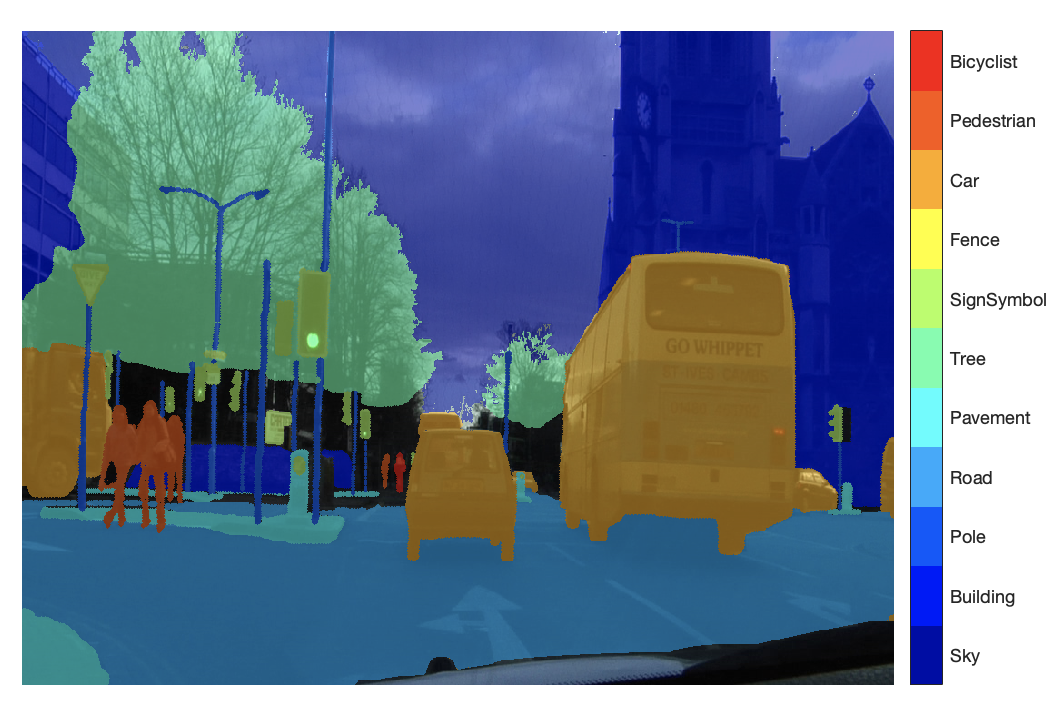
# Image Processing and Analysis

# Assignment 2: Semantic image segmentation using deep learning

Prepared by Prof. Oge Marques, PhD, Professor of Engineering and Computer Science at   
Florida Atlantic University - <http://www.eng.fau.edu/directory/faculty/marques/>

**Goal:** Build and evaluate solutions for segmenting images using deep learning architectures.



**Learning objectives:**

* Learn how to implement an image segmentation workflow in MATLAB®
* Learn how to implement and evaluate contemporary (deep-learning-based) semantic image segmentation techniques in MATLAB
* Get acquainted with representative datasets and problems in image segmentation

**Starter package**

* **Dataset**:
  + CamVid <http://web4.cs.ucl.ac.uk/staff/g.brostow/MotionSegRecData/> [1]
* **MATLAB starter code**: **semantic\_segmentation\_dl\_STARTER.mlx** (available online)

**Instructions:**

* Document all your findings, steps, conclusions, lessons learned, insights, etc. in your **report** *(think of it as a “lab notebook”)*
* **Add your answers to the numbered questions to your report**

**Procedure:**

1. (OPTIONAL) Watch the overview video at: <https://www.mathworks.com/videos/semantic-segmentation-overview-1510858047780.html>
2. Download MATLAB starter code and add to your working folder, adjusting the MATLAB path if necessary.
3. Run "Part 1" of the starter code and ensure that it works as intended. This is essentially the workflow for semantic image segmentation using a convolution-deconvolution neural network created by you and trained from scratch.
4. Answer the questions below:

**QUESTION 1:** What is the difference between an imageDatastore

and a pixelLabelDatastore in MATLAB?

**QUESTION 2:** Explain in your words the meaning of the segmentation metrics used to evaluate the performance of the first neural network, namely: global accuracy, class accuracy, IoU, weighted IoU, and BF score.

1. **Your turn!**Write code to build and train a second network (call it net2). Compare the metrics of net2 versus net. Which one performed better? Why?
2. (OPTIONAL) **Your turn!**Improve the original network (net) to handle *class imbalance*, call the resulting solution net3, and compare the metrics of net3 versus net. Which one performed better? Why?
3. (OPTIONAL) **Your turn!**Using net3 as a starting point, implement the *Tversky loss* and use it to build a custom pixel classification layer. Call this solution, net4. Compare the metrics of net4 against its three variants. Which one performed better? Why?
4. Run "Part 2" of the starter code and ensure that it works as intended. This is essentially the workflow for semantic image segmentation using a pretrained network and a realistic dataset (CamVid).

**QUESTION 3:** What are some of the main characteristics of the CamVid dataset? Think about size, historical context, images, labels, etc.

**QUESTION 4:** How are undefined or void labels handled in the example code? Would you suggest handling it differently, perhaps? Why (not)?

**QUESTION 5:** What are the main insights provided by plotting the histogram of the distribution of pixel counts by class (step 5)?

**QUESTION 6:** If you could refactor/improve the auxiliary function partitionCamVidData, what would you do differently?

**QUESTION 7:** How is the issue of class imbalance handled in the example code? Would you suggest handling it differently, perhaps? Why (not)?

**QUESTION 8:** How is image data augmentation implemented in the example code? Would you suggest doing it differently, perhaps? If so, how exactly?

**QUESTION 9:** During quantitative evaluation of the segmentation results, which classes show the best performance across all metrics? Do these match your expectations? Why (not)?

**QUESTION 10:** During quantitative evaluation of the segmentation results, which classes show the greatest discrepancy between accuracy and IoU? How do you explain that?

1. (OPTIONAL) **Your turn!**Repeat (steps 2.3 through 2.6 of) "Part 2" using different pretrained networks, training options, data augmentation options, and/or metrics.
2. Prepare your **report**, with all relevant plots, code snippets, numerical values and – most importantly – your insights and lessons learned.

**References**

[1] Brostow, G. J., J. Fauqueur, and R. Cipolla. "Semantic object classes in video: A high-definition ground truth database." *Pattern Recognition Letters*. Vol. 30, Issue 2, 2009, pp 88-97.

[2] Salehi, Seyed Sadegh Mohseni, Deniz Erdogmus, and Ali Gholipour. "Tversky loss function for image segmentation using 3D fully convolutional deep networks." International Workshop on Machine Learning in Medical Imaging. Springer, Cham, 2017.

[3] Chen, Liang-Chieh et al. “Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation.” ECCV (2018).

**Reference links**:

* <https://www.mathworks.com/help/images/image-segmentation.html>
* <https://www.mathworks.com/help/vision/examples/multispectral-semantic-segmentation-using-deep-learning.html>
* <https://www.mathworks.com/help/vision/examples/semantic-segmentation-using-deep-learning.html>
* <https://www.mathworks.com/help/vision/ref/deeplabv3pluslayers.html>
* <https://www.mathworks.com/help/vision/ref/semanticseg.html>
* <https://www.mathworks.com/help/vision/ug/define-custom-pixel-classification-layer-with-tversky-loss.html>
* <https://www.mathworks.com/help/vision/ug/getting-started-with-semantic-segmentation-using-deep-learning.html>
* <https://www.mathworks.com/videos/semantic-segmentation-overview-1510858047780.html>