Assignment 2

Keanu Francis

Florida Atlantic University

2-11-2022

Professor Oge Marques

Introduction

In this assignment, we had a choice to learn and demonstrate one of the approved MATLAB applications ranging from batch processors to color grading. For my assignment, I decided to use the image labeler. The image labeler can be used to add labels, sub-labels and attributes to a set of images. These labels can then be used to train various kinds of image recognition algorithms or deep learning algorithms. Since deep learning would take too much time to implement, I decided to teach a simple object recognition algorithm I identify parts of the face. This serves as an important first step in understanding how to implement this process and expand it to even more data fields.

1. Learning Image Labeling

The first step before I experiment with image labeling is to learn how the app works. I will do this be following the tutorial in the MATLAB documentation and focus on the labeling of book covers

The first step is to load the book covers from the vision data to an image folder file then save them then create a datastore and use that data store to load the image labeler app.

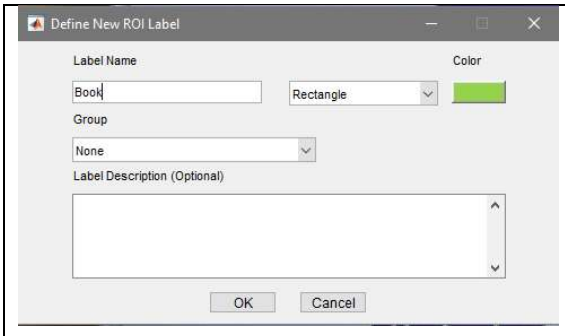
Loading Unlabled Data

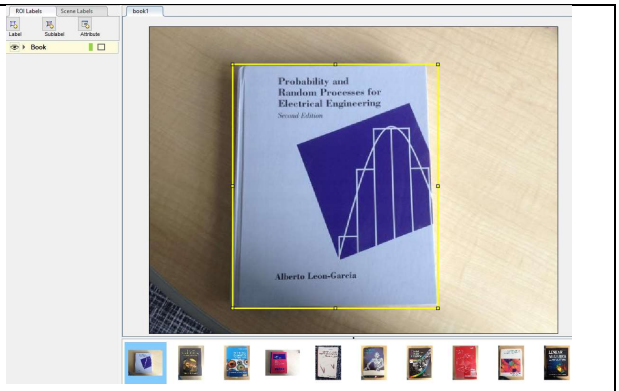
From the Documentation i loaded the books dataset and added labes for the book the title and author. Though this was not done for every book

```
imageFolder = fullfile(toolboxdir('vision'),'visiondata','bookcovers')
imds = imageDatastore(imageFolder);
imageLabeler(imds);
```

a. Adding labels

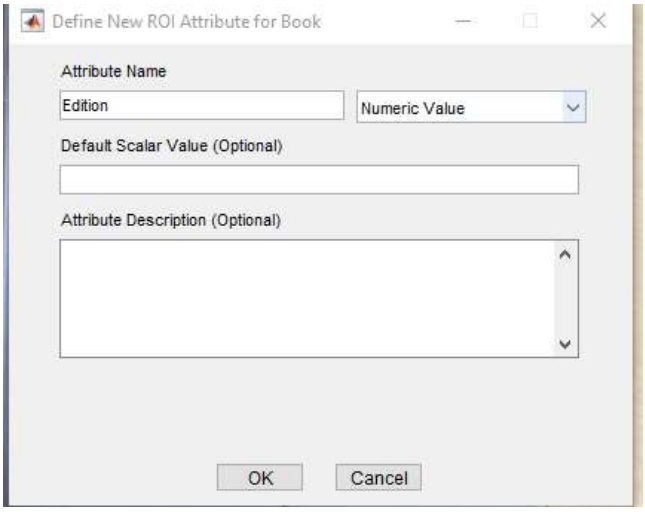
Once the app had been loaded, I added the book label selected a color and started labeling each image.





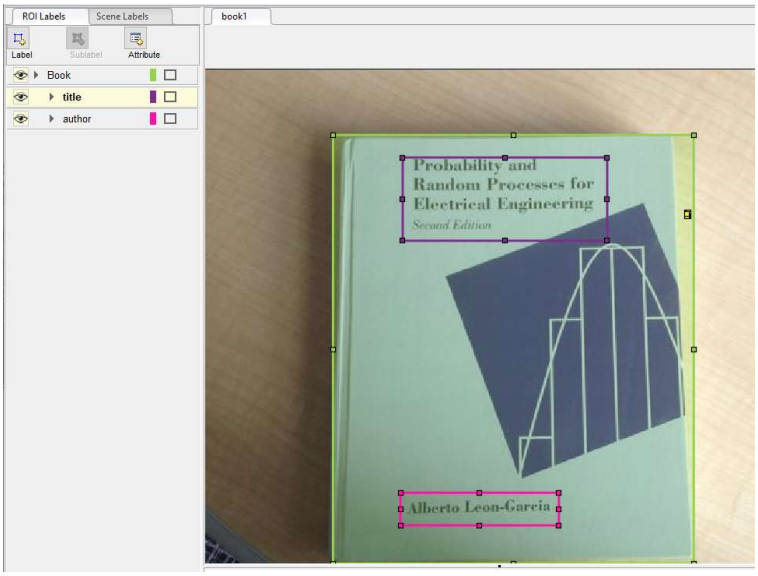
b. Adding Attributs.

Next we add attributes to the image in this case I decided to add the edition of the book as an attribute of numeric value this way I can know the edition of each book.



c. Adding Sub-Labels.

Sub labels for the title and author were added. And given different colors.



Note 1 :Inside the session open imageLabelingSessionBooks.mat to see the labels and save it as gtruthBooks.mat and export as gtruthbooks.mat to update the gtruthfile.

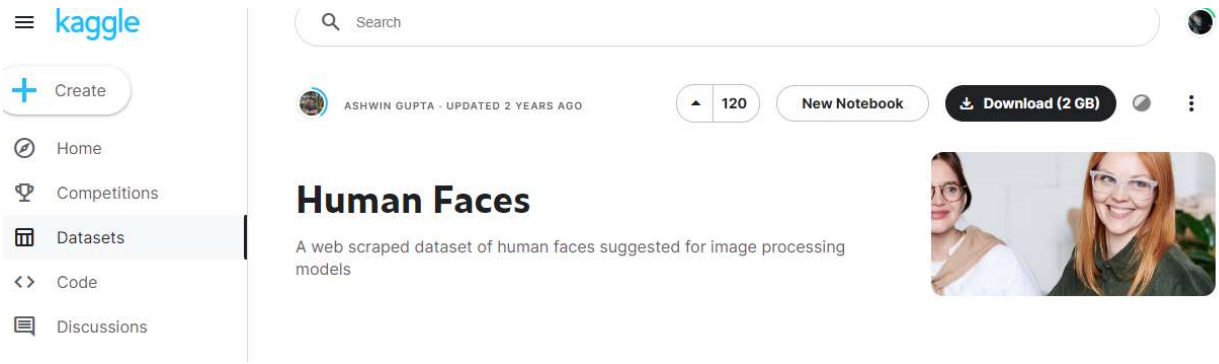
Once several images are label they can be exported to the workspace as a gTruth or to a file in .mat format. In this case I decided to export it to a file to make sure the file is persistent.

2. Preparing my own experiment

Now that I understand the basics of the image labeler app, I decided to perform my own experiment using a foreign data set.

a. Acquiring a Faces Dataset

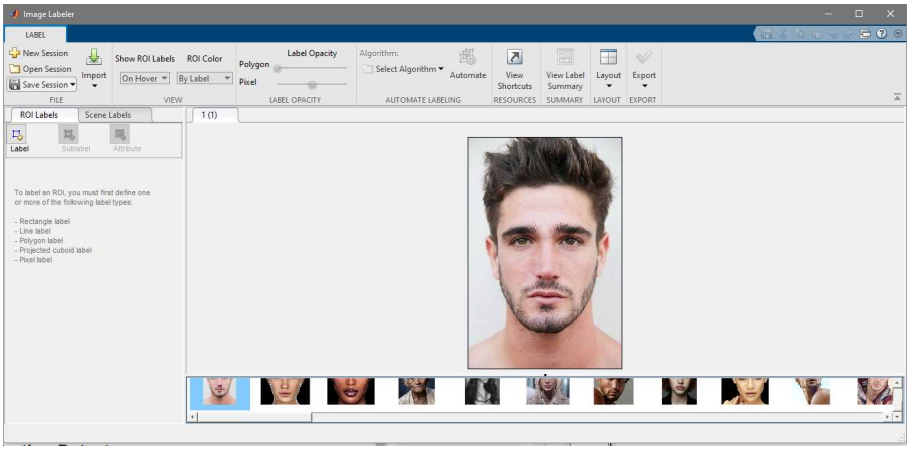
In this case I decided to use the faces data set provided by Kagge.com. This includes over 7000 faces from various ages and genders, however in this case I decided to use only 50 for my purposes. This would serve as a means of training an object detector to recognize parts of a face. I decided to use the mouth.



b. Loading the images.

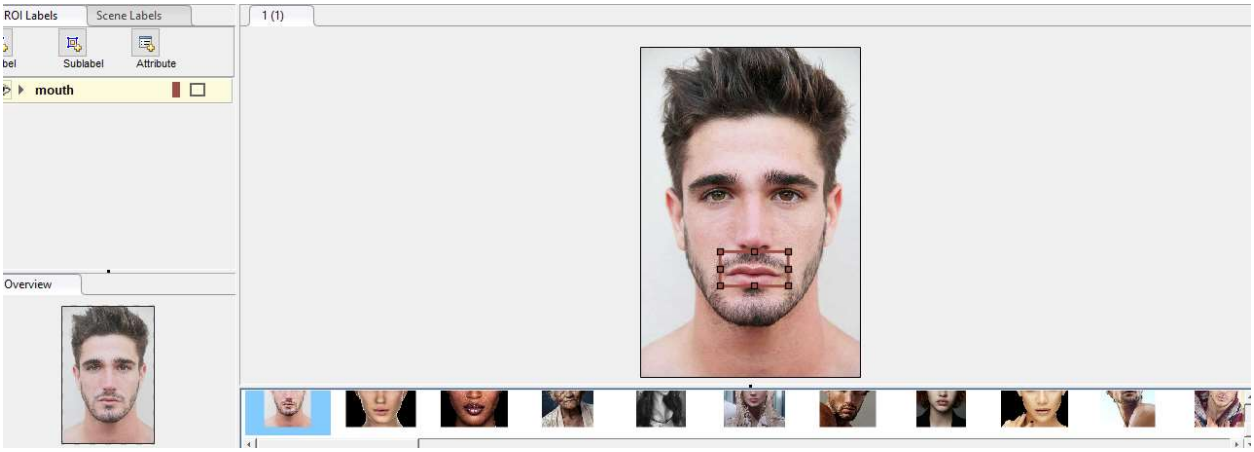
I saved the images to a faces file and loaded them into a variable using the image data store. Then this data store to open the image labeler.

```
fData=imageDatastore("Faces");
imageLabeler(fData);
```



c. Labeling the Mouth

Since I wanted to use the mouth I labeled all the mouths in each picture. I would have done all features of the face including eyes nose and ears, but I was not sure how long the training process would be, so I decided to be safe and only use one feature.



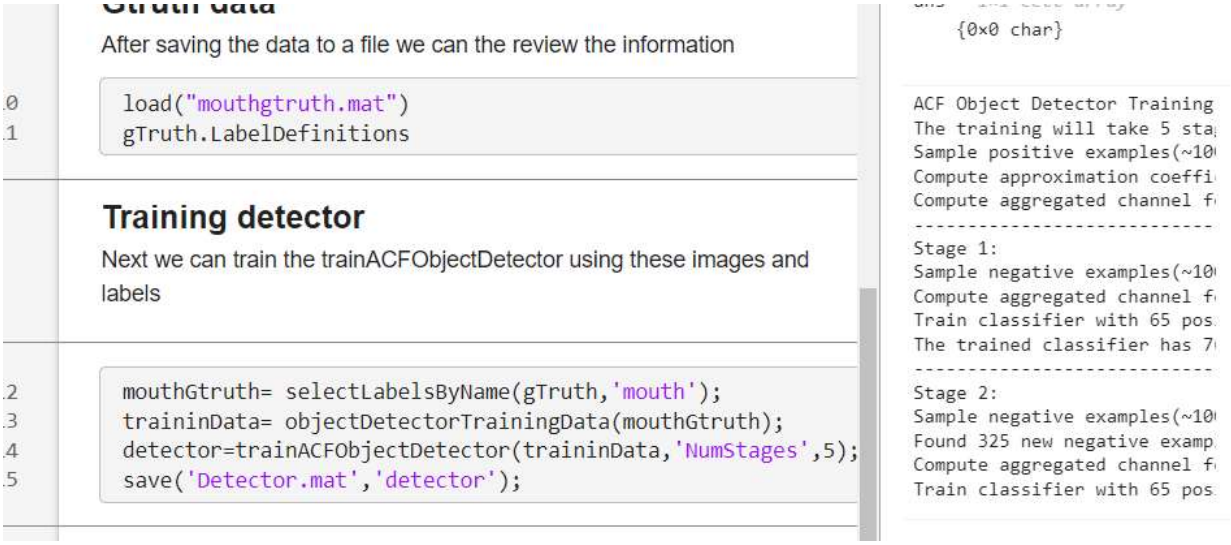
note 2: Open the imageLabelingSessionFaces.mat in the image labeler app to see the mouths and export the gtruth as mouthgtruth.mat to update the gtruth file

d. Training the Detector

Once all the mouths have been I decided to use the ACFObjectDetector to see if it can be trained to recognize the mouths on a face. Using the documentation provided from the MATLAB on the ACFObjectDetector I designed a script that would load the saved gtruth create the object detector training data and train an ACFObjectDetector. Which was saved to an .mat file.

```
load("mouthgtruth.mat")
mouthGtruth= selectLabelsByName(gTruth,'mouth');
traininData= objectDetectorTrainingData(mouthGtruth);
detector=trainACFObjectDetector(traininData,'NumStages',5);
save('Detector.mat','detector');
```

I was unsure how stages there should be, so I decided to use the same number as in the documentation.



e. Testing the Detector

Once the training is complete I decided to select more images from the original data set and test the mouth detector on other image files. I also inserted annotation to show what parts of the image are labeled as mouth.

```
load("Detector.mat")
img=imread("TestData\1 (305).jpg");
[bboxes,scores]=detect(detector,img)
for i = 1:length(scores)
   annotation = sprintf('Mouth Confidence = %.1f',scores(i));
   img = insertObjectAnnotation(img,'rectangle',bboxes(i,:),annotation);
end

imshow(img)
```

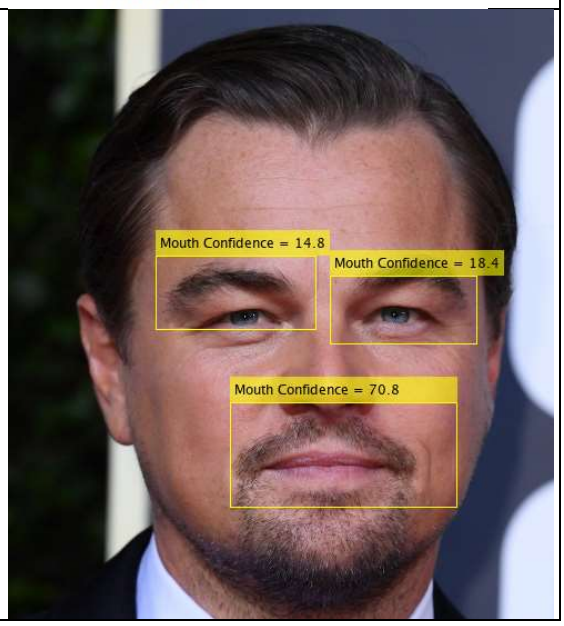
We can see that though sometimes the detector properly detects the mouth it often mistakes the eyes as a mouth if they are not open wide enough. This is an error that can be solved with a larger dataset and specific labels for the eyes. We can also see the relative confidence values for each image as well as the coordinates for the bounding boxes. We can use these in a for loop to annotate the images.

```
bboxes = 1×4
     154   270   323   150

scores = 66.8546
```

```
bboxes = 3×4
     321   247   133    62
     164   229   145    67
     231   361   205    95

scores = 3×1
     18.4106
     14.8030
     70.7803
```



Conclusion

The image labeler is a powerful tool that can be used to label images for different kinds of learning algorithms. However, from this small experiment I learned that choosing the correct kind of training algorithms and proper labeling of the features of an image is important to getting consistent results. If I had more time, I would properly label each feature of the face as using polygons instead of bounding boxes to promote accuracy and label a significantly larger number of faces instead of the 50 I had chosen. It would have also been better to only annotate the parts of the face with the largest confident score. This would have avoided the error that my detector often makes with eyes and mouths. Also, it would have been better to have the mouth as a sub-label to a face label and have used a proper deep learning approach to the training instead of an object detector. This detector could then be used to label all the other images in the large data set using automation. This could then be extended to other features such as nose, mouth, eyes and the entire face with a gender attribute.

Resources

- https://www.mathworks.com/help/vision/ref/trainacfobjectdetector.html (object Detector)
- https://www.mathworks.com/help/vision/ug/get-started-with-the-image-labeler.html (image labeler)
- https://www.kaggle.com/datasets/ashwingupta3012/human-faces (database for human faces)