three different frames.
img96Noisy = imread("Frame96.bmp");
img96 = enhanceImage(img96Noisy);
img113Noisy = imread("Frame113.bmp");
img113 = enhanceImage(img113Noisy);
img227Noisy = imread("Frame227.bmp");
img227 = enhanceImage(img227Noisy);

montage({img96Noisy,img96,img113Noisy,img113,img227Noisy,img227})
% Note: You can click on the images to zoom in
```

Run Function

Assessment: All Tests Passed

Submit

- Does the function convert image to grayscale?
- Does the function adequately remove the noise?

Section 1: Preprocessing the video

Now that you have watched the project video "RoadTraffic.mp4", the remainder of this project is split into three sections, with a graded assessment after each one.

The main task

The goal of section 1 is to preprocess the video by removing the noise from the frames. In section 2, you will isolate cars using segmentation. However, because the cars are not all the same color, you can simplify the problem by removing the color information and converting the video to grayscale.

In summary, your goal is to go from this frame in the original video:

To this:

Section 1: Preprocessing the video

Now that you have watched the project video "RoadTraffic.mp4", the remainder of this project is split into three sections, with a graded assessment after each one.

The main task

The goal of section 1 is to preprocess the video by removing the noise from the frames. In section 2, you will isolate cars using segmentation. However, because the cars are not all the same color, you can simplify the problem by removing the color information and converting the video to grayscale.

In summary, your goal is to go from this frame in the original video:

To this:

Section 2: Isolating the cars

The main task

Now that you have filtered the video and converted it to grayscale, the next task is to use image segmentation to isolate the cars driving on the road. The end result of this section is to have a process for taking frames from the video and generates segmented masks.

For instance, frame number 10:

Would output something like this:

There are several ways to go about isolating the moving cars from the stationary background, so you'll likely get a result that is not identical to what is shown above. Your goal is to reliably isolate cars from the video so that the number of regions detected equals the number of cars in the original frame.

☰ **Item Navigation**

Section 3: Calculating region properties

Now that you have a process for segmenting cars, you'll need to apply it to the whole video so that you can statistically analyze the regions found in each frame.

In this section, your goal is to summarize the region information for each frame. Specifically, you will 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
- total region size

Your table should look something like the following (your exact values will differ from what is shown below and depend on your processing steps from the previous two sections of the project):

`carData = 240×3 table`

	NumberRegions	MeanRegionSize	TotalRegionSize
20	2	18077	36154
21	1	27222	27222
22	1	26306	26306
23	1	25066	25066
24	1	23808	23808
25	1	22216	22216
26	1	16140	16140
27	1	7173	7173
28	0	NaN	0

Once you have created this table, you will use it to take the final assessment for the course.

After you pass this assessment, we will share our example solution for all three steps of this project.

Segmenting and Analyzing Concrete Crack Images

Segment the Crack Images

There is no right answer for how to best segment and analyze the images in the concrete crack dataset. Your approach will also depend on your goals. For example, accurately segmenting both large and small cracks is challenging. Choose one of the goals listed below and try segmenting the cracks with that goal in mind. Recall the approaches to segmenting cracks in Course 2 of this specialization that you can use as a starting point.

Use the Image Batch Processor App to check your results and refine your approach.

Choose a goal:

1. Segment small and large cracks accurately to determine their area
2. Identify images with large cracks that should be prioritized for repair.

Our Segmentation and Analysis function

Stop! Don't read until you try segmenting the cracks.

Below you can see a detailed description of a function we used for goal 1. Later in this course you'll learn more about analyzing the crack images.

...

The following example function was our approach to goal 1 above. It is made to be compatible with the Image Batch Processor App.

```
function results = crackAnalysis(img)
%crackAnalysis segments and analyzes an image from the concrete crack dataset.
%
%  results = crackAnalysis(img) preprocesses, segments, and analyzes image img.
%  results, is returned as a structure array with 4 fields:
%  BW, the final black and white mask.
%  NumRegions, the number of individual regions in the mask.
%  Area, the total area in number of pixels of all regions.
%  MaxWidth, the maximum width in number of pixels of all regions.
%
```