

Sprint 4 Progress

A2_02

The idea structure

1. Create a filter to output an clear image
2. Use python to build an photonic circuit
3. Read the input, print the output

The filter

In a typical image recognition application, a **convolutional layer** is made up of several filters to detect the various features of an image. Understanding how this work is best illustrated with an analogy. Suppose you saw someone walking towards you from a distance.

```
halfring1 = sipann.HalfRing(500e-9, 220e-9, radius, 100e-9)
halfring2 = sipann.HalfRing(500e-9, 220e-9, radius, 100e-9)
terminator = siepic.Terminator()

halfring1.rename_pins("pass", "midb", "in", "midt")
halfring2.rename_pins("out", "midt", "term", "midb")

# the interface method will connect all of the pins with matching names
# between the two components together
halfring1.interface(halfring2)
halfring2["term"].connect(terminator)

# bundling the circuit as a Subcircuit allows us to interact with it
# as if it were a component
return halfring1.circuit.to_subcircuit()

# Behold, we can run a simulation on a single ring resonator.
ring1 = ring_factory(10e-6)

simulator = SweepSimulator(1500e-9, 1600e-9)
simulator.multiconnect(ring1["in"], ring1["pass"])

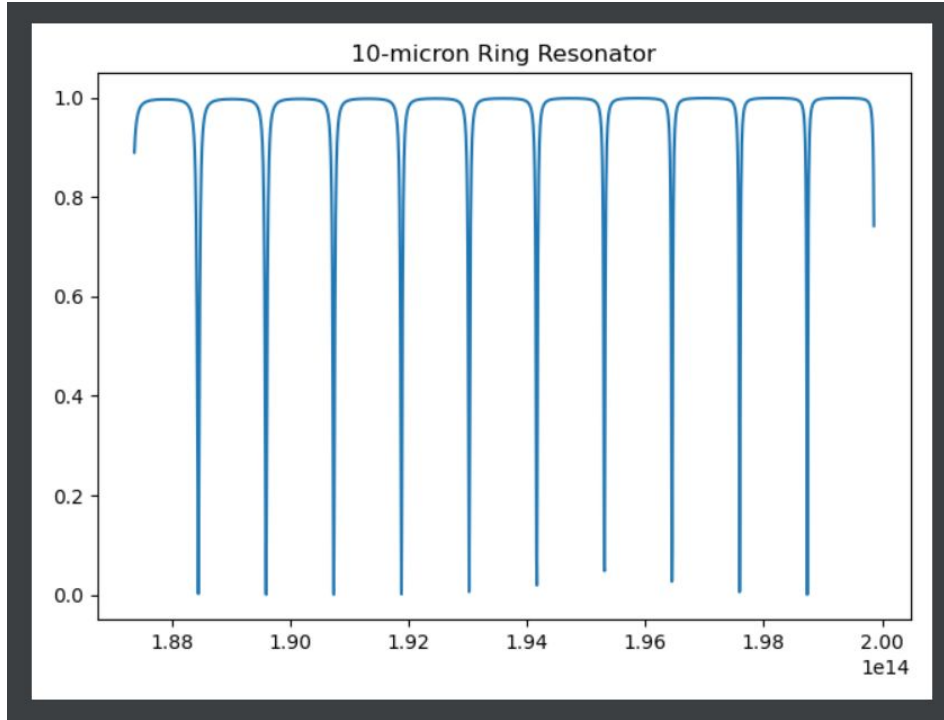
f, t = simulator.simulate(mode="freq")
plt.plot(f, t)
plt.title("10-micron Ring Resonator")
plt.tight_layout()
plt.show()

simulator.disconnect()

# Now, we'll create the circuit (using several ring resonator subcircuits)
# instantiate the basic components
wg_input = siepic.Waveguide(100e-6)
wg_out1 = siepic.Waveguide(100e-6)
wg_connect1 = siepic.Waveguide(100e-6)
wg_out2 = siepic.Waveguide(100e-6)
wg_connect2 = siepic.Waveguide(100e-6)
wg_out3 = siepic.Waveguide(100e-6)
terminator = siepic.Terminator()

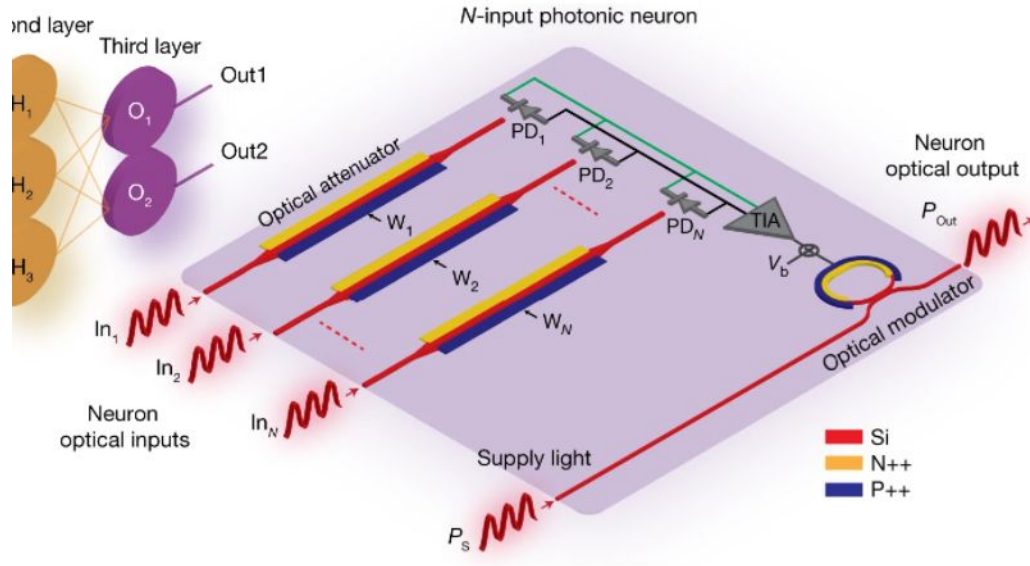
# instantiate the rings with varying radii
ring1 = ring_factory(10e-6)
ring2 = ring_factory(11e-6)
ring3 = ring_factory(12e-6)
```

The filter image



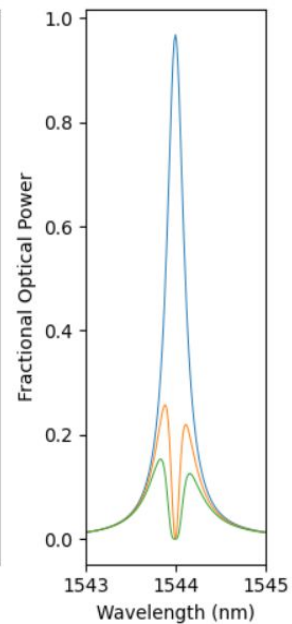
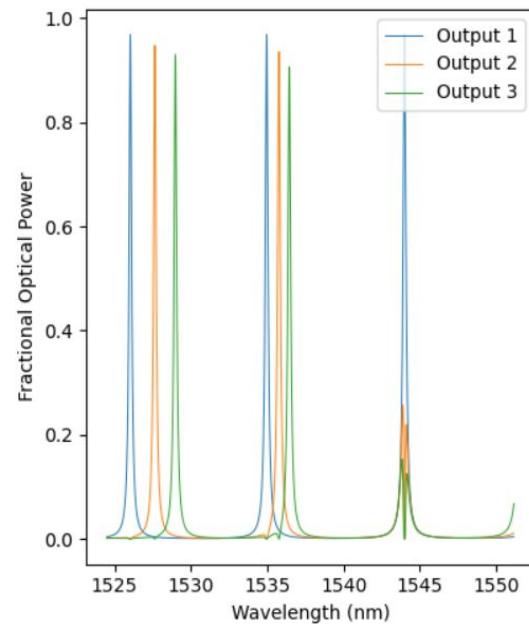
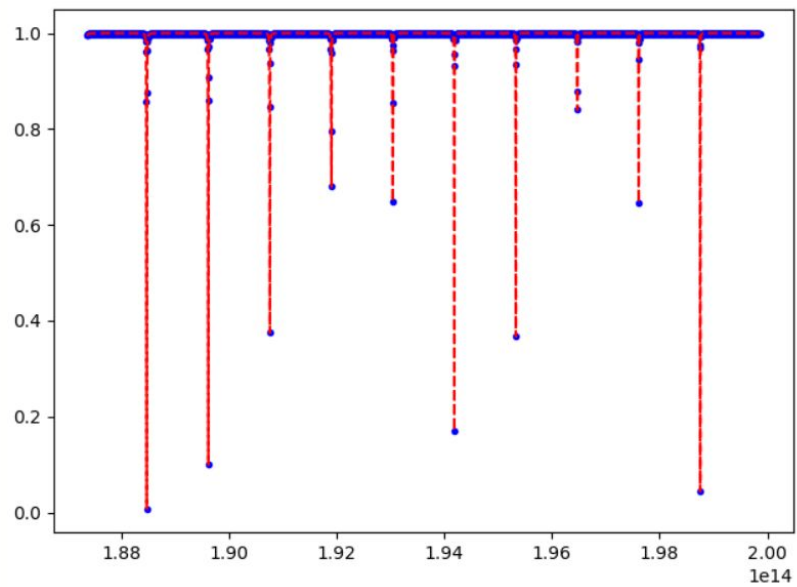
The filter design is a key to complete our image recognition.

Circuit design



Here is the circuit image. There are three input and three output.

However, the output1 is the inout2. By using the convolution (Filter), the image will be recognized step by step.



The problems we still need to fix:

1. The complex circuit need more than three convolution parts.
2. How to test the circuit's running test.
3. Get a complete output image.

Thank you