

PIXELS TO PICTURES

A PROGRAMMING COURSE ON IMAGES WITH MATLAB

Instructor Guide

Module 14: Applying Custom Masks on Images

Prerequisite Domain Knowledge: Applying masks, resizing/rotation images

Expected Completion Time: 60 minutes

Table of Contents

Apply Custom Masks on Images.....	1
Learning Objectives.....	1
Materials.....	1
Steps.....	1

Apply Custom Masks on Images

Expected Duration: 60 minutes

Learning Objectives

- Create custom masks from images
- Overlay masks on images
- Algorithm development

Materials

- MATLAB®

Steps

Tell the students that they will now take what they have learned about applying masks a step further. The masks used in the last activity may have been great, but you can make your own masks using MATLAB. Have the students follow along as you guide them to create their own masks.

Ask the students to open and run the following script:

```
open 'createMask.mlx'
```



Next tell them that we will use the `SelectFreehand` function to select anything we want from the image.

```
mask = SelectFreehand(I);
```

When we run this, the image shows up and the mouse pointer turns into a small cross



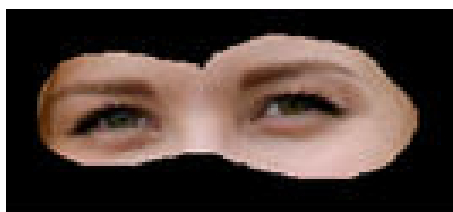
In this state, we can click anywhere in the image and drag the mouse pointer around anything we want to select in the image, for example, the eyes.



This creates a mask variable the same size as the original image. Use `imshow` to view it.



There is a lot of black in the mask image, which is not needed. We can use the `imsnip` function to snip out our mask of interest from the rest of the image.



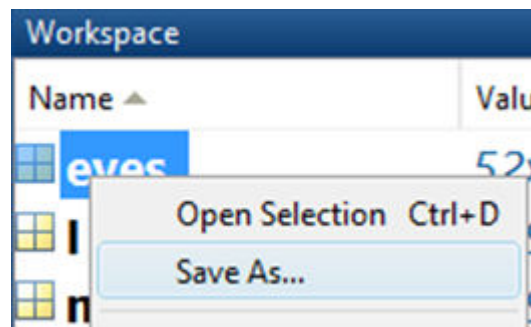
So here we have created our first mask! Here is what the script of create mask should look like:

```
I = imread('happy.jpg');  
figure  
imshow(I)  
  
mask = SelectFreehand(I);  
figure  
imshow(mask)  
  
eyes = imsnip(mask);  
figure  
imshow(eyes)
```

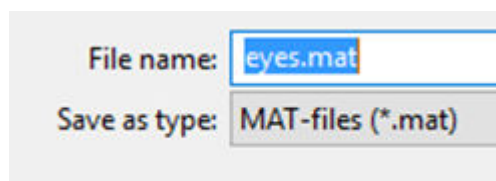
It would be great if we can save the eyes variable from the workspace so that we can load it and use it whenever we want to. Well, we can.

Remind the students about MAT files that load variables in the workspace. We will now learn how to create these MAT files and save *our* variables in them.

In the Workspace, right-click on the eyes variable and click on 'Save as'.



Save the file as eyes.mat. The students should notice that the file is being saved as a MAT file.



Clear the **Workspace** and load the eyes variable in the workspace with the following code at the Command Line:

```
>> clear all
>> load eyes.mat
```

Let the students know that they can use this method to save any Workspace variable they like, for example, all the images where they are applying masks, can be saved from variables into MAT files.

So we have our custom mask in a MAT file ready to be loaded whenever we need it. Lets go use it.

Ask the students to open the following script:

```
open 'ApplyCustomMask.mlx'
```

This should open up a script with steps identical to ApplyMask.mlx.

Run the script. The glasses mask should now be replaced with the eyes mask.

Load a mask images

```
load eyes.mat
```

View mask

```
figure
imshow(eyes)
```

Get location where mask should be overlayed on the face

```
location = [140 180];
```

Superimpose the mask on the face image

```
final1 = superImpose(I,eyes,location);
```



Students should now feel comfortable about adjusting the mask to fit the dog's face by using `imresize` and `imrotate` as needed. They may also need to adjust the location of the mask placement.

View mask

```
figure  
imshow(eyes)
```

Adjust mask

```
eyes = imresize(eyes,0.8);  
eyes = imrotate(eyes,-30);
```

Get location where mask should be overlayed on the face

```
location = [140 200];
```



Students have learned a lot of concepts here. Give the remaining time to practice everything they have learned so far with other images. Mention that the most important thing is to break down whatever they want to do into small steps (one command or function) and build an algorithm. (Be sure to reserve about 10 minutes to close the activity)

Tell the students that they can reuse the `createMask.mlx` script and `ApplyCustomMask.mlx` script to create and apply their own custom masks. They can also save any variable they like in MAT files.

Ask them to be careful to save the MAT files with different names. If they use the same name again and again to save variables in a MAT file, MATLAB will only save the latest variable and forget the older ones. So each MAT file must have a distinct name. Alternatively, they can store several variables in a single MAT file.

When there is about 10 minutes remaining, gather the students.

Ask:

- *What was a fun image that you created?*
- Encourage students to share what special effects they used on an image along with the resulting image.
- *What challenges did you run into while working on your image?*
- Accept all reasonable responses from students. Encourage students to share any solutions to any challenges that are discussed.

Encourage students to walk around the classroom and view what others have done.