EECS 4422/5323 Lab 1: Resources

Presented by: Abdullah Abuolaim

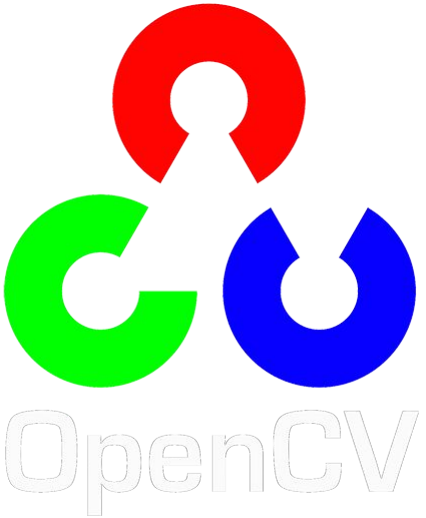
Outline

This lab will introduce and review basic usage of two useful resources:

- OpenCV in Python
- LaTeX

OpenCV

- Widely used open library of computer vision methods and tools
- Very useful for rapid prototyping
- Good starting benchmark for comparing new methods
- Large user base means most popular functions have been heavily stress-tested
- Large corpus of documentation and tutorials, including official tutorials available on the OpenCV website



A Few Things to Note about OpenCV

- As a large open source project, there are many different versions and backward compatibility is not always enforced
  - Keep in mind what version documentation and tutorials are written for!
  - It is useful for others to note what version of OpenCV you used for any projects you release which rely on it
- For projects with stringent performance requirements, OpenCV is not always the best solution (you will likely be able to write purpose-built code which is more computationally efficient)

Using OpenCV

- The default OpenCV package (version 4.1.1.26) should be available under Linux for all EECS students
- You can check this by opening a terminal and running `ipython`
- In the python environment launched by `ipython`, type the command `import cv2`
- This import command should execute without error; please let me know if you have any problems

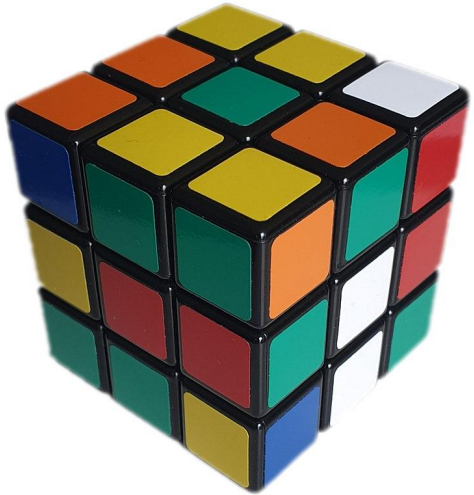
Installing OpenCV on your own system

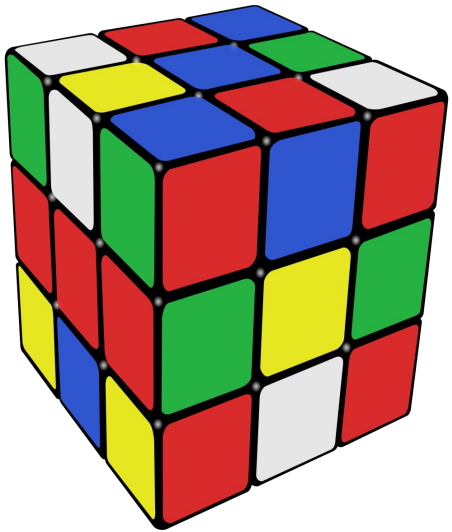
There are a number of ways to install OpenCV.

- On a system with Python available using the `pip` package manager, you can install OpenCV with the command `pip install opencv-python`
- On a system using Anaconda you can install OpenCV with the command `conda install -c conda-forge opencv`
- If you have further questions or concerns about installing OpenCV, let us know

OpenCV Exercise

In this lab, you will be experimenting with loading and displaying images and manipulating colour channels. We will use two example images for this, a photorealistic Rubik's cube (left) and a computer generated Rubik's cube (right):





Tasks for OpenCV Tutorial

- Download the images from the course website on the Schedule page under the Lab Schedule at the bottom of the page
- Using OpenCV, load each image as a colour image and display it on your screen
  - `cv2.imread(), cv2.imshow()`
- Load each image directly as a grayscale image and display it on your screen
- For the colour images, display each individual colour channel.
  - How does the appearance of each channel compare to the grayscale images? Can you tell how the channels are ordered in OpenCV?
- For each cube, can you create a binary mask for each colour?
  - Does your code translate between cubes? Why or why not? Are there any other challenges you faced in isolating colour pixels?

LaTeX

A powerful document preparation system, LaTeX is essentially a programming language for document writing. Benefits include:

- Nicely formatted, easy to write equations
- Automatic generation of section, figure, table, and equation numbering
- Reference management through BibTeX; so long as your BibTeX entries are correct, no need to mess with reference formatting!
- Easy to include cross-references within your document

LaTeX Downsides

As useful and wonderful as LaTeX is, there are some downsides

- Steep learning curve, poor error messaging
  - Common strategies include commenting and uncommenting lines to find the error
- Tables are a pain
- Highly flexible, distributed development means that sometimes packages will interfere or become deprecated

Useful LaTeX Tools

- Very good and comprehensive wiki: https://en.wikibooks.org/wiki/LaTeX
- LyX, a front-end to ease you into the world of LaTeX: https://www.lyx.org/
- Some potentially useful LaTeX environments:
  - TeXStudio: https://www.texstudio.org/
    - Cross-platform, fairly clean
  - Overleaf: https://www.overleaf.com/
    - Online editor, good for collaborations, requires an account
  - Atom, with `atom-latex` package: https://ide.atom.io/
    - If you like all your development in the same environment, Atom is an option

A Good Strategy

It's daunting to start LaTeX from scratch. Thankfully, you don't have to! There are many available templates, and conferences or journals often have an official template available. Find one you like the look of, download it, and start composing in that.

A few good options with fairly different appearance include:

- British Machine Vision Conference
- Computer Vision and Pattern Recognition Conference

Download one of these templates and make sure you can compile it, then try changing it to become the skeleton for your future project report.