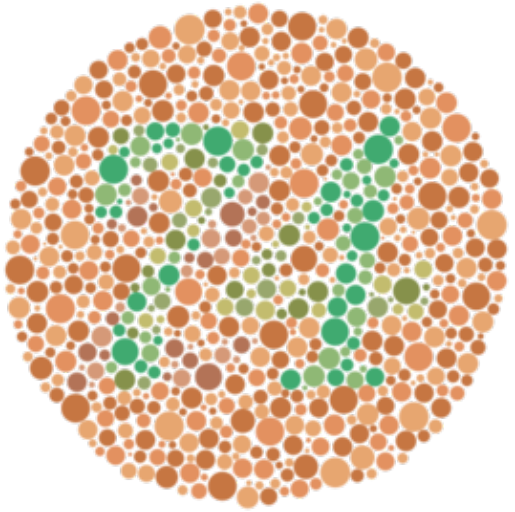
Monolens: view part of your screen in grayscale or simulated color vision deficiency

Hans Dembinski, TU Dortmund, Germany

Motivation

- Reviewer:
  "Please make your plots readable in b/w print"
  - Real case: Our current LHCb paper

- UI design: make color themes useful for people with color vision deficiency
  - Real case: iminuit Jupyter UI



Ishihara Plate 9
(public domain, from Wikimedia)

Monolens

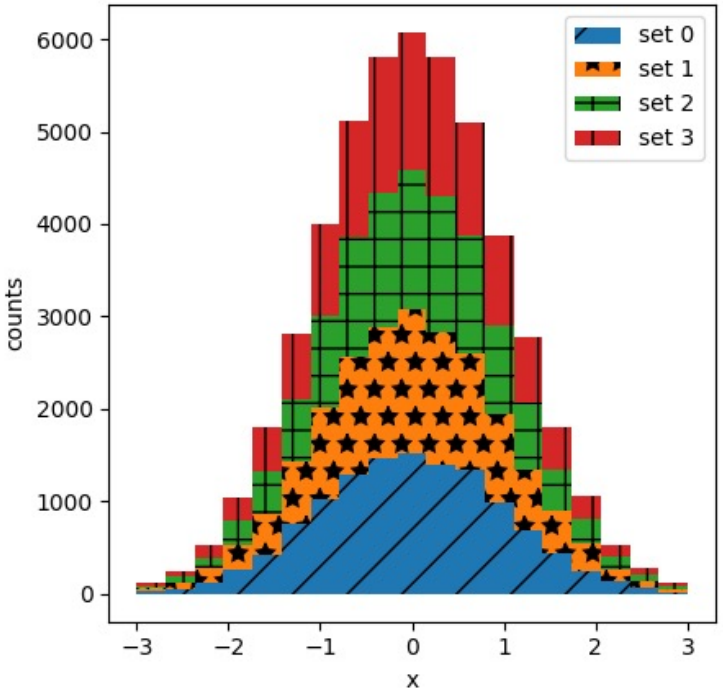
pip install monolens

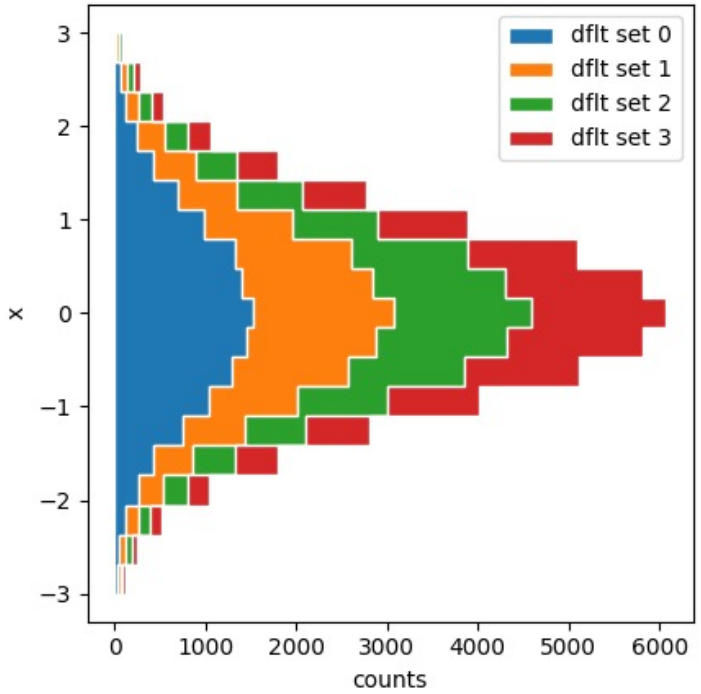
or

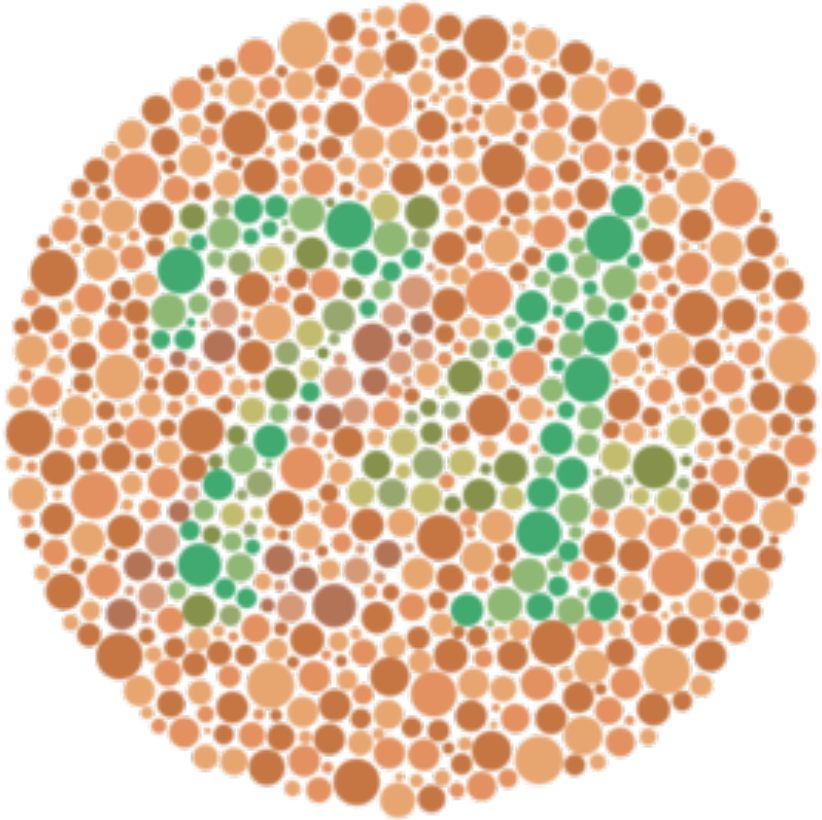
pipx run monolens

- Pure Python: Qt GUI (PySide6), Numpy, Numba
- Works on OSX, Linux… and perhaps even Windows (untested)
- Easy to use and fun to use

Live demo


set 0
set 1
set 2
set 3
counts
x


dflt set 0
dflt set 1
dflt set 2
dflt set 3
x
counts



A look under the hood

- Monolens makes screenshots to give illusion of a filter
  - Fun fact: Apps cannot read screen pixels under themselves
  - Thus only works for static content: no movies, no animations
- As you move lens, screenshot is updated
  - Full HD or more with retina displays: 8 to 20 MB at X frames per second
  - Challenge color-convert large images in real-time in Python
  - Simplified color conversion in monolens is implemented as vectorized add/multiply
  - Embarrassingly parallelizable and profits from SIMD
- PySide6 has no Numpy interface, but QImage.bits supports buffer protocol
  - Can view image memory as Numpy array and mess with it (read & write)

Numpy vs. Numba

```
a, r, g, b = util.argb


def _color_conversion_np(d, s, cb):
    d[:, r] = cb[0, 0] * s[:, r] + cb[0, 1] * s[:, g] + cb[0, 2] * s[:, b]
    d[:, g] = cb[1, 0] * s[:, r] + cb[1, 1] * s[:, g] + cb[1, 2] * s[:, b]
    d[:, b] = cb[2, 0] * s[:, r] + cb[2, 1] * s[:, g] + cb[2, 2] * s[:, b]


@nb.njit(parallel=True)
def _color_conversion_nbp(d, s, cb):
    for i in nb.prange(len(s)):
        p = s[i]
        d[i, r] = cb[0, 0] * p[r] + cb[0, 1] * p[g] + cb[0, 2] * p[b]
        d[i, g] = cb[1, 0] * p[r] + cb[1, 1] * p[g] + cb[1, 2] * p[b]
        d[i, b] = cb[2, 0] * p[r] + cb[2, 1] * p[g] + cb[2, 2] * p[b]
```

Numba version **20x faster** than Numpy code

Conclusions

- Numba rocks
- Color vision deficiency is not rare
  - Red-green: 8 % of European males, 0.5 % of females
  - About 1-2 persons in an average seminar room of 30 people
  - Easy to forget, if you are not affected, but also easy to fix if you are willing
- My other projects
  - boostorg/histogram
  - scikit-hep/boost-histogram with Henry Schreiner
  - scikit-hep/iminuit
  - scikit-hep/pyhepmc
  - resample-project/resample (will join scikit-hep)