

# OpenCV

Aug 29<sup>th</sup>

# Tools and Software

- Numpy – numerical computing (matrices)
- Matplotlib – basic graphing + plotting
- OpenCV – Computer Vision library

# Numpy

- Main numerical computing library for python
- Images represented by 2D or 3D numpy arrays
- Typically images are 8-bit, which means pixel values range from 0 (dark) to 255 (light).  
Scientific images are often higher-precision (e.g., 16-bit).
- Designed look like Matlab syntax
- C backend means it's fast

# Numpy

- Main numerical computing library for python
  - Images represented by 2D or 3D numpy arrays
  - Typically images are 8-bit, which means pixel values range from 0 (dark) to 255 (light).  
Scientific images are often higher-precision (e.g., 16-bit).
  - Designed look like Matlab syntax
  - C backend means it's fast
- Jupyter

# OpenCV

- OpenCV (Open Source Computer Vision Library) is an open source computer vision and machine learning software library.
- ~2500 optimized algorithms

# OpenCV Algorithm

- Detect and recognize faces,
- Identify objects,
- Classify human actions in videos,
- Track camera movements,
- Track moving objects,
- Extract 3D models of objects,
- Produce 3D point clouds from stereo cameras,
- Stitch images together to produce a high resolution image of an entire scene,
- Find similar images from an image database,
- Remove red eyes from images taken using flash,
- And etc.

# OpenCV Examples

1. Load and Display Image
  1. Image Shape
  2. Converting to grey scale/color spaces
2. Create Images
3. Access Pixels
  1. ROI
  2. Change Brightness
  3. Invert
4. Writing/Drawing on images

# Matplotlib

- Plotting library
- Important functions: `imread`, `imsave`, `imshow`, `plot`
- Be careful with the coordinates!
  - `imshow` puts the y axis origin at the top of the frame (i.e., the y axis is “upside down”)
  - Coordinates are usually represented as (x, y), but matrices are indexed (row, col)