PIXELS TO PICTURES

A PROGRAMMING COURSE ON IMAGES WITH MATLAB

Instructor Guide

Module 11: Text on Images

Prerequisite Domain Knowledge: None

Expected Completion Time: 60 minutes

Table of Contents

Recognizing Text in Images

Expected Duration: 20 minutes

Learning Objectives

- Text in images is recognized by analyzing light and dark areas in the image to identify each alphabetic letter or numeric digit.
- Text needs to have a solid background for MATLAB to read it
- A region of interest (ROI) can be selected in an image where MATLAB ignores everything in the image apart from the ROI.

Materials

- MATLAB®

Steps

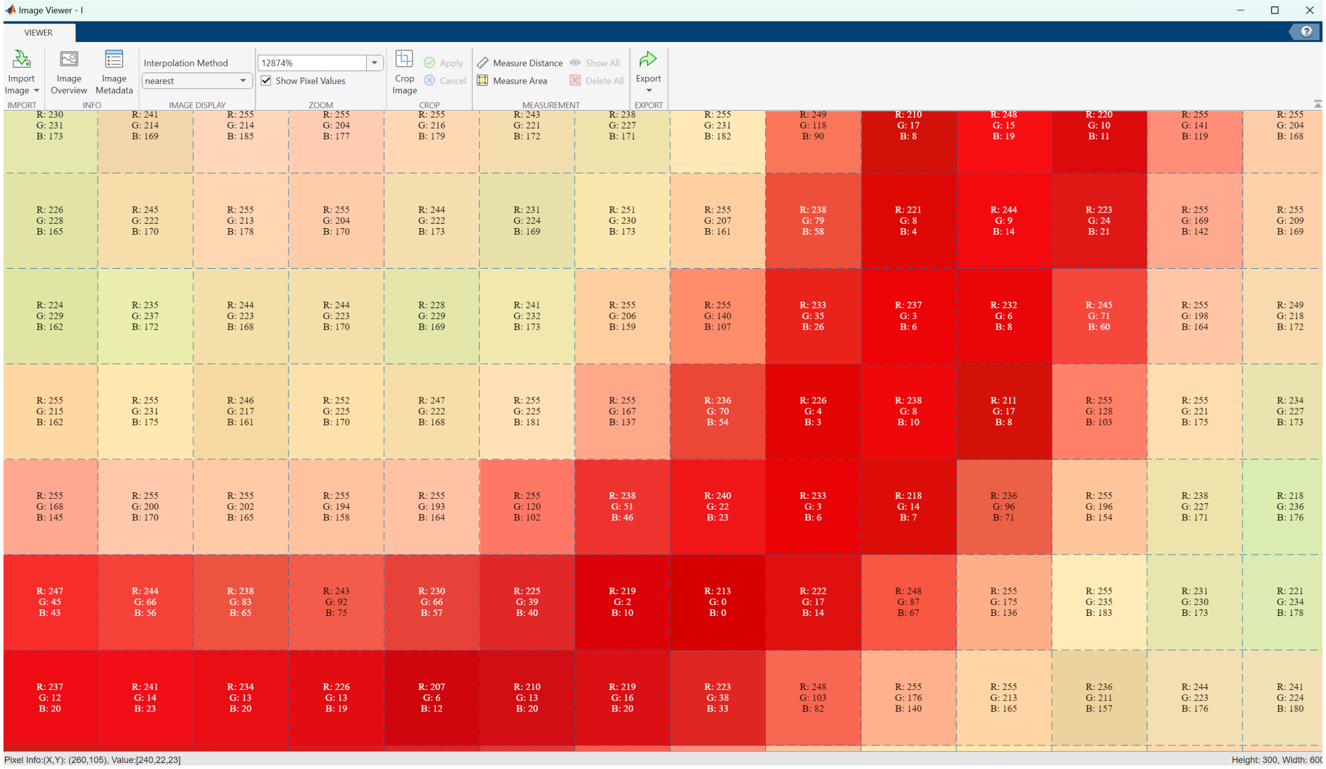
Tell the students that we will be talking about images that have text in them.

In the **Command Window** in MATLAB, type:

```
>> I = imread('hello.jpg');
>> imageViewer(I)
```

This will open the Image Viewer App with a simple image that has just the words ‘Hello there!’ in it.

Students can now explore the pixel values of the image, especially for the text. It should just be like any other image and have RGB values.



Interpolation Method: nearest

12874%

☑ Show Pixel Values

R: 230 G: 231 B: 173	R: 241 G: 214 B: 169	R: 255 G: 214 B: 185	R: 255 G: 204 B: 177	R: 255 G: 216 B: 179	R: 243 G: 221 B: 172	R: 238 G: 227 B: 171	R: 255 G: 231 B: 182	R: 249 G: 118 B: 90	R: 210 G: 17 B: 8	R: 248 G: 15 B: 19	R: 220 G: 10 B: 11	R: 255 G: 141 B: 119	R: 255 G: 204 B: 168
R: 226 G: 228 B: 165	R: 245 G: 222 B: 170	R: 255 G: 213 B: 178	R: 255 G: 204 B: 170	R: 244 G: 222 B: 173	R: 231 G: 224 B: 169	R: 251 G: 230 B: 173	R: 255 G: 207 B: 161	R: 238 G: 79 B: 58	R: 221 G: 8 B: 4	R: 244 G: 9 B: 14	R: 223 G: 24 B: 21	R: 255 G: 169 B: 142	R: 255 G: 209 B: 169
R: 224 G: 229 B: 162	R: 235 G: 237 B: 172	R: 244 G: 223 B: 168	R: 244 G: 223 B: 170	R: 228 G: 229 B: 169	R: 241 G: 232 B: 173	R: 255 G: 206 B: 159	R: 255 G: 140 B: 107	R: 233 G: 35 B: 26	R: 237 G: 3 B: 6	R: 232 G: 6 B: 8	R: 245 G: 71 B: 60	R: 255 G: 198 B: 164	R: 249 G: 218 B: 172
R: 255 G: 215 B: 162	R: 255 G: 231 B: 175	R: 246 G: 217 B: 161	R: 252 G: 225 B: 170	R: 247 G: 222 B: 168	R: 255 G: 225 B: 181	R: 255 G: 167 B: 137	R: 236 G: 70 B: 54	R: 226 G: 4 B: 3	R: 238 G: 8 B: 10	R: 211 G: 17 B: 8	R: 255 G: 128 B: 103	R: 255 G: 221 B: 175	R: 234 G: 227 B: 173
R: 255 G: 168 B: 145	R: 255 G: 200 B: 170	R: 255 G: 202 B: 165	R: 255 G: 194 B: 158	R: 255 G: 193 B: 164	R: 255 G: 120 B: 102	R: 238 G: 51 B: 46	R: 240 G: 22 B: 23	R: 233 G: 3 B: 6	R: 218 G: 14 B: 7	R: 236 G: 96 B: 71	R: 255 G: 196 B: 154	R: 238 G: 227 B: 171	R: 218 G: 236 B: 176
R: 247 G: 45 B: 43	R: 244 G: 66 B: 56	R: 238 G: 83 B: 65	R: 243 G: 92 B: 75	R: 230 G: 66 B: 57	R: 225 G: 39 B: 40	R: 219 G: 2 B: 10	R: 213 G: 0 B: 0	R: 222 G: 17 B: 14	R: 248 G: 87 B: 67	R: 255 G: 175 B: 136	R: 255 G: 235 B: 183	R: 231 G: 230 B: 173	R: 221 G: 234 B: 178
R: 237 G: 12 B: 20	R: 241 G: 14 B: 23	R: 234 G: 13 B: 20	R: 226 G: 13 B: 19	R: 207 G: 6 B: 12	R: 210 G: 13 B: 20	R: 219 G: 16 B: 20	R: 223 G: 38 B: 33	R: 248 G: 103 B: 82	R: 255 G: 176 B: 140	R: 255 G: 213 B: 165	R: 236 G: 211 B: 157	R: 244 G: 223 B: 176	R: 241 G: 224 B: 180

Pixel Info:(X,Y): (260,105), Value:[240,22,23]

Height: 300, Width: 600

As humans, our eyes can see text in an image and immediately read it.

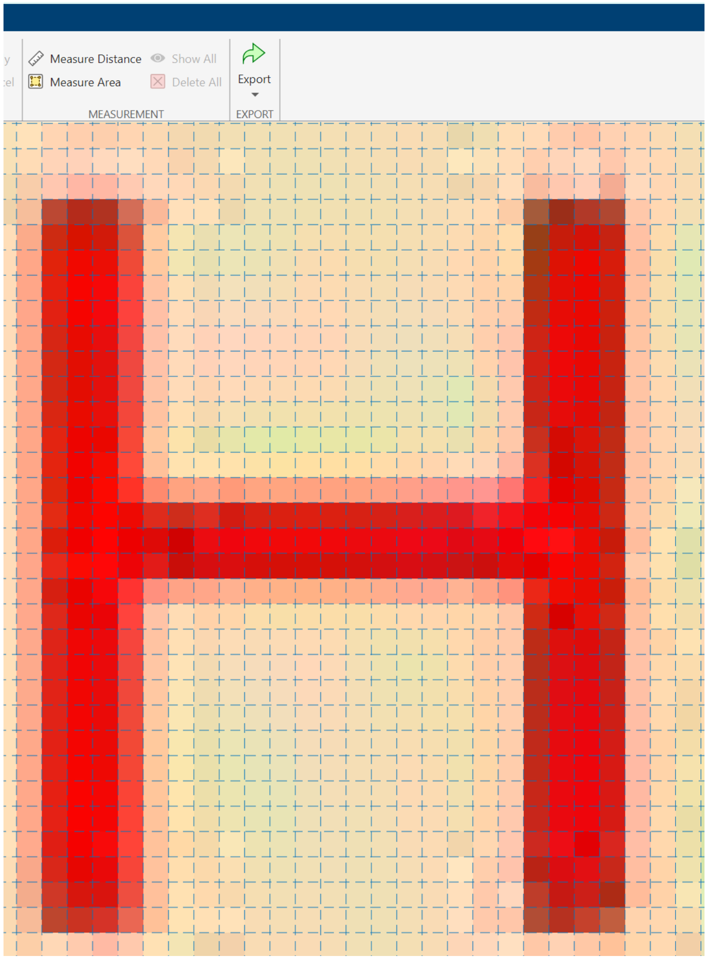
Ask:

- **But can computers read text from images as well?**
- Encourage student to think about applications where computers can recognize text from images. Some examples are:
- There are machines on the toll highways that can see your car’s license plate number and know whom the car belongs to and charge toll accordingly.
- ATM machines accept cash and check deposits by reading the number in value in the cash notes and the text in the check.
- Self-driving cars need to read road signs (STOP, YIELD) and act accordingly.

So if all a computer sees in an image is a bunch of pixels with RGB values, how are they able to distinguish text?

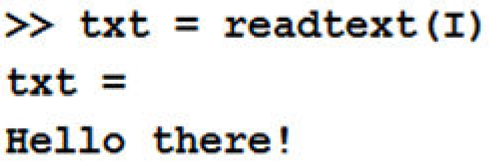
People over the years have been training computers to recognize text by using hundreds of different images of the same alphabet and letting the computer determine a pattern for the images its seeing. For example, in case of the letter ‘H’, after seeing many different kinds of ‘H’, the computer knows that if it sees a pattern similar to two parallel lines (equal size) with a short perpendicular line joining them in the middle, the pattern matches that of the letter ‘H’.

Computers are very good at looking for patterns and very quick at it. They analyze the light and dark areas in the image to see if the pattern matches any of the known alphabets or numbers. For the example of letter ‘H’, the background color pixels contrast distinctly with the foreground pixels in the particular pattern mentioned above and the computer knows it is looking at the alphabet ‘H’.



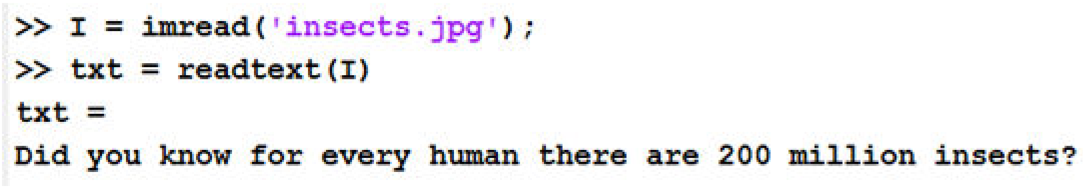
Give them an example of the pattern of the letter ‘A’, which is similar to ‘H’ since it also has two long lines and a short line joining them in the middle. However, the pattern differs because for ‘H’, the long lines are parallel, but for ‘A’, the lines slant towards each other till the tip meets. Similarly, each alphabet has a particular pattern that the computer knows to look for.

Ask them to try a function called `readtext` to see if MATLAB can read the text in the image.



```
>> txt = readtext(I)
txt =
Hello there!
```

Ask the students to try out another example, by reading in the image ‘insects.jpg’ using the `readtext` function on it



```
>> I = imread('insects.jpg');
>> txt = readtext(I)
txt =
Did you know for every human there are 200 million insects?
```

They can view the image using `imshow` to check if it read the message correctly.

Writing Text on Images

Expected Duration: 30 minutes

Learning Objectives

- Write text on images using MATLAB

Materials

- MATLAB®

Steps

Let the students know that we can not only read text from images, but also write text on images - to create memes!

Ask the students to open the following script:

```
open 'TextOnImages.mlx'
```

The first line of code says:

```
I = createNewPage(100,400);
```

This creates a blank white page of 100 rows and 400 columns.

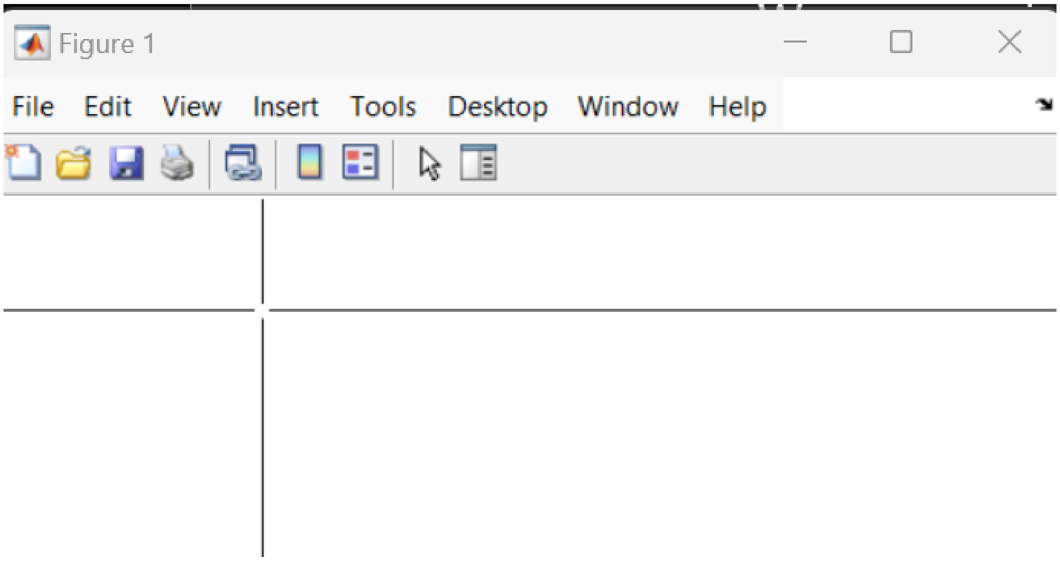
Next they will use the `writetext` function to write on the blank page:

```
J = writetext(I, 'I love MATLAB!');
```

Have the students run the script.

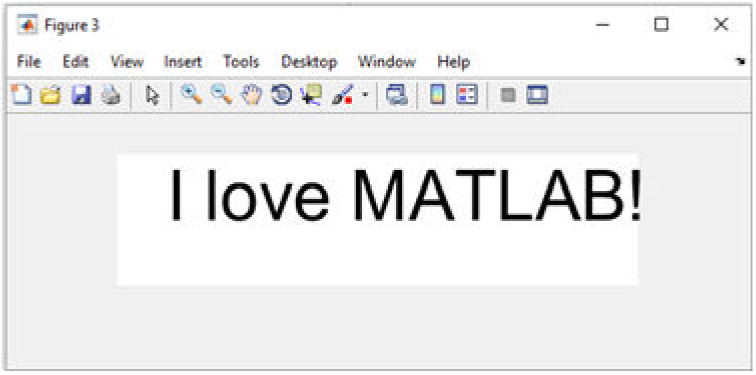
This will show the blank white page and the mouse pointer will change to a large cross. You can use the center of the cross to select where the text will appear. After displaying the text, ask:

- *What might be some ways that you would want to customize this text?*
- Students may mention characteristics like font size, font color, background color, etc.



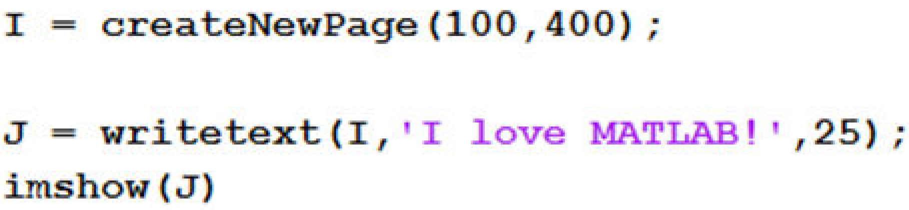
We can use `imshow` to view the image with the writing:

```
imshow(J)
```



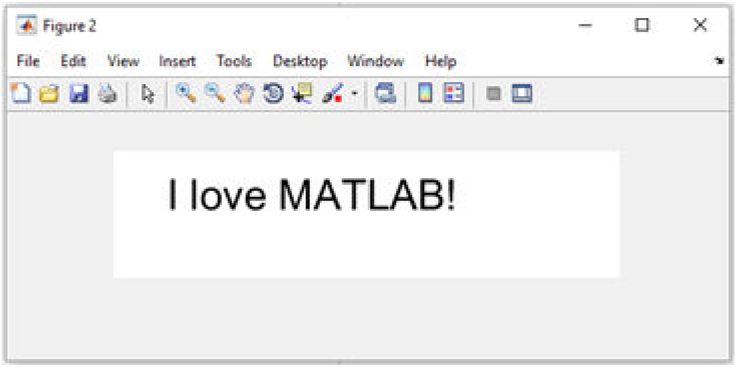
I love MATLAB!

The default font size is 40, and it can be changed as the third input of the `writetext` function:



```
I = createNewPage(100,400);

J = writetext(I,'I love MATLAB!',25);
imshow(J)
```

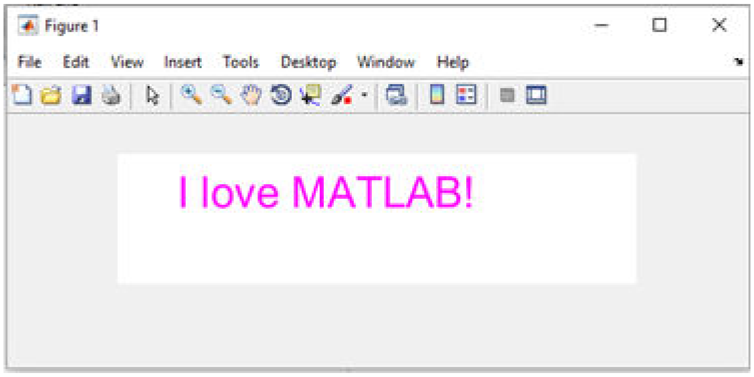


I love MATLAB!

Also the color of the text can be changed using RGB values as the fourth input of the `writetext` function:

```
I = createNewPage(100,400);

J = writetext(I,'I love MATLAB!',25,[255 0 255]);
imshow(J)
```

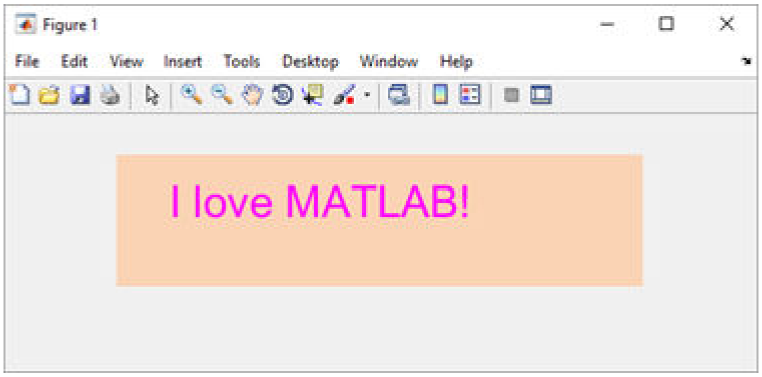


I love MATLAB!

The function `createNewPage` also takes a third input which lets us change the color of the page from white to something else using RGB values again:

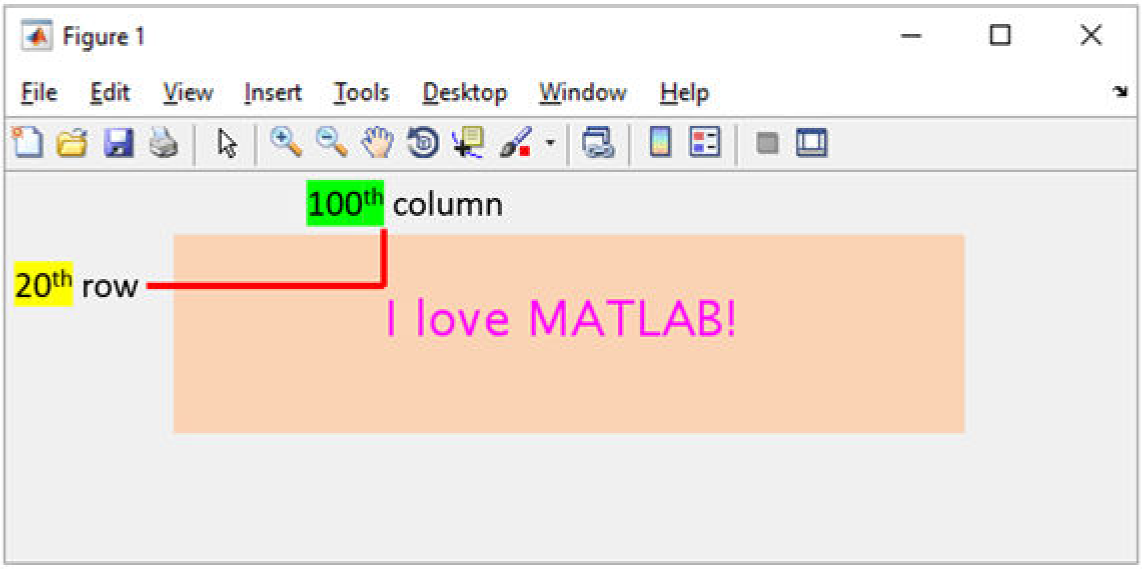
```
I = createNewPage(100,400,[250 210 180]);

J = writetext(I,'I love MATLAB!',25,[255 0 255]);
imshow(J)
```



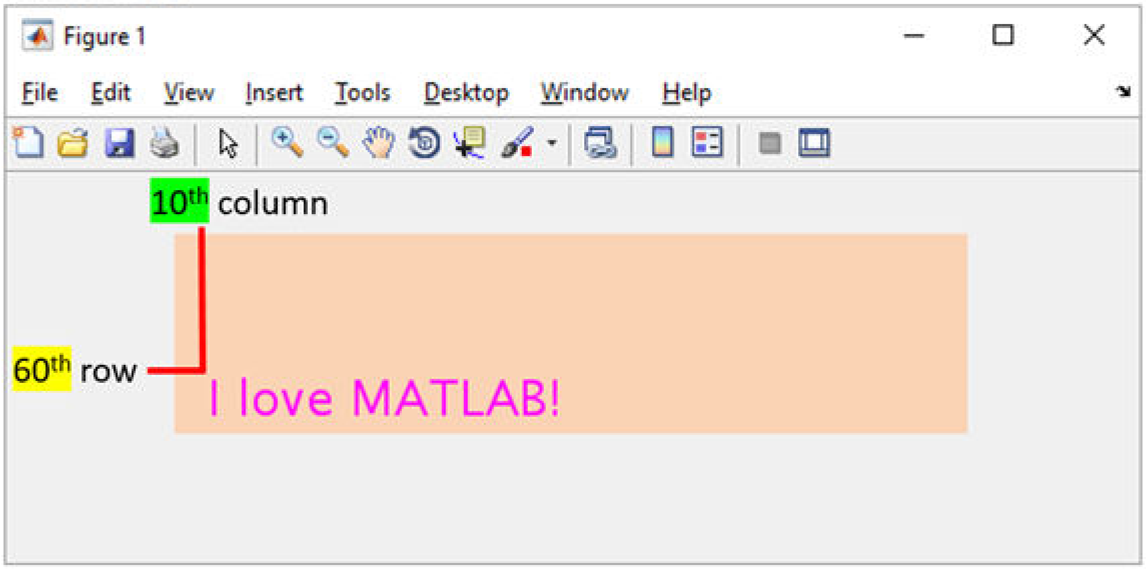
Additionally, if we do not want to keep clicking on an area of the image every time we want to write the text, we can provide a row and column location for the text to appear as the fifth input:

```
J = writetext(I,'I love MATLAB!',25,[255 0 255],[20 100]);
imshow(J)
```



or

```
J = writetext(I,'I love MATLAB!',25,[255 0 255],[60 10]
imshow(J)
```



Ask the students to use the `readtext` function to check if MATLAB can read what they are writing:

```
I = createNewPage(100,400,[250 210 180]);

J = writetext(I,'I love MATLAB!',25,[255 0 255]);
imshow(J)

txt = readtext(J)
```

```
txt =
I love MATLAB!
```

Students should explore and try to find an answer to these questions.

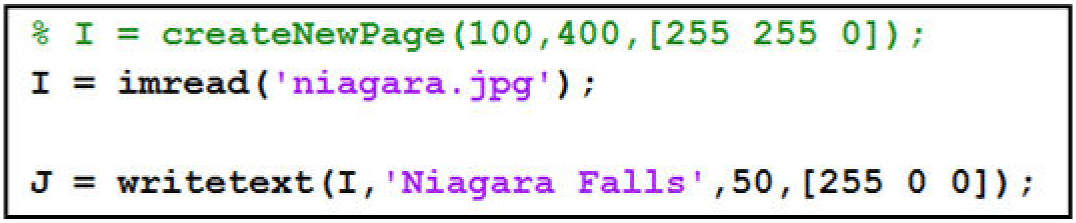
Ask:

1. *What will happen if you continue to decrease the font size?*
2. Students should notice that there becomes a point where MATLAB is not able to make out the patterns correctly anymore.
3. *How does the background color and font color combinations impact how well MATLAB can read the messages?*
4. Students may notice that there are some color combinations that make it more difficult for MATLAB to read the messages.

Tell the students that instead of using a plain background, we can actually use any image we want as background.

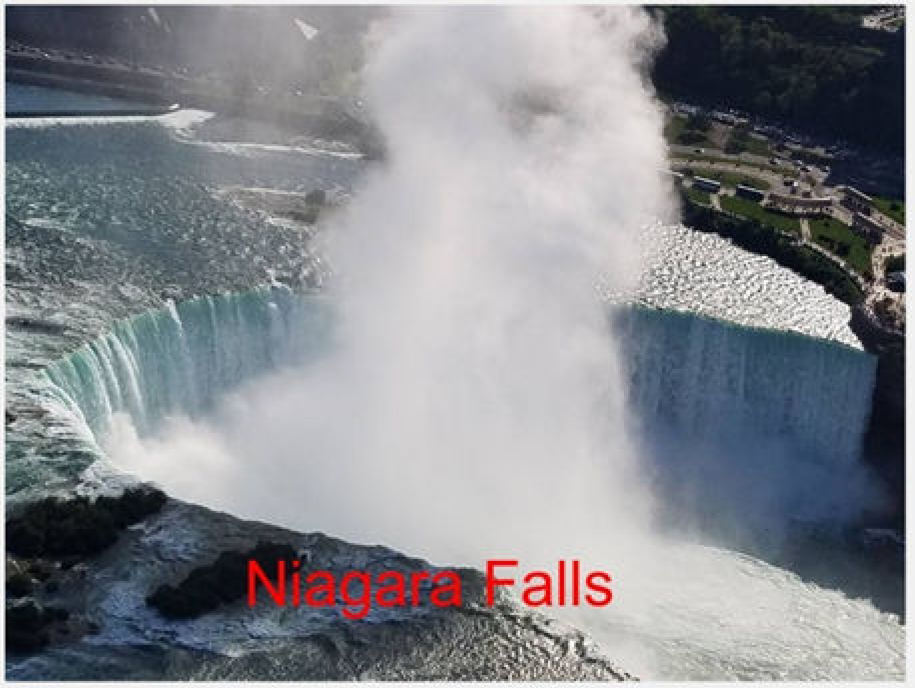
In the script, comment out the first line by putting a % sign in front of it. Remind the students that anytime MATLAB sees anything in green it knows that it can ignore that line because it is not a command but just a comment.

Instead, read in an image of your choice and write text on it:



```
% I = createNewPage(100,400,[255 255 0]);
I = imread('niagara.jpg');

J = writetext(I,'Niagara Falls',50,[255 0 0]);
```



Remind the students that since the background is now no longer a solid color, the `readtext` function can no longer reliably extract the text patterns from the image.

Give the students 10 minutes to play around with the various options and create their own messages.

Inform the students that the maximum font size allowed in an image is 200. Anything more than 200 will result in MATLAB giving an error.

When time has expired, ask:

1. *What surprises did you have while adding text?*
2. Students may mention that they were surprised that MATLAB cannot read font sizes that are too small.

3. *How might your new understanding of color and pattern recognition be helpful?*
4. Students may mention a real-world use or perhaps a use for their final project.