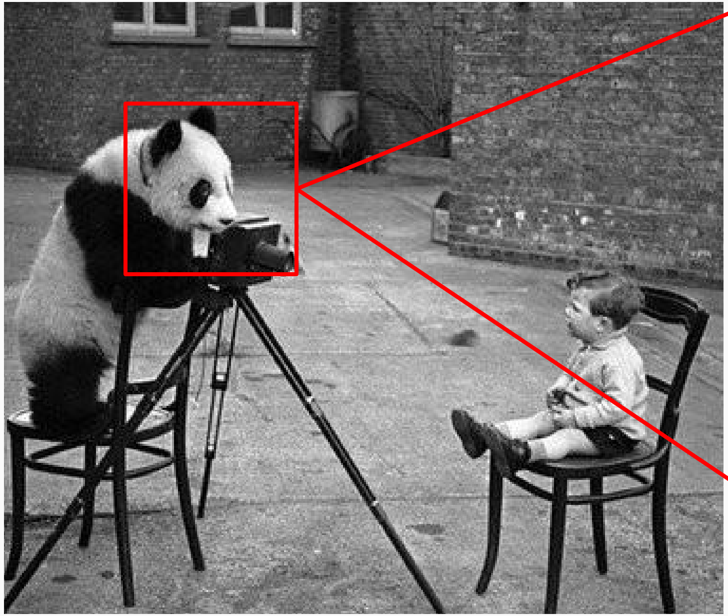
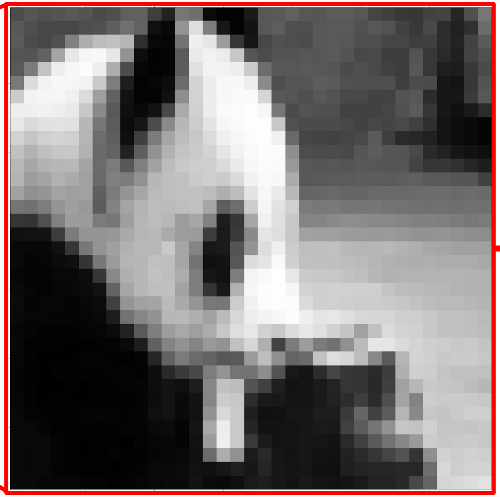
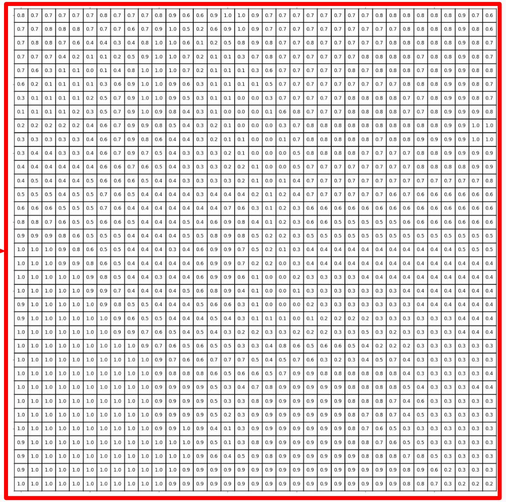
Representing Images

Why is Computer Vision so Hard?

Why is Computer Vision so Hard?





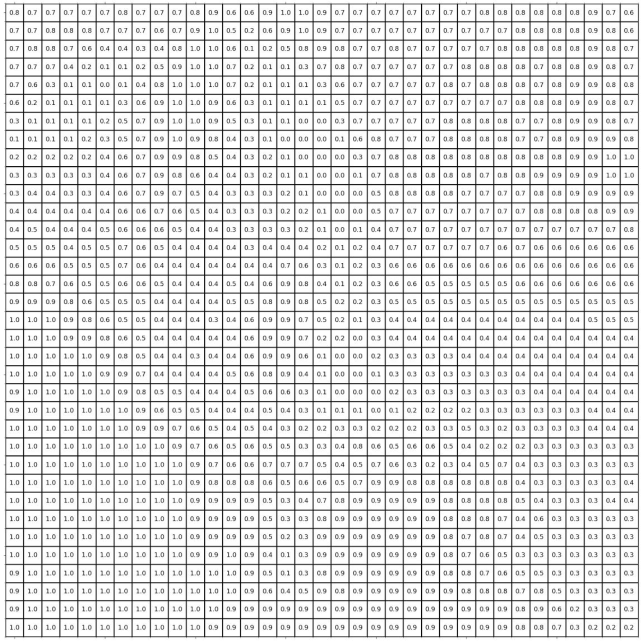


0.8	0.7	0.7	0.7	0.7	0.7	0.8	0.7	0.7	0.7	0.8	0.9	0.6	0.6	0.9	1.0	1.0	0.9	0.7	0.7	0.7	0.7	0.7	0.7	0.7	0.7	0.8	0.8	0.8	0.8	0.8	0.8	0.9	0.7	0.6
0.7	0.7	0.8	0.8	0.8	0.7	0.7	0.7	0.6	0.7	0.9	1.0	0.5	0.2	0.6	0.9	1.0	0.9	0.7	0.7	0.7	0.7	0.7	0.7	0.7	0.7	0.7	0.7	0.8	0.8	0.8	0.8	0.9	0.8	0.6
0.7	0.8	0.8	0.7	0.6	0.4	0.4	0.3	0.4	0.8	1.0	1.0	0.6	0.1	0.2	0.5	0.8	0.9	0.8	0.7	0.7	0.8	0.7	0.7	0.7	0.7	0.7	0.8	0.8	0.8	0.8	0.8	0.9	0.8	0.7
0.7	0.7	0.7	0.4	0.2	0.1	0.1	0.2	0.5	0.9	1.0	1.0	0.7	0.2	0.1	0.1	0.3	0.7	0.8	0.7	0.7	0.7	0.7	0.7	0.7	0.8	0.8	0.8	0.8	0.7	0.8	0.8	0.9	0.8	0.7
0.7	0.6	0.3	0.1	0.1	0.0	0.1	0.4	0.8	1.0	1.0	1.0	0.7	0.2	0.1	0.1	0.1	0.3	0.6	0.7	0.7	0.7	0.7	0.7	0.8	0.7	0.8	0.8	0.8	0.7	0.8	0.9	0.9	0.8	0.8
0.6	0.2	0.1	0.1	0.1	0.1	0.3	0.6	0.9	1.0	1.0	0.9	0.6	0.3	0.1	0.1	0.1	0.1	0.5	0.7	0.7	0.7	0.7	0.7	0.7	0.7	0.7	0.7	0.8	0.8	0.8	0.9	0.9	0.8	0.7
0.3	0.1	0.1	0.1	0.1	0.2	0.5	0.7	0.9	1.0	1.0	0.9	0.5	0.3	0.1	0.1	0.0	0.0	0.3	0.7	0.7	0.7	0.7	0.7	0.8	0.8	0.8	0.8	0.7	0.7	0.8	0.9	0.9	0.8	0.7
0.1	0.1	0.1	0.1	0.2	0.3	0.5	0.7	0.9	1.0	0.9	0.8	0.4	0.3	0.1	0.0	0.0	0.0	0.1	0.6	0.8	0.7	0.7	0.7	0.8	0.8	0.8	0.8	0.7	0.7	0.8	0.9	0.9	0.9	0.8
0.2	0.2	0.2	0.2	0.2	0.4	0.6	0.7	0.9	0.9	0.8	0.5	0.4	0.3	0.2	0.1	0.0	0.0	0.0	0.3	0.7	0.8	0.8	0.8	0.8	0.8	0.8	0.8	0.8	0.8	0.8	0.9	0.9	1.0	1.0
0.3	0.3	0.3	0.3	0.3	0.4	0.6	0.7	0.9	0.8	0.6	0.4	0.4	0.3	0.2	0.1	0.1	0.0	0.0	0.1	0.7	0.8	0.8	0.8	0.8	0.8	0.7	0.8	0.8	0.9	0.9	0.9	0.9	1.0	1.0
0.3	0.4	0.4	0.3	0.3	0.4	0.6	0.7	0.9	0.7	0.5	0.4	0.3	0.3	0.3	0.2	0.1	0.0	0.0	0.0	0.5	0.8	0.8	0.8	0.8	0.7	0.7	0.7	0.7	0.8	0.8	0.9	0.9	0.9	0.9
0.4	0.4	0.4	0.4	0.4	0.4	0.6	0.6	0.7	0.6	0.5	0.4	0.3	0.3	0.3	0.2	0.2	0.1	0.0	0.0	0.5	0.7	0.7	0.7	0.7	0.7	0.7	0.7	0.7	0.8	0.8	0.8	0.8	0.9	0.9
0.4	0.5	0.4	0.4	0.4	0.5	0.6	0.6	0.6	0.5	0.4	0.4	0.3	0.3	0.3	0.3	0.2	0.1	0.0	0.1	0.4	0.7	0.7	0.7	0.7	0.7	0.7	0.7	0.7	0.7	0.7	0.7	0.7	0.7	0.8
0.5	0.5	0.5	0.4	0.5	0.5	0.7	0.6	0.5	0.4	0.4	0.4	0.4	0.3	0.4	0.4	0.4	0.2	0.1	0.2	0.4	0.7	0.7	0.7	0.7	0.7	0.7	0.6	0.7	0.6	0.6	0.6	0.6	0.6	0.6
0.6	0.6	0.6	0.5	0.5	0.5	0.7	0.6	0.4	0.4	0.4	0.4	0.4	0.4	0.4	0.7	0.6	0.3	0.1	0.2	0.3	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6
0.8	0.8	0.7	0.6	0.5	0.5	0.6	0.6	0.5	0.4	0.4	0.4	0.5	0.4	0.6	0.9	0.8	0.4	0.1	0.2	0.3	0.6	0.6	0.5	0.5	0.5	0.5	0.5	0.6	0.6	0.6	0.6	0.6	0.6	0.6
0.9	0.9	0.9	0.8	0.6	0.5	0.5	0.5	0.4	0.4	0.4	0.4	0.5	0.5	0.8	0.9	0.8	0.5	0.2	0.2	0.3	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5
1.0	1.0	1.0	0.9	0.8	0.6	0.5	0.5	0.4	0.4	0.4	0.3	0.4	0.6	0.9	0.9	0.7	0.5	0.2	0.1	0.3	0.4	0.4	0.4	0.4	0.4	0.4	0.4	0.4	0.4	0.4	0.4	0.5	0.5	0.5
1.0	1.0	1.0	0.9	0.9	0.8	0.6	0.5	0.4	0.4	0.4	0.4	0.4	0.6	0.9	0.9	0.7	0.2	0.2	0.0	0.3	0.4	0.4	0.4	0.4	0.4	0.4	0.4	0.4	0.4	0.4	0.4	0.4	0.4	0.4
1.0	1.0	1.0	1.0	1.0	0.9	0.8	0.5	0.4	0.4	0.3	0.4	0.4	0.6	0.9	0.9	0.6	0.1	0.0	0.0	0.2	0.3	0.3	0.3	0.3	0.4	0.4	0.4	0.4	0.4	0.4	0.4	0.4	0.4	0.4
1.0	1.0	1.0	1.0	1.0	0.9	0.9	0.7	0.4	0.4	0.4	0.4	0.5	0.6	0.8	0.9	0.4	0.1	0.0	0.0	0.1	0.3	0.3	0.3	0.3	0.3	0.3	0.3	0.4	0.4	0.4	0.4	0.4	0.4	0.4
0.9	1.0	1.0	1.0	1.0	1.0	0.9	0.8	0.5	0.5	0.4	0.4	0.4	0.5	0.6	0.6	0.3	0.1	0.0	0.0	0.0	0.2	0.3	0.3	0.3	0.3	0.3	0.3	0.3	0.4	0.4	0.4	0.4	0.4	0.4
0.9	1.0	1.0	1.0	1.0	1.0	1.0	0.9	0.6	0.5	0.5	0.4	0.4	0.4	0.5	0.4	0.3	0.1	0.1	0.1	0.0	0.1	0.2	0.2	0.2	0.2	0.3	0.3	0.3	0.3	0.3	0.3	0.4	0.4	0.4
1.0	1.0	1.0	1.0	1.0	1.0	1.0	0.9	0.9	0.7	0.6	0.5	0.4	0.5	0.4	0.3	0.2	0.2	0.3	0.3	0.2	0.2	0.2	0.3	0.3	0.5	0.3	0.2	0.3	0.3	0.3	0.3	0.4	0.4	0.4
1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	0.9	0.7	0.6	0.5	0.6	0.5	0.5	0.3	0.3	0.4	0.8	0.6	0.5	0.6	0.6	0.5	0.4	0.2	0.2	0.2	0.3	0.3	0.3	0.3	0.3	0.3
1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	0.9	0.7	0.6	0.6	0.7	0.7	0.7	0.5	0.4	0.5	0.7	0.6	0.3	0.2	0.3	0.4	0.5	0.7	0.4	0.3	0.3	0.3	0.3	0.3	0.3
1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	0.9	0.8	0.8	0.8	0.6	0.5	0.6	0.6	0.5	0.7	0.9	0.9	0.8	0.8	0.8	0.8	0.8	0.8	0.4	0.3	0.3	0.3	0.3	0.3	0.4
1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	0.9	0.9	0.9	0.9	0.5	0.3	0.4	0.7	0.8	0.9	0.9	0.9	0.9	0.9	0.8	0.8	0.8	0.8	0.5	0.4	0.3	0.3	0.3	0.4	0.4
1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	0.9	0.9	0.9	0.9	0.5	0.3	0.3	0.8	0.9	0.9	0.9	0.9	0.9	0.9	0.9	0.8	0.8	0.8	0.7	0.4	0.6	0.3	0.3	0.3	0.3
1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	0.9	0.9	0.9	0.9	0.5	0.2	0.3	0.9	0.9	0.9	0.9	0.9	0.9	0.9	0.8	0.7	0.8	0.7	0.4	0.5	0.3	0.3	0.3	0.3	0.3
1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	0.9	0.9	1.0	0.9	0.4	0.1	0.3	0.9	0.9	0.9	0.9	0.9	0.9	0.9	0.8	0.7	0.6	0.5	0.3	0.3	0.3	0.3	0.3	0.3	0.3
0.9	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	0.9	0.5	0.1	0.3	0.8	0.9	0.9	0.9	0.9	0.9	0.9	0.8	0.8	0.7	0.6	0.5	0.5	0.3	0.3	0.3	0.3	0.3
0.9	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	0.9	0.6	0.4	0.5	0.9	0.8	0.9	0.9	0.9	0.9	0.9	0.9	0.8	0.8	0.8	0.7	0.8	0.5	0.3	0.3	0.3	0.3
0.9	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	0.9	0.9	0.9	0.9	0.9	0.9	0.9	0.9	0.9	0.9	0.9	0.9	0.9	0.9	0.9	0.9	0.8	0.9	0.6	0.2	0.3	0.3	0.3
1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	0.9	0.9	0.9	0.9	0.9	0.9	0.9	0.9	0.9	0.9	0.9	0.9	0.9	0.9	0.9	0.9	0.9	0.8	0.8	0.7	0.3	0.2	0.2	0.2

An image is just a grid, or **Matrix** of numbers between 0 and 1.

The number gives the **intensity**, or darkness, of a **Pixel**.

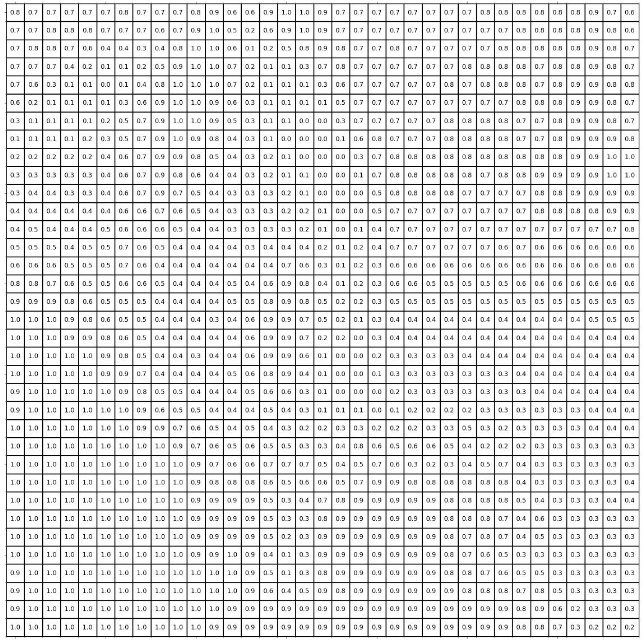
Why is Computer Vision so Hard?



0.8	0.7	0.7	0.7	0.7	0.7	0.8	0.7	0.7	0.7	0.8	0.9	0.6	0.6	0.9	1.0	1.0	0.9	0.7	0.7	0.7	0.7	0.7	0.7	0.7	0.7	0.8	0.8	0.8	0.8	0.8	0.8	0.9	0.7	0.6
0.7	0.7	0.8	0.8	0.8	0.7	0.7	0.7	0.6	0.7	0.9	1.0	0.5	0.2	0.6	0.9	1.0	0.9	0.7	0.7	0.7	0.7	0.7	0.7	0.7	0.7	0.7	0.7	0.8	0.8	0.8	0.8	0.9	0.8	0.6
0.7	0.8	0.8	0.7	0.6	0.4	0.4	0.3	0.4	0.8	1.0	1.0	0.6	0.1	0.2	0.5	0.8	0.9	0.8	0.7	0.7	0.8	0.7	0.7	0.7	0.7	0.7	0.8	0.8	0.8	0.8	0.8	0.9	0.8	0.7
0.7	0.7	0.7	0.4	0.2	0.1	0.1	0.2	0.5	0.9	1.0	1.0	0.7	0.2	0.1	0.1	0.3	0.7	0.8	0.7	0.7	0.7	0.7	0.7	0.7	0.8	0.8	0.8	0.8	0.7	0.8	0.8	0.9	0.8	0.7
0.7	0.6	0.3	0.1	0.1	0.0	0.1	0.4	0.8	1.0	1.0	1.0	0.7	0.2	0.1	0.1	0.1	0.3	0.6	0.7	0.7	0.7	0.7	0.7	0.8	0.7	0.8	0.8	0.8	0.7	0.8	0.9	0.9	0.8	0.8
0.6	0.2	0.1	0.1	0.1	0.1	0.3	0.6	0.9	1.0	1.0	0.9	0.6	0.3	0.1	0.1	0.1	0.1	0.5	0.7	0.7	0.7	0.7	0.7	0.7	0.7	0.7	0.7	0.8	0.8	0.8	0.9	0.9	0.8	0.7
0.3	0.1	0.1	0.1	0.1	0.2	0.5	0.7	0.9	1.0	1.0	0.9	0.5	0.3	0.1	0.1	0.0	0.0	0.3	0.7	0.7	0.7	0.7	0.7	0.8	0.8	0.8	0.8	0.7	0.7	0.8	0.9	0.9	0.8	0.7
0.1	0.1	0.1	0.1	0.2	0.3	0.5	0.7	0.9	1.0	0.9	0.8	0.4	0.3	0.1	0.0	0.0	0.0	0.1	0.6	0.8	0.7	0.7	0.7	0.8	0.8	0.8	0.8	0.7	0.7	0.8	0.9	0.9	0.9	0.8
0.2	0.2	0.2	0.2	0.2	0.4	0.6	0.7	0.9	0.9	0.8	0.5	0.4	0.3	0.2	0.1	0.0	0.0	0.0	0.3	0.7	0.8	0.8	0.8	0.8	0.8	0.8	0.8	0.8	0.8	0.8	0.9	0.9	1.0	1.0
0.3	0.3	0.3	0.3	0.3	0.4	0.6	0.7	0.9	0.8	0.6	0.4	0.4	0.3	0.2	0.1	0.1	0.0	0.0	0.1	0.7	0.8	0.8	0.8	0.8	0.8	0.7	0.8	0.8	0.9	0.9	0.9	0.9	1.0	1.0
0.3	0.4	0.4	0.3	0.3	0.4	0.6	0.7	0.9	0.7	0.5	0.4	0.3	0.3	0.3	0.2	0.1	0.0	0.0	0.0	0.5	0.8	0.8	0.8	0.8	0.7	0.7	0.7	0.7	0.8	0.8	0.9	0.9	0.9	0.9
0.4	0.4	0.4	0.4	0.4	0.4	0.6	0.6	0.7	0.6	0.5	0.4	0.3	0.3	0.3	0.2	0.2	0.1	0.0	0.0	0.5	0.7	0.7	0.7	0.7	0.7	0.7	0.7	0.7	0.8	0.8	0.8	0.8	0.9	0.9
0.4	0.5	0.4	0.4	0.4	0.5	0.6	0.6	0.6	0.5	0.4	0.4	0.3	0.3	0.3	0.3	0.2	0.1	0.0	0.1	0.4	0.7	0.7	0.7	0.7	0.7	0.7	0.7	0.7	0.7	0.7	0.7	0.7	0.7	0.8
0.5	0.5	0.5	0.4	0.5	0.5	0.7	0.6	0.5	0.4	0.4	0.4	0.4	0.3	0.4	0.4	0.4	0.2	0.1	0.2	0.4	0.7	0.7	0.7	0.7	0.7	0.7	0.6	0.7	0.6	0.6	0.6	0.6	0.6	0.6
0.6	0.6	0.6	0.5	0.5	0.5	0.7	0.6	0.4	0.4	0.4	0.4	0.4	0.4	0.4	0.7	0.6	0.3	0.1	0.2	0.3	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6
0.8	0.8	0.7	0.6	0.5	0.5	0.6	0.6	0.5	0.4	0.4	0.4	0.5	0.4	0.6	0.9	0.8	0.4	0.1	0.2	0.3	0.6	0.6	0.5	0.5	0.5	0.5	0.5	0.6	0.6	0.6	0.6	0.6	0.6	0.6
0.9	0.9	0.9	0.8	0.6	0.5	0.5	0.5	0.4	0.4	0.4	0.4	0.5	0.5	0.8	0.9	0.8	0.5	0.2	0.2	0.3	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5
1.0	1.0	1.0	0.9	0.8	0.6	0.5	0.5	0.4	0.4	0.4	0.3	0.4	0.6	0.9	0.9	0.7	0.5	0.2	0.1	0.3	0.4	0.4	0.4	0.4	0.4	0.4	0.4	0.4	0.4	0.4	0.4	0.5	0.5	0.5
1.0	1.0	1.0	0.9	0.9	0.8	0.6	0.5	0.4	0.4	0.4	0.4	0.4	0.6	0.9	0.9	0.7	0.2	0.2	0.0	0.3	0.4	0.4	0.4	0.4	0.4	0.4	0.4	0.4	0.4	0.4	0.4	0.4	0.4	0.4
1.0	1.0	1.0	1.0	1.0	0.9	0.8	0.5	0.4	0.4	0.3	0.4	0.4	0.6	0.9	0.9	0.6	0.1	0.0	0.0	0.2	0.3	0.3	0.3	0.3	0.4	0.4	0.4	0.4	0.4	0.4	0.4	0.4	0.4	0.4
1.0	1.0	1.0	1.0	1.0	0.9	0.9	0.7	0.4	0.4	0.4	0.4	0.5	0.6	0.8	0.9	0.4	0.1	0.0	0.0	0.1	0.3	0.3	0.3	0.3	0.3	0.3	0.3	0.4	0.4	0.4	0.4	0.4	0.4	0.4
0.9	1.0	1.0	1.0	1.0	1.0	0.9	0.8	0.5	0.5	0.4	0.4	0.4	0.5	0.6	0.6	0.3	0.1	0.0	0.0	0.0	0.2	0.3	0.3	0.3	0.3	0.3	0.3	0.3	0.4	0.4	0.4	0.4	0.4	0.4
0.9	1.0	1.0	1.0	1.0	1.0	1.0	0.9	0.6	0.5	0.5	0.4	0.4	0.4	0.5	0.4	0.3	0.1	0.1	0.1	0.0	0.1	0.2	0.2	0.2	0.2	0.3	0.3	0.3	0.3	0.3	0.3	0.4	0.4	0.4
1.0	1.0	1.0	1.0	1.0	1.0	1.0	0.9	0.9	0.7	0.6	0.5	0.4	0.5	0.4	0.3	0.2	0.2	0.3	0.3	0.2	0.2	0.2	0.3	0.3	0.5	0.3	0.2	0.3	0.3	0.3	0.3	0.4	0.4	0.4
1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	0.9	0.7	0.6	0.5	0.6	0.5	0.5	0.3	0.3	0.4	0.8	0.6	0.5	0.6	0.6	0.5	0.4	0.2	0.2	0.2	0.3	0.3	0.3	0.3	0.3	0.3
1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	0.9	0.7	0.6	0.6	0.7	0.7	0.7	0.5	0.4	0.5	0.7	0.6	0.3	0.2	0.3	0.4	0.5	0.7	0.4	0.3	0.3	0.3	0.3	0.3	0.3
1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	0.9	0.8	0.8	0.8	0.6	0.5	0.6	0.6	0.5	0.7	0.9	0.9	0.8	0.8	0.8	0.8	0.8	0.8	0.4	0.3	0.3	0.3	0.3	0.3	0.4
1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	0.9	0.9	0.9	0.9	0.5	0.3	0.4	0.7	0.8	0.9	0.9	0.9	0.9	0.9	0.8	0.8	0.8	0.8	0.5	0.4	0.3	0.3	0.3	0.4	0.4
1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	0.9	0.9	0.9	0.9	0.5	0.3	0.3	0.8	0.9	0.9	0.9	0.9	0.9	0.9	0.8	0.8	0.8	0.7	0.4	0.6	0.3	0.3	0.3	0.3	0.3
1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	0.9	0.9	0.9	0.9	0.5	0.2	0.3	0.9	0.9	0.9	0.9	0.9	0.9	0.9	0.8	0.7	0.8	0.7	0.4	0.5	0.3	0.3	0.3	0.3	0.3
1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	0.9	0.9	1.0	0.9	0.4	0.1	0.3	0.9	0.9	0.9	0.9	0.9	0.9	0.9	0.8	0.7	0.6	0.5	0.3	0.3	0.3	0.3	0.3	0.3	0.3
0.9	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	0.9	0.5	0.1	0.3	0.8	0.9	0.9	0.9	0.9	0.9	0.9	0.8	0.8	0.7	0.6	0.5	0.5	0.3	0.3	0.3	0.3	0.3
0.9	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	0.9	0.6	0.4	0.5	0.9	0.8	0.9	0.9	0.9	0.9	0.9	0.9	0.8	0.8	0.8	0.7	0.8	0.5	0.3	0.3	0.3	0.3
0.9	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	0.9	0.9	0.9	0.9	0.9	0.9	0.9	0.9	0.9	0.9	0.9	0.9	0.9	0.9	0.9	0.9	0.8	0.9	0.6	0.2	0.3	0.3	0.3
1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	0.9	0.9	0.9	0.9	0.9	0.9	0.9	0.9	0.9	0.9	0.9	0.9	0.9	0.9	0.9	0.9	0.9	0.8	0.8	0.7	0.3	0.2	0.2	0.2

Can you see the Panda’s head in this Matrix?

Why is Computer Vision so Hard?



0.8	0.7	0.7	0.7	0.7	0.7	0.8	0.7	0.7	0.7	0.8	0.9	0.6	0.6	0.9	1.0	1.0	0.9	0.7	0.7	0.7	0.7	0.7	0.7	0.7	0.7	0.8	0.8	0.8	0.8	0.8	0.8	0.9	0.7	0.6
0.7	0.7	0.8	0.8	0.8	0.7	0.7	0.7	0.6	0.7	0.9	1.0	0.5	0.2	0.6	0.9	1.0	0.9	0.7	0.7	0.7	0.7	0.7	0.7	0.7	0.7	0.7	0.7	0.8	0.8	0.8	0.8	0.9	0.8	0.6
0.7	0.8	0.8	0.7	0.6	0.4	0.4	0.3	0.4	0.8	1.0	1.0	0.6	0.1	0.2	0.5	0.8	0.9	0.8	0.7	0.7	0.8	0.7	0.7	0.7	0.7	0.7	0.8	0.8	0.8	0.8	0.8	0.9	0.8	0.7
0.7	0.7	0.7	0.4	0.2	0.1	0.1	0.2	0.5	0.9	1.0	1.0	0.7	0.2	0.1	0.1	0.3	0.7	0.8	0.7	0.7	0.7	0.7	0.7	0.7	0.8	0.8	0.8	0.8	0.7	0.8	0.8	0.9	0.8	0.7
0.7	0.6	0.3	0.1	0.1	0.0	0.1	0.4	0.8	1.0	1.0	1.0	0.7	0.2	0.1	0.1	0.1	0.3	0.6	0.7	0.7	0.7	0.7	0.7	0.8	0.7	0.8	0.8	0.8	0.7	0.8	0.9	0.9	0.8	0.8
0.6	0.2	0.1	0.1	0.1	0.1	0.3	0.6	0.9	1.0	1.0	0.9	0.6	0.3	0.1	0.1	0.1	0.1	0.5	0.7	0.7	0.7	0.7	0.7	0.7	0.7	0.7	0.7	0.8	0.8	0.8	0.9	0.9	0.8	0.7
0.3	0.1	0.1	0.1	0.1	0.2	0.5	0.7	0.9	1.0	1.0	0.9	0.5	0.3	0.1	0.1	0.0	0.0	0.3	0.7	0.7	0.7	0.7	0.7	0.8	0.8	0.8	0.8	0.7	0.7	0.8	0.9	0.9	0.8	0.7
0.1	0.1	0.1	0.1	0.2	0.3	0.5	0.7	0.9	1.0	0.9	0.8	0.4	0.3	0.1	0.0	0.0	0.0	0.1	0.6	0.8	0.7	0.7	0.7	0.8	0.8	0.8	0.8	0.7	0.7	0.8	0.9	0.9	0.9	0.8
0.2	0.2	0.2	0.2	0.2	0.4	0.6	0.7	0.9	0.9	0.8	0.5	0.4	0.3	0.2	0.1	0.0	0.0	0.0	0.3	0.7	0.8	0.8	0.8	0.8	0.8	0.8	0.8	0.8	0.8	0.8	0.9	0.9	1.0	1.0
0.3	0.3	0.3	0.3	0.3	0.4	0.6	0.7	0.9	0.8	0.6	0.4	0.4	0.3	0.2	0.1	0.1	0.0	0.0	0.1	0.7	0.8	0.8	0.8	0.8	0.8	0.7	0.8	0.8	0.9	0.9	0.9	0.9	1.0	1.0
0.3	0.4	0.4	0.3	0.3	0.4	0.6	0.7	0.9	0.7	0.5	0.4	0.3	0.3	0.3	0.2	0.1	0.0	0.0	0.0	0.5	0.8	0.8	0.8	0.8	0.7	0.7	0.7	0.7	0.8	0.8	0.9	0.9	0.9	0.9
0.4	0.4	0.4	0.4	0.4	0.4	0.6	0.6	0.7	0.6	0.5	0.4	0.3	0.3	0.3	0.2	0.2	0.1	0.0	0.0	0.5	0.7	0.7	0.7	0.7	0.7	0.7	0.7	0.7	0.8	0.8	0.8	0.8	0.9	0.9
0.4	0.5	0.4	0.4	0.4	0.5	0.6	0.6	0.6	0.5	0.4	0.4	0.3	0.3	0.3	0.3	0.2	0.1	0.0	0.1	0.4	0.7	0.7	0.7	0.7	0.7	0.7	0.7	0.7	0.7	0.7	0.7	0.7	0.7	0.8
0.5	0.5	0.5	0.4	0.5	0.5	0.7	0.6	0.5	0.4	0.4	0.4	0.4	0.3	0.4	0.4	0.4	0.2	0.1	0.2	0.4	0.7	0.7	0.7	0.7	0.7	0.7	0.6	0.7	0.6	0.6	0.6	0.6	0.6	0.6
0.6	0.6	0.6	0.5	0.5	0.5	0.7	0.6	0.4	0.4	0.4	0.4	0.4	0.4	0.4	0.7	0.6	0.3	0.1	0.2	0.3	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6
0.8	0.8	0.7	0.6	0.5	0.5	0.6	0.6	0.5	0.4	0.4	0.4	0.5	0.4	0.6	0.9	0.8	0.4	0.1	0.2	0.3	0.6	0.6	0.5	0.5	0.5	0.5	0.5	0.6	0.6	0.6	0.6	0.6	0.6	0.6
0.9	0.9	0.9	0.8	0.6	0.5	0.5	0.5	0.4	0.4	0.4	0.4	0.5	0.5	0.8	0.9	0.8	0.5	0.2	0.2	0.3	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5
1.0	1.0	1.0	0.9	0.8	0.6	0.5	0.5	0.4	0.4	0.4	0.3	0.4	0.6	0.9	0.9	0.7	0.5	0.2	0.1	0.3	0.4	0.4	0.4	0.4	0.4	0.4	0.4	0.4	0.4	0.4	0.4	0.5	0.5	0.5
1.0	1.0	1.0	0.9	0.9	0.8	0.6	0.5	0.4	0.4	0.4	0.4	0.4	0.6	0.9	0.9	0.7	0.2	0.2	0.0	0.3	0.4	0.4	0.4	0.4	0.4	0.4	0.4	0.4	0.4	0.4	0.4	0.4	0.4	0.4
1.0	1.0	1.0	1.0	1.0	0.9	0.8	0.5	0.4	0.4	0.3	0.4	0.4	0.6	0.9	0.9	0.6	0.1	0.0	0.0	0.2	0.3	0.3	0.3	0.3	0.4	0.4	0.4	0.4	0.4	0.4	0.4	0.4	0.4	0.4
1.0	1.0	1.0	1.0	1.0	0.9	0.9	0.7	0.4	0.4	0.4	0.4	0.5	0.6	0.8	0.9	0.4	0.1	0.0	0.0	0.1	0.3	0.3	0.3	0.3	0.3	0.3	0.3	0.4	0.4	0.4	0.4	0.4	0.4	0.4
0.9	1.0	1.0	1.0	1.0	1.0	0.9	0.8	0.5	0.5	0.4	0.4	0.4	0.5	0.6	0.6	0.3	0.1	0.0	0.0	0.0	0.2	0.3	0.3	0.3	0.3	0.3	0.3	0.3	0.4	0.4	0.4	0.4	0.4	0.4
0.9	1.0	1.0	1.0	1.0	1.0	1.0	0.9	0.6	0.5	0.5	0.4	0.4	0.4	0.5	0.4	0.3	0.1	0.1	0.1	0.0	0.1	0.2	0.2	0.2	0.2	0.3	0.3	0.3	0.3	0.3	0.3	0.4	0.4	0.4
1.0	1.0	1.0	1.0	1.0	1.0	1.0	0.9	0.9	0.7	0.6	0.5	0.4	0.5	0.4	0.3	0.2	0.2	0.3	0.3	0.2	0.2	0.2	0.3	0.3	0.5	0.3	0.2	0.3	0.3	0.3	0.3	0.4	0.4	0.4
1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	0.9	0.7	0.6	0.5	0.6	0.5	0.5	0.3	0.3	0.4	0.8	0.6	0.5	0.6	0.6	0.5	0.4	0.2	0.2	0.2	0.3	0.3	0.3	0.3	0.3	0.3
1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	0.9	0.7	0.6	0.6	0.7	0.7	0.7	0.5	0.4	0.5	0.7	0.6	0.3	0.2	0.3	0.4	0.5	0.7	0.4	0.3	0.3	0.3	0.3	0.3	0.3
1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	0.9	0.8	0.8	0.8	0.6	0.5	0.6	0.6	0.5	0.7	0.9	0.9	0.8	0.8	0.8	0.8	0.8	0.8	0.4	0.3	0.3	0.3	0.3	0.3	0.4
1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	0.9	0.9	0.9	0.9	0.5	0.3	0.4	0.7	0.8	0.9	0.9	0.9	0.9	0.9	0.8	0.8	0.8	0.8	0.5	0.4	0.3	0.3	0.3	0.4	0.4
1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	0.9	0.9	0.9	0.9	0.5	0.3	0.3	0.8	0.9	0.9	0.9	0.9	0.9	0.9	0.8	0.8	0.8	0.7	0.4	0.6	0.3	0.3	0.3	0.3	0.3
1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	0.9	0.9	0.9	0.9	0.5	0.2	0.3	0.9	0.9	0.9	0.9	0.9	0.9	0.9	0.8	0.7	0.8	0.7	0.4	0.5	0.3	0.3	0.3	0.3	0.3
1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	0.9	0.9	1.0	0.9	0.4	0.1	0.3	0.9	0.9	0.9	0.9	0.9	0.9	0.9	0.8	0.7	0.6	0.5	0.3	0.3	0.3	0.3	0.3	0.3	0.3
0.9	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	0.9	0.5	0.1	0.3	0.8	0.9	0.9	0.9	0.9	0.9	0.9	0.8	0.8	0.7	0.6	0.5	0.5	0.3	0.3	0.3	0.3	0.3
0.9	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	0.9	0.6	0.4	0.5	0.9	0.8	0.9	0.9	0.9	0.9	0.9	0.9	0.8	0.8	0.8	0.7	0.8	0.5	0.3	0.3	0.3	0.3
0.9	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	0.9	0.9	0.9	0.9	0.9	0.9	0.9	0.9	0.9	0.9	0.9	0.9	0.9	0.9	0.9	0.9	0.8	0.9	0.6	0.2	0.3	0.3	0.3
1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	0.9	0.9	0.9	0.9	0.9	0.9	0.9	0.9	0.9	0.9	0.9	0.9	0.9	0.9	0.9	0.9	0.9	0.8	0.8	0.7	0.3	0.2	0.2	0.2

Computer Vision is all about making sense of data like **this**.

Matrices and Arrays

Matrix is a generic term for a grid of numbers.

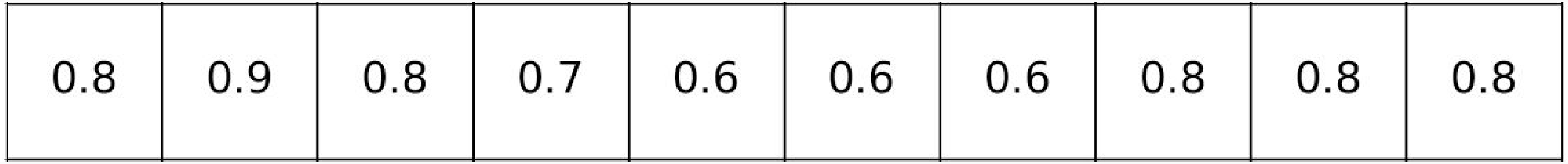
Array is a type of object in Python, which can represent Matrices.

Image is what we call it when we draw out arrays in a way that's easy to see.

We will use these terms mostly interchangeably.

Matrices and Arrays

Arrays can have different numbers of **axes**, which describe how many rows and columns there are. For example, an array with **one axis** looks like this:

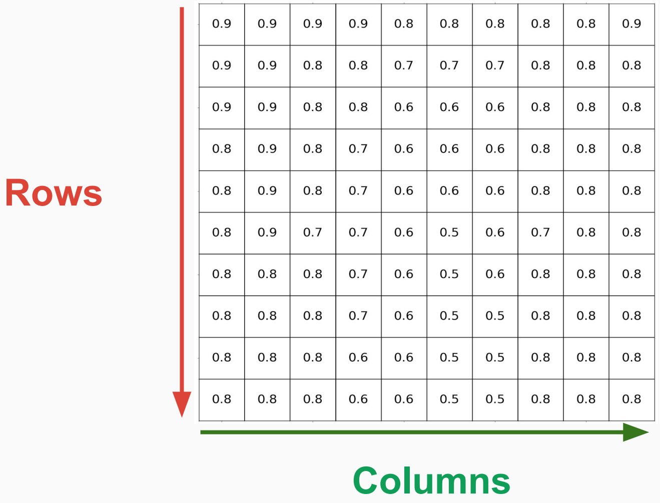


0.8	0.9	0.8	0.7	0.6	0.6	0.6	0.8	0.8	0.8

We call such an array a **vector**, or **1-dimensional array**.

Matrices and Arrays

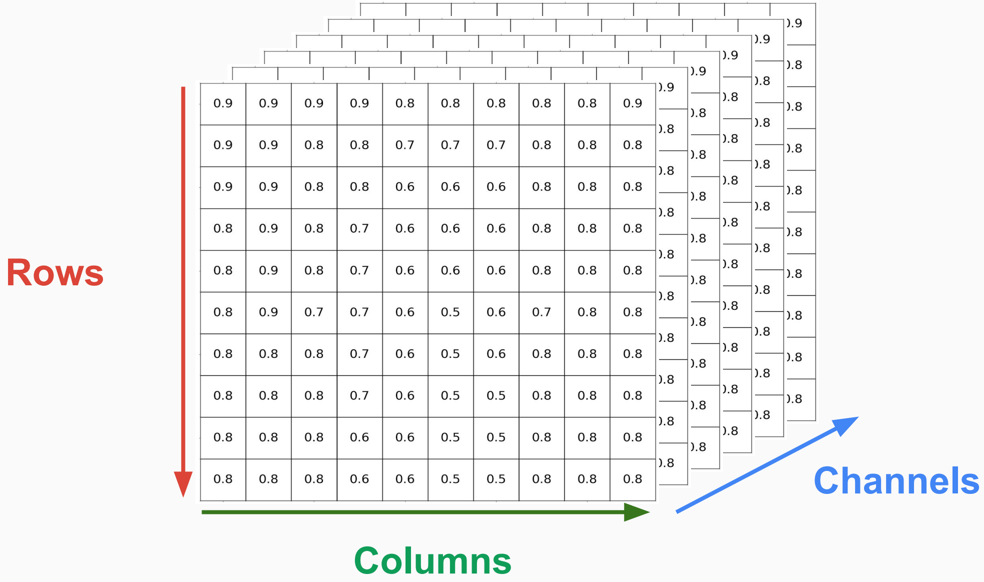
An array with **two axes** looks like this:



0.9	0.9	0.9	0.9	0.8	0.8	0.8	0.8	0.8	0.9
0.9	0.9	0.8	0.8	0.7	0.7	0.7	0.8	0.8	0.8
0.9	0.9	0.8	0.8	0.6	0.6	0.6	0.8	0.8	0.8
0.8	0.9	0.8	0.7	0.6	0.6	0.6	0.8	0.8	0.8
0.8	0.9	0.8	0.7	0.6	0.6	0.6	0.8	0.8	0.8
0.8	0.9	0.7	0.7	0.6	0.5	0.6	0.7	0.8	0.8
0.8	0.8	0.8	0.7	0.6	0.5	0.6	0.8	0.8	0.8
0.8	0.8	0.8	0.7	0.6	0.5	0.5	0.8	0.8	0.8
0.8	0.8	0.8	0.6	0.6	0.5	0.5	0.8	0.8	0.8
0.8	0.8	0.8	0.6	0.6	0.5	0.5	0.8	0.8	0.8

Matrices and Arrays

In fact, arrays can have even more than two axes! Here is an array **with 3 axes**:



0.9	0.9	0.9	0.9	0.8	0.8	0.8	0.8	0.8	0.9
0.9	0.9	0.8	0.8	0.7	0.7	0.7	0.8	0.8	0.8
0.9	0.9	0.8	0.8	0.6	0.6	0.6	0.8	0.8	0.8
0.8	0.9	0.8	0.7	0.6	0.6	0.6	0.8	0.8	0.8
0.8	0.9	0.8	0.7	0.6	0.6	0.6	0.8	0.8	0.8
0.8	0.9	0.7	0.7	0.6	0.5	0.6	0.7	0.8	0.8
0.8	0.8	0.8	0.7	0.6	0.5	0.6	0.8	0.8	0.8
0.8	0.8	0.8	0.7	0.6	0.5	0.5	0.8	0.8	0.8
0.8	0.8	0.8	0.6	0.6	0.5	0.5	0.8	0.8	0.8
0.8	0.8	0.8	0.6	0.6	0.5	0.5	0.8	0.8	0.8

Matrices and Arrays

We’ve already seen how a black and white image can be shown with a 2D array.

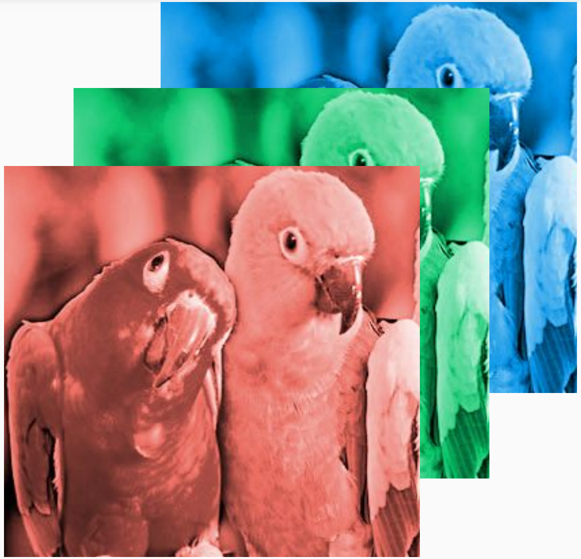
Can you think of something we might do with 3D arrays?

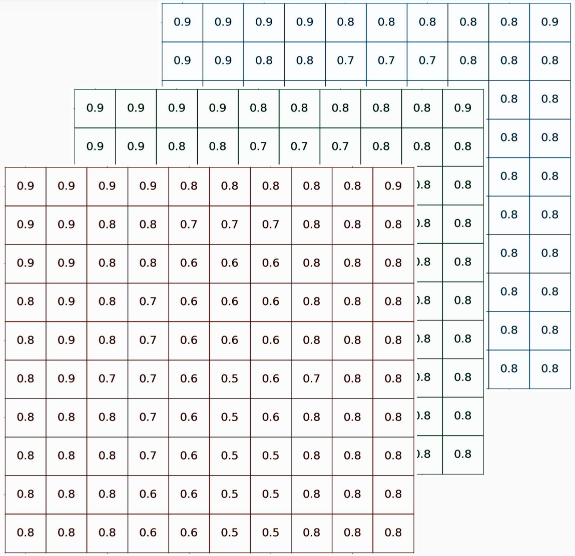
Representing Color

Pixels now vary in **color** and **saturation** as well as **intensity**.

Its no longer possible to express every pixel with just **ONE** number!

Representing Color





0.9	0.9	0.9	0.9	0.8	0.8	0.8	0.8	0.8	0.9
0.9	0.9	0.8	0.8	0.7	0.7	0.7	0.8	0.8	0.8
								0.8	0.8
								0.8	0.8
								0.8	0.8
								0.8	0.8
								0.8	0.8
								0.8	0.8
								0.8	0.8
								0.8	0.8

0.9	0.9	0.9	0.9	0.8	0.8	0.8	0.8	0.8	0.9
0.9	0.9	0.8	0.8	0.7	0.7	0.7	0.8	0.8	0.8
								).8	0.8
								).8	0.8
								).8	0.8
								).8	0.8
								).8	0.8
								).8	0.8
								).8	0.8
								).8	0.8

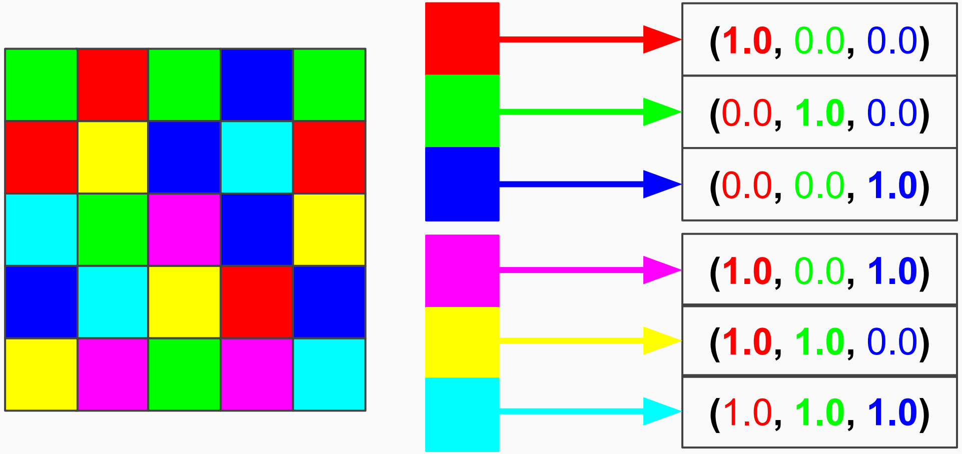
0.9	0.9	0.9	0.9	0.8	0.8	0.8	0.8	0.8	0.9
0.9	0.9	0.8	0.8	0.7	0.7	0.7	0.8	0.8	0.8
0.9	0.9	0.8	0.8	0.6	0.6	0.6	0.8	0.8	0.8
0.8	0.9	0.8	0.7	0.6	0.6	0.6	0.8	0.8	0.8
0.8	0.9	0.8	0.7	0.6	0.6	0.6	0.8	0.8	0.8
0.8	0.9	0.7	0.7	0.6	0.5	0.6	0.7	0.8	0.8
0.8	0.8	0.8	0.7	0.6	0.5	0.6	0.8	0.8	0.8
0.8	0.8	0.8	0.7	0.6	0.5	0.5	0.8	0.8	0.8
0.8	0.8	0.8	0.6	0.6	0.5	0.5	0.8	0.8	0.8
0.8	0.8	0.8	0.6	0.6	0.5	0.5	0.8	0.8	0.8

We can use 3D Arrays to store information about colors along the **third axis.**

RGB Color Model

Every pixel in an image is represented with 3 Numbers, telling us how

Red, **Green**, or **Blue** it is.



(**1.0**, 0.0, 0.0)

(0.0, **1.0**, 0.0)

(0.0, 0.0, **1.0**)

(**1.0**, 0.0, **1.0**)

(**1.0**, **1.0**, 0.0)

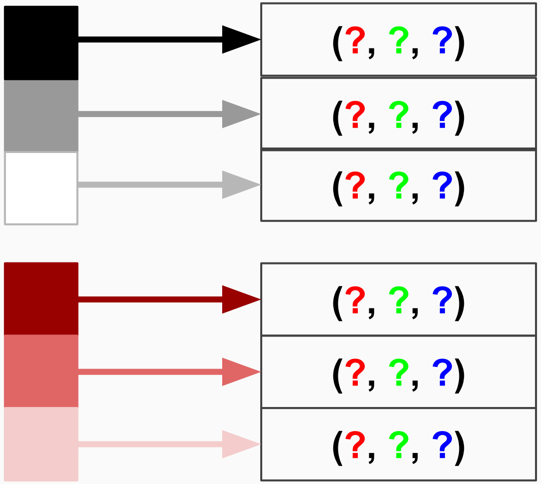
(1.0, **1.0**, **1.0**)

Colors can be purely red, green or blue.

Or merged to make different colors.

RGB Color Model

How can we express different shades of colors in RGB?



(?, ?, ?)

(?, ?, ?)

(?, ?, ?)

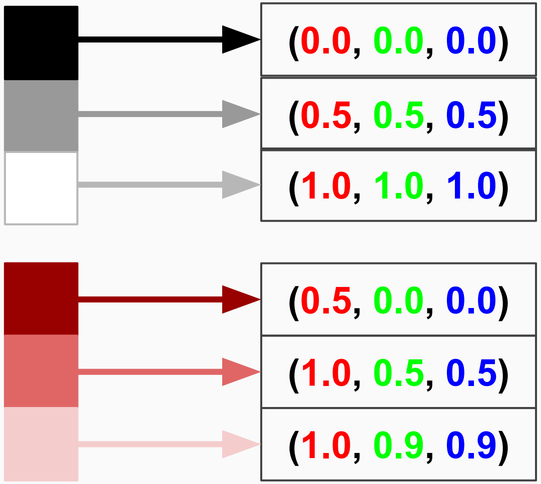
(?, ?, ?)

(?, ?, ?)

(?, ?, ?)

RGB Color Model

How can we express different shades of colors in RGB?



(0.0, 0.0, 0.0)

(0.5, 0.5, 0.5)

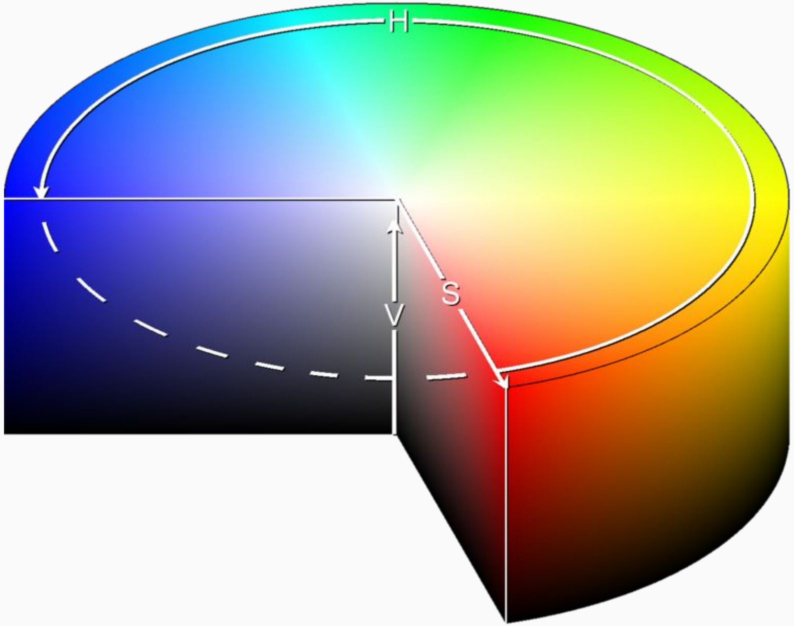
(1.0, 1.0, 1.0)

(0.5, 0.0, 0.0)

(1.0, 0.5, 0.5)

(1.0, 0.9, 0.9)

HSV Color Model



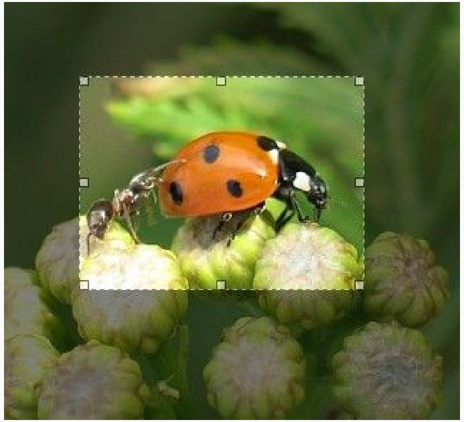
Hue, Saturation, Value is another popular format for colors.

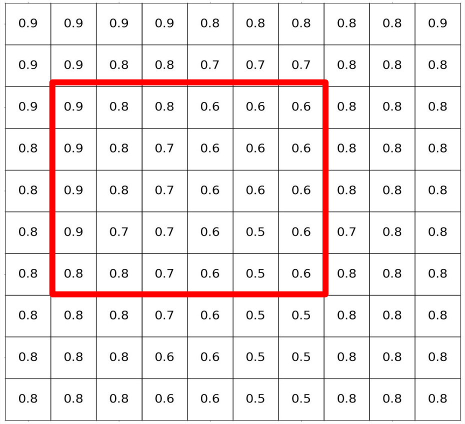
This is nice because **all** the colors (hues) can be expressed on one axis.

🟥	(0.0, 1.0, 1.0)
🟩	(0.3, 1.0, 1.0)
🟦	(0.6, 1.0, 1.0)

We use the other two axes to tell how dark or saturated a pixel is.

Applications

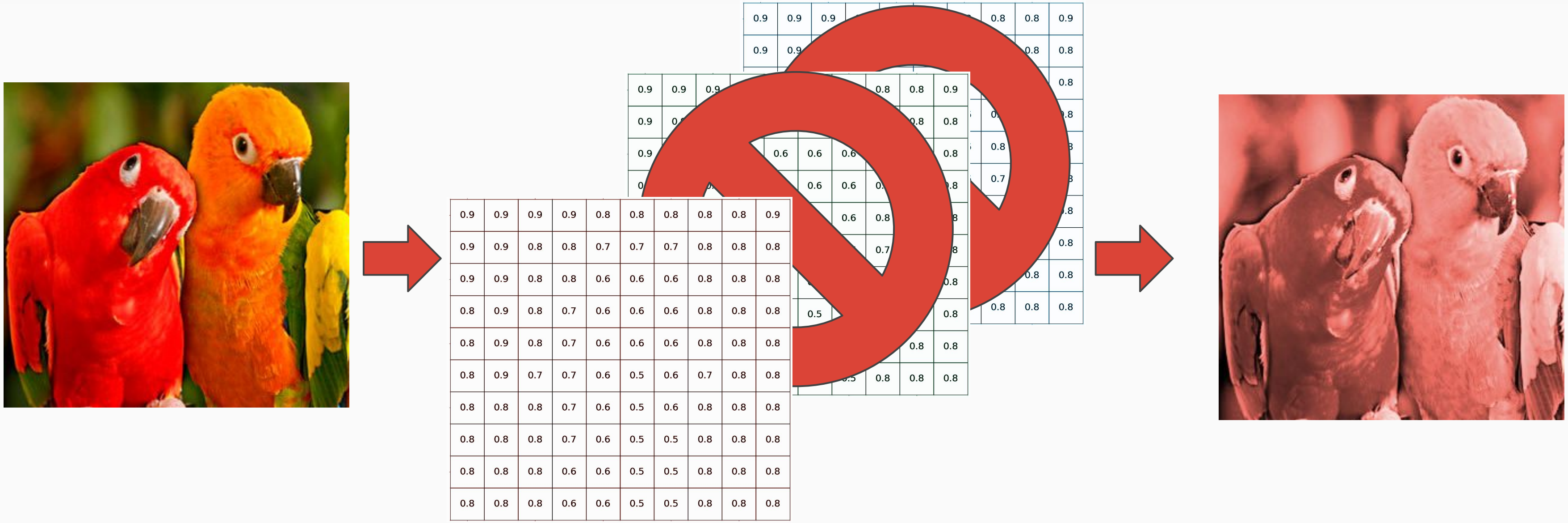




0.9	0.9	0.9	0.9	0.8	0.8	0.8	0.8	0.8	0.9
0.9	0.9	0.8	0.8	0.7	0.7	0.7	0.8	0.8	0.8
0.9	0.9	0.8	0.8	0.6	0.6	0.6	0.8	0.8	0.8
0.8	0.9	0.8	0.7	0.6	0.6	0.6	0.8	0.8	0.8
0.8	0.9	0.8	0.7	0.6	0.6	0.6	0.8	0.8	0.8
0.8	0.9	0.7	0.7	0.6	0.5	0.6	0.7	0.8	0.8
0.8	0.8	0.8	0.7	0.6	0.5	0.6	0.8	0.8	0.8
0.8	0.8	0.8	0.7	0.6	0.5	0.5	0.8	0.8	0.8
0.8	0.8	0.8	0.6	0.6	0.5	0.5	0.8	0.8	0.8
0.8	0.8	0.8	0.6	0.6	0.5	0.5	0.8	0.8	0.8

Cropping an image involves **slicing** out rows and columns of the matrix you don’t want.

Applications



0.9	0.9	0.9	0.9	0.8	0.8	0.8	0.8	0.8	0.9
0.9	0.9	0.8	0.8	0.7	0.7	0.7	0.8	0.8	0.8
0.9	0.9	0.8	0.8	0.6	0.6	0.6	0.8	0.8	0.8
0.8	0.9	0.8	0.7	0.6	0.6	0.6	0.8	0.8	0.8
0.8	0.9	0.8	0.7	0.6	0.6	0.6	0.8	0.8	0.8
0.8	0.9	0.7	0.7	0.6	0.5	0.6	0.7	0.8	0.8
0.8	0.8	0.8	0.7	0.6	0.5	0.6	0.8	0.8	0.8
0.8	0.8	0.8	0.7	0.6	0.5	0.5	0.8	0.8	0.8
0.8	0.8	0.8	0.6	0.6	0.5	0.5	0.8	0.8	0.8
0.8	0.8	0.8	0.6	0.6	0.5	0.5	0.8	0.8	0.8

Tinting an image requires “zeroing-out” channels for other colors.

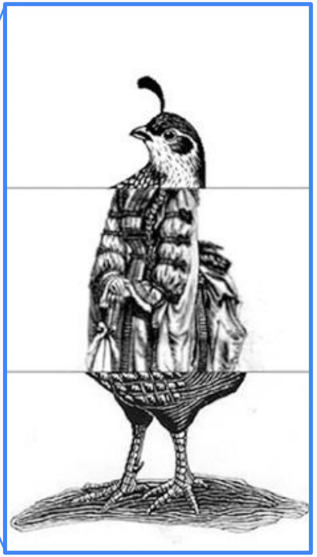
Applications

0.9	0.9	0.9	0.9	0.8	0.8	0.8	0.8
0.9	0.9	0.8	0.8	0.7	0.7	0.7	0.8
0.9	0.9	0.8	0.8	0.6	0.6	0.6	0.8
0.8	0.9	0.8	0.7	0.6	0.6	0.6	0.8
0.8	0.9	0.8	0.7	0.6	0.6	0.6	0.8

0.9	0.9	0.9	0.9	0.8	0.8	0.8	0.8
0.9	0.9	0.8	0.8	0.7	0.7	0.7	0.8
0.9	0.9	0.8	0.8	0.6	0.6	0.6	0.8
0.8	0.9	0.8	0.7	0.6	0.6	0.6	0.8
0.8	0.9	0.8	0.7	0.6	0.6	0.6	0.8

0.9	0.9	0.9	0.9	0.8	0.8	0.8	0.8
0.9	0.9	0.8	0.8	0.7	0.7	0.7	0.8
0.9	0.9	0.8	0.8	0.6	0.6	0.6	0.8
0.8	0.9	0.8	0.7	0.6	0.6	0.6	0.8
0.8	0.9	0.8	0.7	0.6	0.6	0.6	0.8

0.9	0.9	0.9	0.9	0.8	0.8	0.8	0.8
0.9	0.9	0.8	0.8	0.7	0.7	0.7	0.8
0.9	0.9	0.8	0.8	0.6	0.6	0.6	0.8
0.8	0.9	0.8	0.7	0.6	0.6	0.6	0.8
0.8	0.9	0.8	0.7	0.6	0.6	0.6	0.8
0.9	0.9	0.9	0.9	0.8	0.8	0.8	0.8
0.9	0.9	0.8	0.8	0.7	0.7	0.7	0.8
0.9	0.9	0.8	0.8	0.6	0.6	0.6	0.8
0.8	0.9	0.8	0.7	0.6	0.6	0.6	0.8
0.8	0.9	0.8	0.7	0.6	0.6	0.6	0.8
0.9	0.9	0.9	0.9	0.8	0.8	0.8	0.8
0.9	0.9	0.8	0.8	0.7	0.7	0.7	0.8
0.9	0.9	0.8	0.8	0.6	0.6	0.6	0.8
0.8	0.9	0.8	0.7	0.6	0.6	0.6	0.8
0.8	0.9	0.8	0.7	0.6	0.6	0.6	0.8



Matrices can also be stacked, or **concatenated**, to create new images.

Recap

1. Every image is represented with a matrix, or an array with two or three axes.
2. Color Images use **3D arrays** to store information about colors along their third axis.
3. The **RGB** format describes every pixel with a **red**, **green**, and **blue** value.
4. Operations like **slicing** and **concatenation** let us manipulate images.