Labeling and gTruth

https://www.mathworks.com/help/vision/ug/get-started-with-the-image-labeler.html

1) Download MATAB and run on your machine to get familiar with code – the path problem we saw is the gTruth data paths, which are generated by imageLabler – I would load the "*session.mat" files and Export gTruth as a "table" in your environment
   a. Perhaps there is a better solution to the gTruth pathing issue, but I do not know how to change – this work-around has worked for the group so far
   b. Otherwise, the data can just be unzipped in place as long as the paths in gTruth match where the files are
2) Had to recall how we made the cow gTruth – we really somehow made "cowDatasetGroundTruth.mat", which can be imported by just double clicking and becomes "gTruth" as a MATLAB variable.
   a. We repeated this for eleDatasetGroundTruth.mat and it works the same.
   b. This was based on vehicleDatasetGroundTruth.mat, the starting example.
3) gTruth table should be loaded (or saved as a truth table)

Use the truth models in:

1) vehicleTruth
2) satVehicleTruth
3) satElephantTruth
4) satCowTruth
5) elephTruth
6) cowTruth

Satellite Locations of interest

Location	Lat, Long	Worldview-3		Google Earth		Google Maps	

CSU farm, Beef Unit	- 39.6956963, -121.8298722	nwLat = 39.73227800;
		nwLong = -121.86661600;
		seLat = 39.68734000;
		seLong = -121.80820000;
			C	39.6956963, -121.8298722
				Pro: 8/1998 – 5/2021
					C	39.6956963, -121.8298722
						(cows visible)
						Link
							C

Addo Elephant Reserve Watering hole, (Main), (Hapoor Drinking Hole)	-33.498693, 25.734161 (-33.4834244, 25.7483911) (-33.5025481, 25.7397333)	None	X	-33.5023548, 25.7381934 Pro: 11/2020, 1/2019, 12/2018, 10/2018, 3/2018, 3/2016 Best Link	E	-33.4991619, 25.7330841 Link	X
Medikwe	-24.7604298, 26.2689526	nwLat = -24.73795899;	E	-24.7604528, 26.2745408	X	-24.7604528, 26.2745408	X

		nwLong = 26.22669199; seLat = -24.78291099; seLong = 26.27619499;		Pro: 1/2020 Link		Link	
Rietvlei	-25.882449, 28.2616808	nwLat = -25.85860300; nwLong = 28.26369800; seLat = -25.90355600; seLong = 28.31366200;	**U**	-25.8838398, 28.26481284 Link	**X**	-25.882449, 28.2616808 Link	**X**
Prescott	34.841986, -112.3790217	nwLat = 34.84198599; nwLong = -112.37683299; seLat = 34.80400800; seLong = -112.31203099;	**?**	34.841986, -112.3790217 Link	X	34.8419904, -112.4140406 **Link**	X
Coleridge	-33.349866, 26.6157291	nwLat = -33.33009800; nwLong = 26.59009000; seLat = -33.37182500; seLong = 26.64802600;	**V**	-33.3516649, 26.61389674 (Look on paved road for vehicles) Link	**V**	-33.349866, 26.6135404 Link	X

C=cows, E=elephant, V=vehicles, U=Unknown large animals (Elephants?), X=no animals

MATLAB Deep Learning Examples

1. SSD - https://www.mathworks.com/help/vision/ug/object-detection-using-single-shot-detector.html

```
openExample('deeplearning_shared/ObjectDetectionUsingSSDDeepLearningExample')
```

2. RCNN - https://www.mathworks.com/help/vision/ug/object-detection-using-faster-r-cnn-deep-learning.html

```
openExample('deeplearning_shared/DeepLearningFasterRCNNObjectDetectionExample'
)
```

3. YOLO V2 - https://www.mathworks.com/help/vision/ug/train-yolo-v2-network-for-vehicle-detection.html

```
openExample('vision/TrainYOLOV2NetworkForVehicleDetectionExample')
```

4. YOLO V3 - https://www.mathworks.com/help/vision/ug/object-detection-using-yolo-v3-deep-learning.html

```
openExample('deeplearning_shared/ObjectDetectionUsingYOLOV3DeepLearningExample
')
```

```
doTraining = true;
```

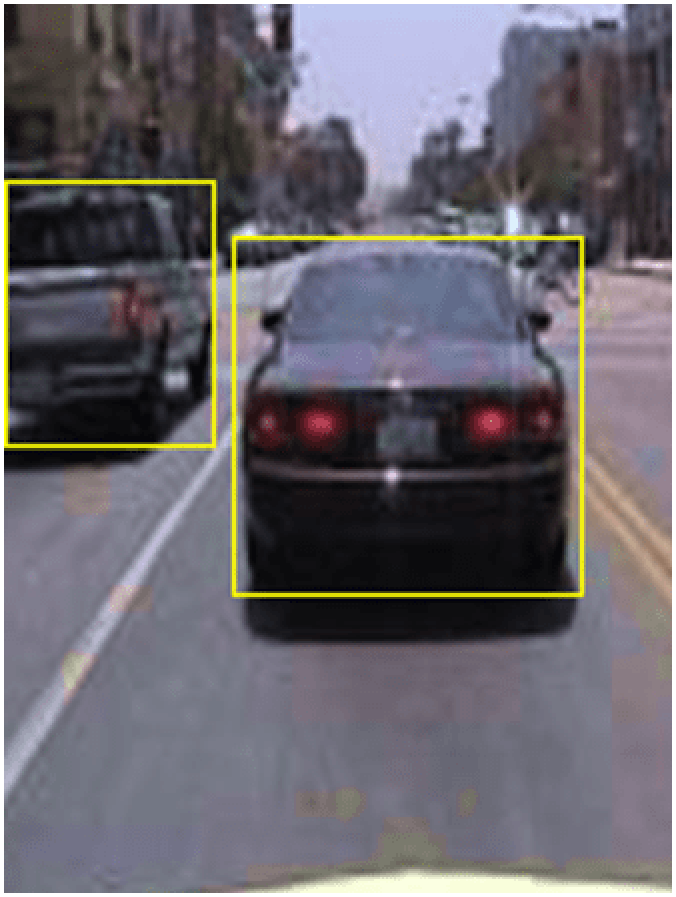
```
C:\Program Files\MATLAB\R2020a\examples\deeplearning_shared\data
```

vehicleDatasetGroundTruth.mat	1/7/2019 10:38 AM	Microsoft Access ...	5 KB
vehicleDatasetImages.zip	1/7/2019 10:38 AM	Compressed (zipp...	3,577 KB

We need new GroundTruth and new DatasetImages for our objects of interest.

https://www.mathworks.com/help/deeplearning/ug/data-sets-for-deep-learning.html

Vehicle



The Vehicle data set consists of 295 images containing one or two labeled instances of a vehicle. This small data set is useful for exploring the YOLO-v2 training procedure, but in practice, more labeled images are needed to train a robust detector.

The images are of size 720-by-960-by-3.

Extract the Vehicle data set. Set `dataFolder` to the location of the data.

```
filename =
'vehicleDatasetImages.zip';

dataFolder =
fullfile(tempdir,'vehicleImages');
if ~exist(dataFolder,'dir')
    unzip(filename,tempdir);
end
```

Load the data set as a table of file names and bounding boxes from the extracted MAT file and convert the file names to absolute file paths.

```
data =
load('vehicleDatasetGroundTruth.mat
');
vehicleDataset =
data.vehicleDataset;
```

```
vehicleDataset.imageFilename =
fullfile(tempdir,vehicleDataset.ima
geFilename);
```

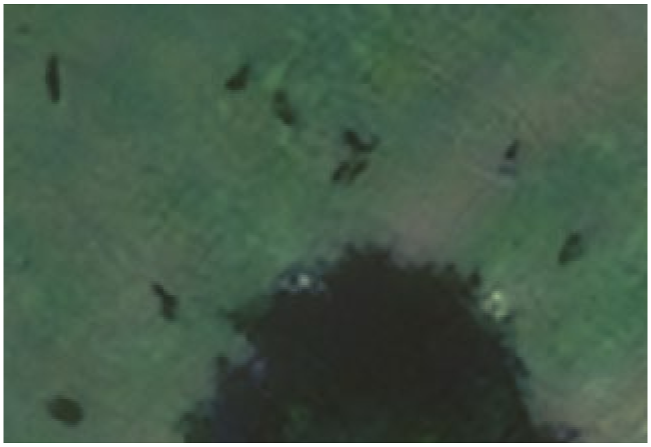
Create an image datastore containing the images and a box label datastore containing the bounding boxes using the `imageDatastore` and `boxLabelDatastore` functions, respectively. Combine the resulting datastores using the `combine` function.

```
filenamesImages =
vehicleDataset.imageFilename;
tblBoxes =
vehicleDataset(:,'vehicle');

imds =
imageDatastore(filenamesImages);
blds = boxLabelDatastore(tblBoxes);

cds = combine(imds,blds);
```

For an example showing how to process this data for deep learning, see Object Detection Using YOLO v2 Deep Learning.

Images with just one target of interest -   find a way to click on an object I see in an image, and then extract a bounding box sub-image of an exact size.

E.g., here’s a bunch of cows, I want to click on the center of one, and extract a square area of say 30x30 pixels around it – adjustable, but generally nxn square.



11 cows, I want to separate into 11 images containing just one cow.

This is to create my training and validation sets for each animal type with “known” targets.

I see things like imcrop, but I don’t want odd-sized images if possible, and I don’t really want to draw rubber-band boxes over and over, I just want to click, extract, imshow, imwrite, over and over.

Google Earth Large Animal Images – References

1. https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=&cad=rja&uact=8&ved=2ahUKEwiQ0_m0punzAhXAGTQIHUuEBO0QFnoECAcQAQ&url=https%3A%2F%2Fearth.google.com%2Fweb%2Fdata%3DCiQSIhIgNTk5NDhiMjMwYjQ2MTFlNzhlMDQxNWY2OWQzNDJmYmY&usg=AOvVaw2iDxih9-1_3ffdo2ps1DTf

2. https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=&cad=rja&uact=8&ved=2ahUKEwiQ0_m0punzAhXAGTQIHUuEBO0QFnoECAYQAQ&url=https%3A%2F%2Fearth.google.com%2Fweb%2Fdata%3DCiQSIhIgMmFjODk3NzA3OGIzMTFlOTgyY2E0ZDFmNmYyMjAxNmE&usg=AOvVaw1140tnIG5Li4YdEAG0UJAn

3. https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=&cad=rja&uact=8&ved=2ahUKEwiQ0_m0punzAhXAGTQIHUuEBO0QFnoECB4QAQ&url=https%3A%2F%2Fwww.gearthblog.com%2Fblog%2Farchives%2F2006%2F08%2Fsee_african_ani.html&usg=AOvVaw3rV2kmvsdIrCWVefa9J7Ek

MATLAB version and toolboxes used

```
>> ver

--------------------------------------------------------------------------------------
---------------

MATLAB Version: 9.8.0.1451342 (R2020a) Update 5

MATLAB License Number: 40937346

Operating System: Microsoft Windows 10 Pro Version 10.0 (Build 18362)

Java Version: Java 1.8.0_202-b08 with Oracle Corporation Java HotSpot(TM) 64-Bit
Server VM mixed mode

--------------------------------------------------------------------------------------
---------------

MATLAB                                                Version 9.8           (R2020a)

Simulink                                              Version 10.1          (R2020a)

Aerospace Blockset                                    Version 4.3           (R2020a)

Aerospace Toolbox                                     Version 3.3           (R2020a)

Automated Driving Toolbox                             Version 3.1           (R2020a)

Computer Vision Toolbox                               Version 9.2           (R2020a)

Control System Toolbox                                Version 10.8          (R2020a)

DSP System Toolbox                                    Version 9.10          (R2020a)

Deep Learning Toolbox                                 Version 14.0          (R2020a)

Embedded Coder                                        Version 7.4           (R2020a)

GPU Coder                                             Version 1.5           (R2020a)

Image Acquisition Toolbox                             Version 6.2           (R2020a)

Image Processing Toolbox                              Version 11.1          (R2020a)

MATLAB Coder                                          Version 5.0           (R2020a)

Parallel Computing Toolbox                            Version 7.2           (R2020a)

Partial Differential Equation Toolbox                 Version 3.4           (R2020a)

ROS Toolbox                                           Version 1.1           (R2020a)

Robotics System Toolbox                               Version 3.1           (R2020a)

Sensor Fusion and Tracking Toolbox                    Version 1.3           (R2020a)
```

```
Signal Processing Toolbox                             Version 8.4         (R2020a)
Statistics and Machine Learning Toolbox               Version 11.7        (R2020a)
Symbolic Math Toolbox                                 Version 8.5         (R2020a)
Vehicle Dynamics Blockset                             Version 1.4         (R2020a)
>>
```