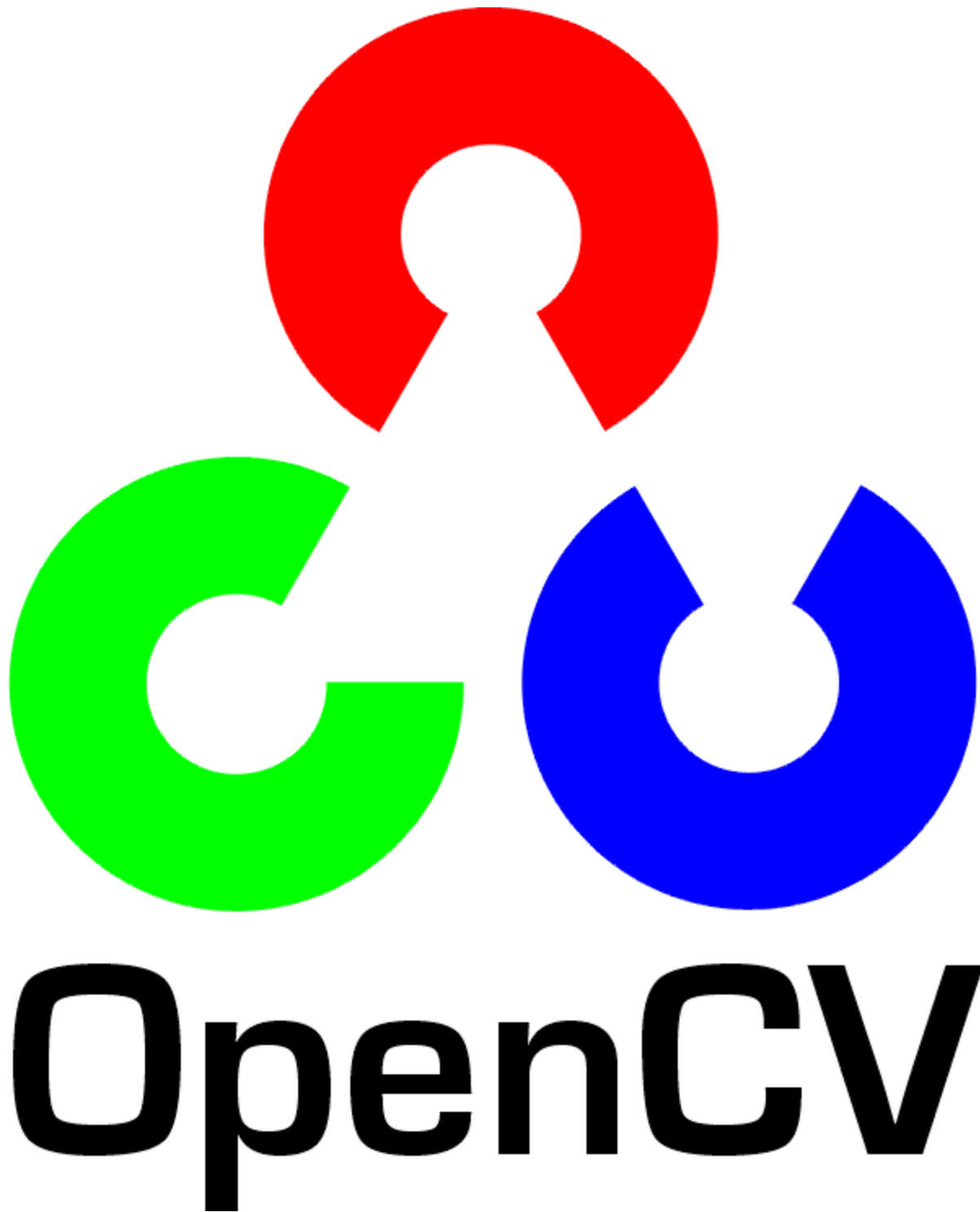


## OpenCV

Before we jump into coding our own convolutional kernels/filters, I'll introduce you to a new library that will be useful to use when dealing with computer vision tasks, such as image classification: OpenCV!



[OpenCV logo](#)

OpenCV is a **computer vision and machine learning software library** that includes many common image analysis algorithms that will help us build custom, intelligent computer vision applications. To start with, this includes tools that help us process images and select areas of

interest! The library is widely used in academic and industrial applications; from [their site](#), OpenCV includes an impressive list of users: *“Along with well-established companies like Google, Yahoo, Microsoft, Intel, IBM, Sony, Honda, Toyota that employ the library, there are many startups such as Applied Minds, VideoSurf, and Zeitera, that make extensive use of OpenCV.”*

So, note, how we import cv2 in the next notebook and use it to create and apply image filters!

## Notebook: Custom Filters

The next notebook is called `custom_filters.ipynb`.

To open the notebook, you have two options:

- Go to the next page in the classroom (recommended).
- Clone the repo from [Github](#) and open the notebook **custom\_filters.ipynb** in the **convolutional-neural-networks > conv-visualization** folder. You can either download the repository with git clone `https://github.com/udacity/deep-learning-v2-pytorch.git`, or download it as an archive file from [this link](#).

## Instructions

- Define your own convolutional filters and apply them to an image of a road
- See if you can define filters that detect horizontal or vertical edges

This notebook is meant to be a playground where you can try out different filter sizes and weights and see the resulting, filtered output image!