qDerm

Detecting Skin Cancer with **Quantum Machine Learning**

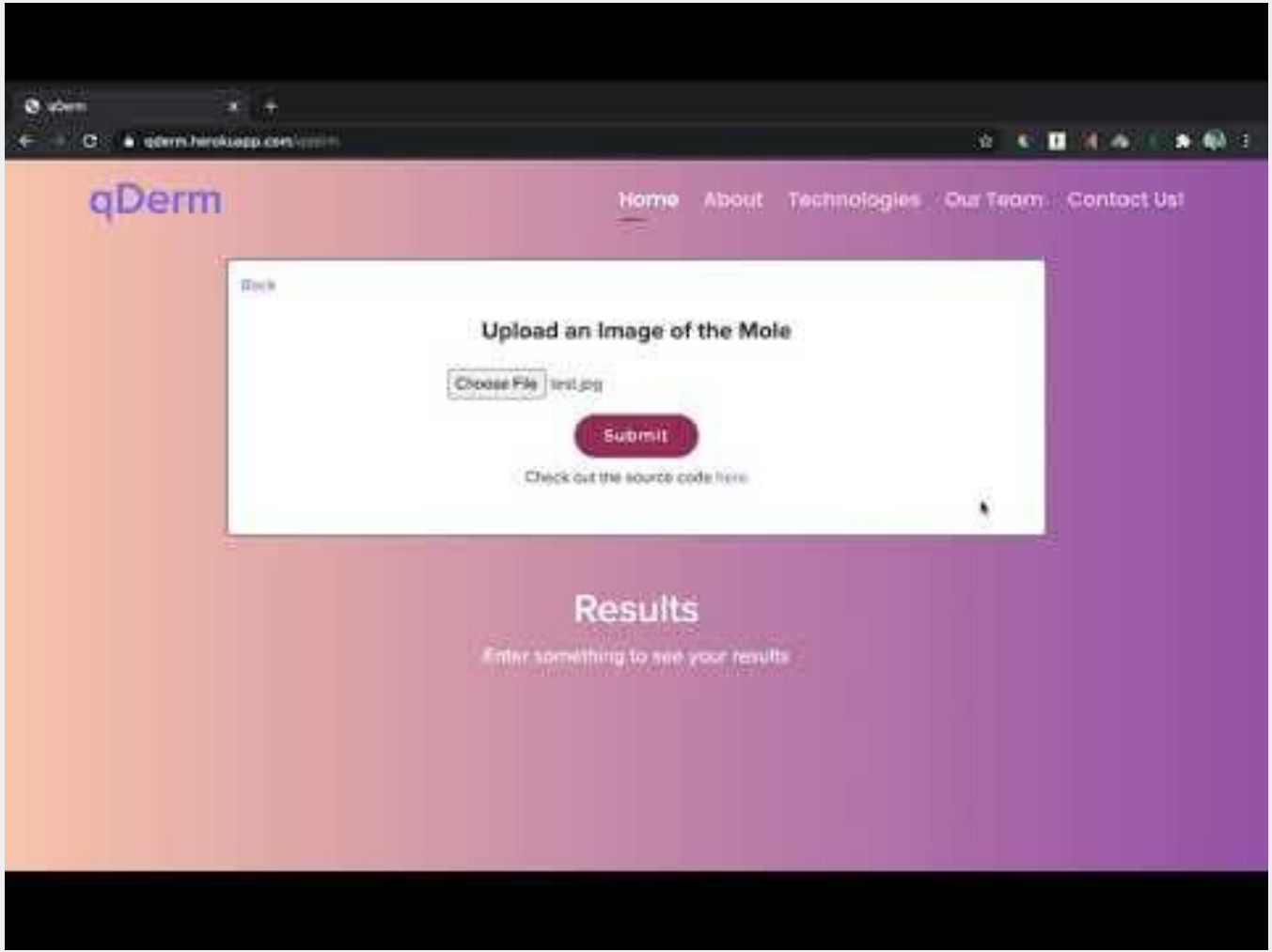
Quantum Computing

100M x faster

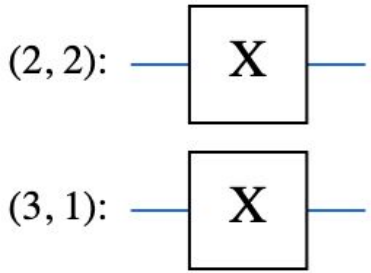
1000 x less energy

qDerm


Skin Cancer
qDerm


qDerm
Home
About
Technologies
Our Team
Contact Us!
Upload an Image of the Mole
Choose File
test.jpg
Submit
Check out the source code here
Results
Enter something to see your results

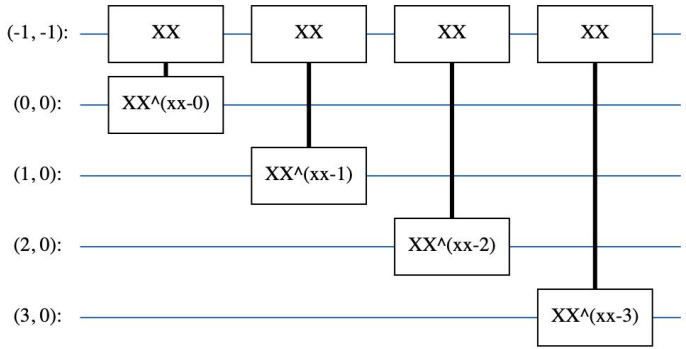
```
[ ]
def convert_to_circuit(image):
    """Encode truncated classical image into quantum datapoint."""
    values = np.ndarray.flatten(image)
    qubits = cirq.GridQubit.rect(10, 10)
    circuit = cirq.Circuit()
    for i, value in enumerate(values):
        if value:
            circuit.append(cirq.X(qubits[i]))
    return circuit
```




```
x_train_circ = [convert_to_circuit(x) for x in gray_X_train]
x_test_circ = [convert_to_circuit(x) for x in gray_X_test]

#convert cirq circuits to tensors for tfq:
x_train_tfcirc = tfq.convert_to_tensor(x_train_circ)
x_test_tfcirc = tfq.convert_to_tensor(x_test_circ)
```

+ Code

```
[ ] #QNN Implementation: classification is based on the expectation of t

class CircuitLayerBuilder():
    def __init__(self, data_qubits, readout):
        self.data_qubits = data_qubits
        self.readout = readout

    def add_layer(self, circuit, gate, prefix):
        for i, qubit in enumerate(self.data_qubits):
            symbol = sympy.Symbol(prefix + '-' + str(i))
            circuit.append(gate(qubit, self.readout)**symbol)
```

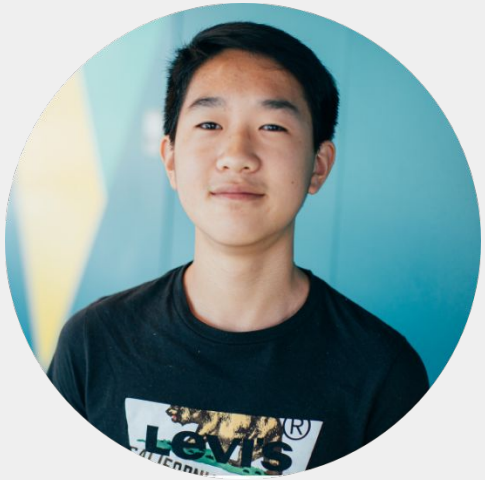


Results

```
[21] EPOCHS = 3
     BATCH_SIZE = 8

     NUM_EXAMPLES = 10

     x_train_tfcirc_sub = x_train_tfcirc[:NUM_EXAMPLES]
     y_train_hinge_sub = y_train_hinge[:NUM_EXAMPLES]

     qnn_history = model.fit(
           x_train_tfcirc_sub, y_train_hinge_sub,
           batch_size=8,
           epochs=EPOCHS,
           verbose=1,
           validation_data=(x_test_tfcirc, y_test_hinge))

     qnn_results = model.evaluate(x_test_tfcirc, y_test)
```

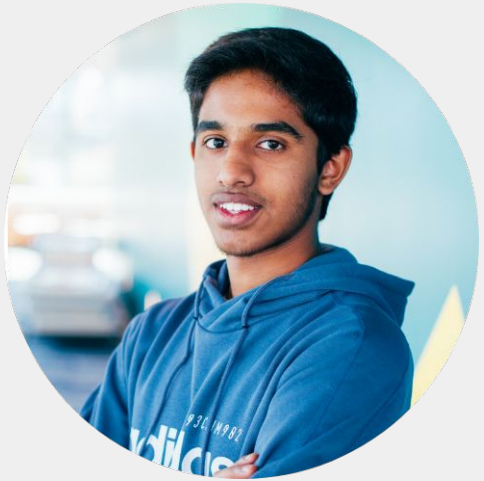
```
Train on 10 samples, validate on 660 samples
Epoch 1/3
10/10 [==============================] - 8s 793ms/sample - loss: 1.0000 - hinge_accuracy: 0.2500
Epoch 2/3
10/10 [==============================] - 7s 680ms/sample - loss: 1.0000 - hinge_accuracy: 0.2500
Epoch 3/3
10/10 [==============================] - 7s 679ms/sample - loss: 1.0000 - hinge_accuracy: 0.2500
```

Thank You!



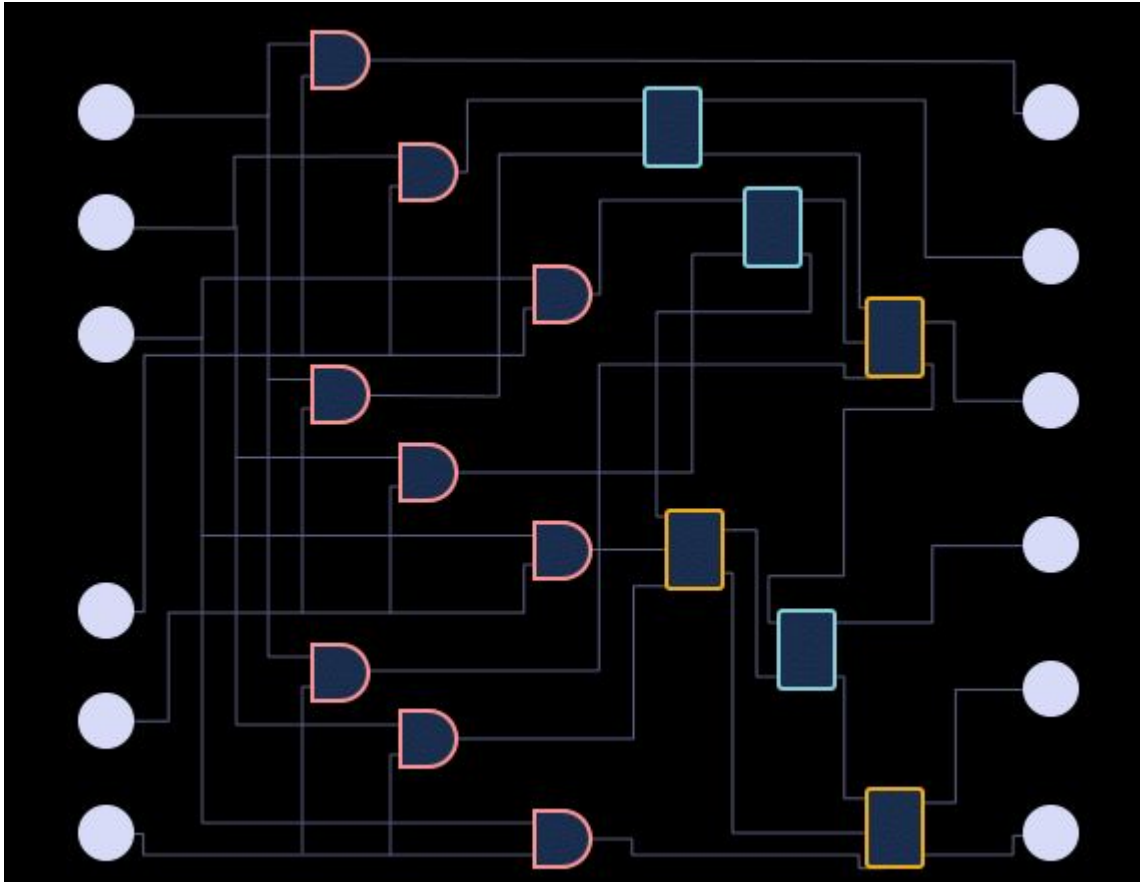
Kevin Wang

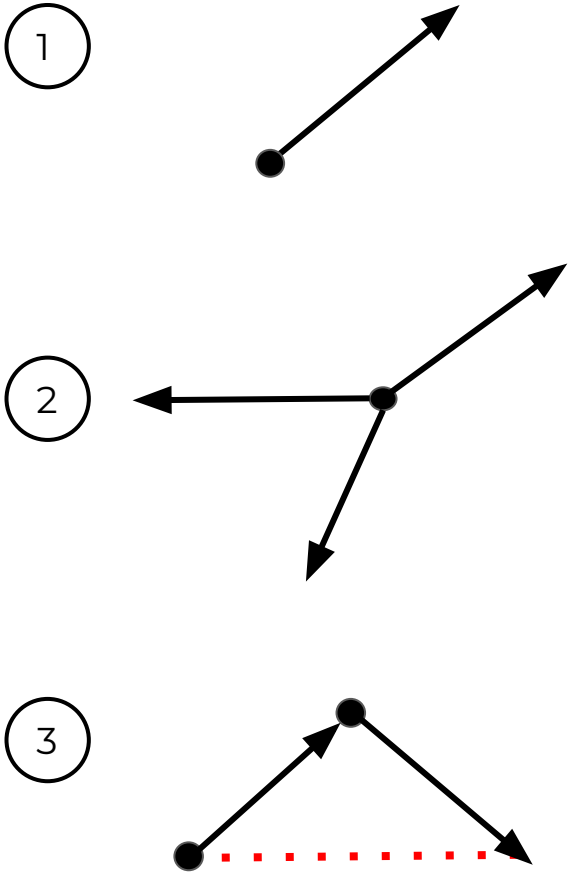
Alice Liu



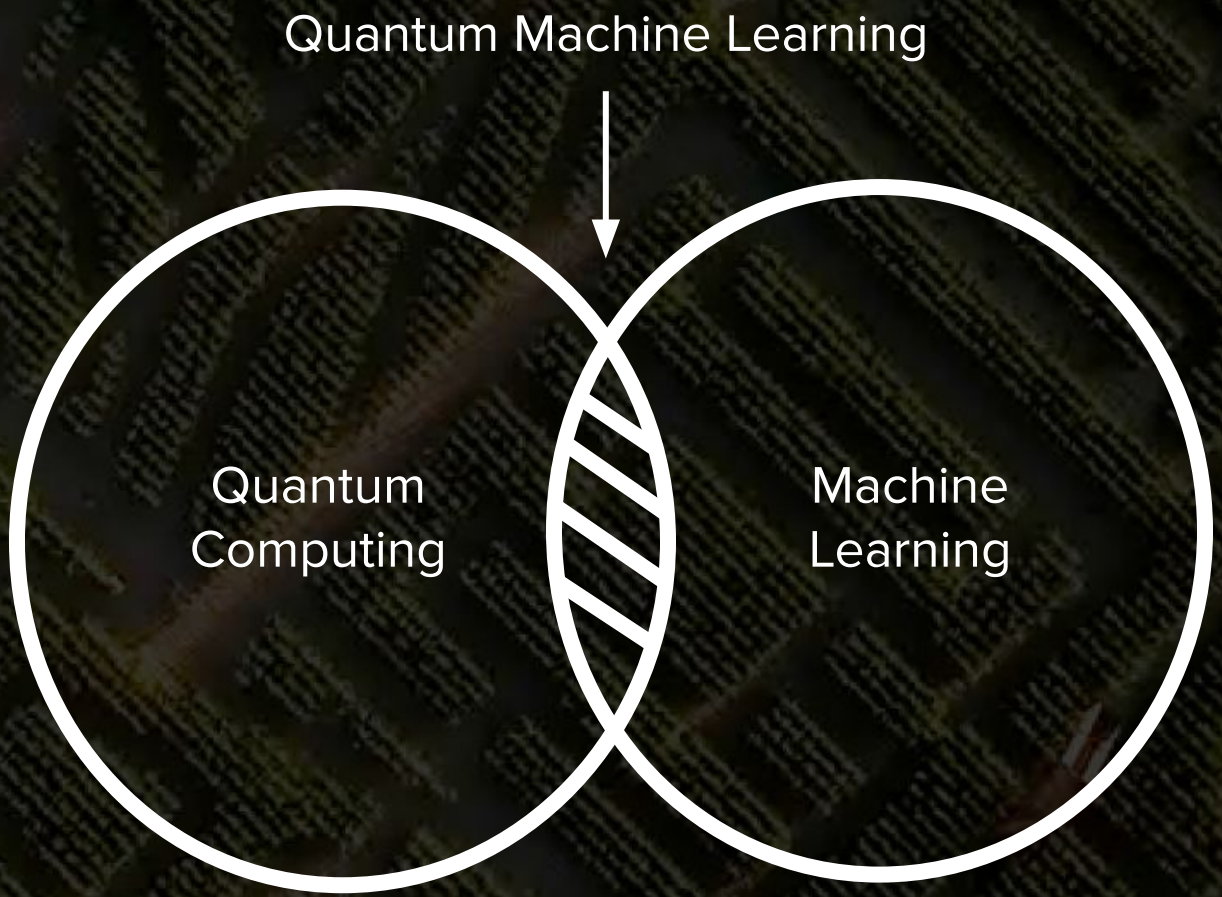
Sohail Mohammed

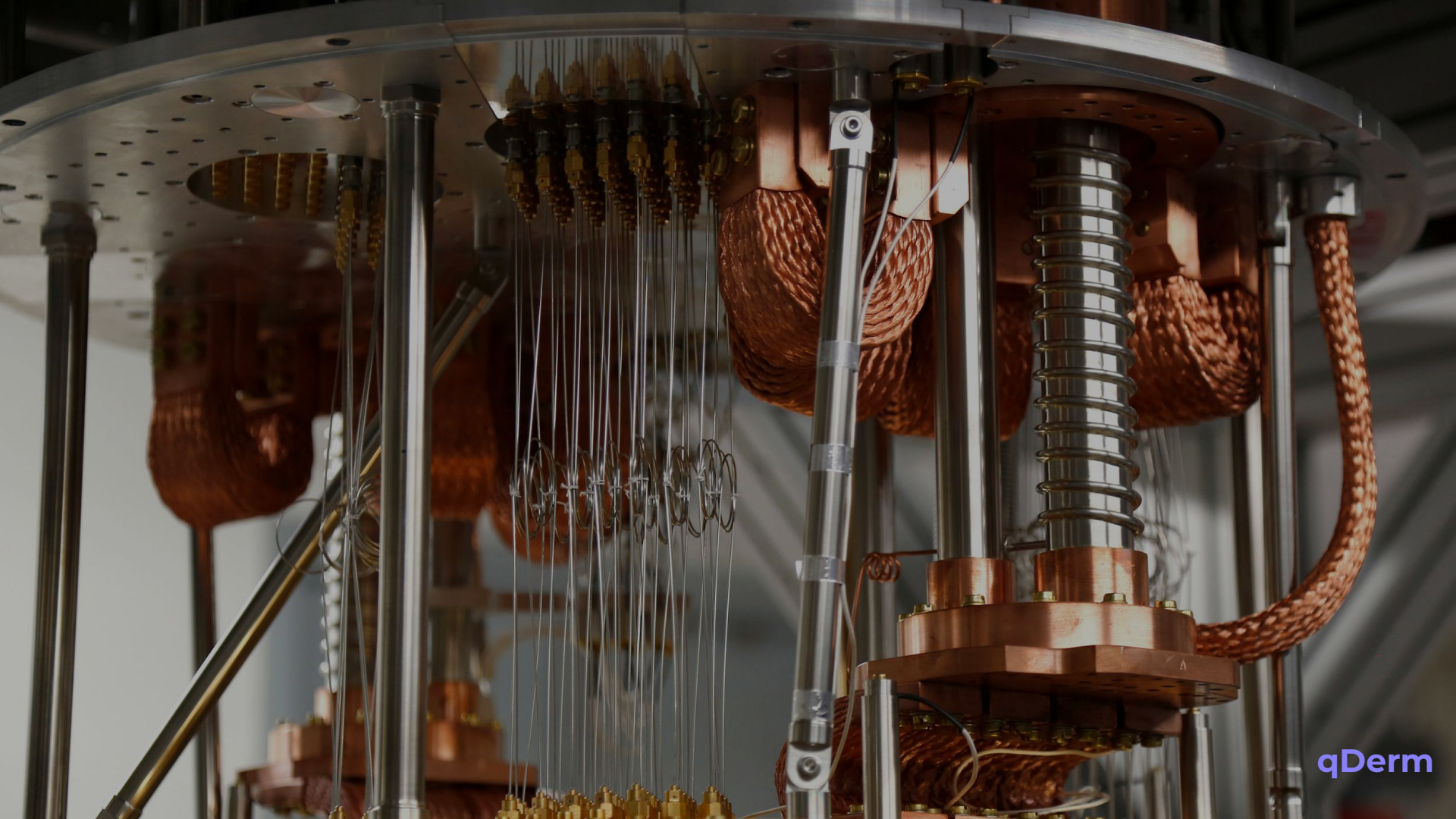
qDerm





Amplitudes → Optimization


Quantum Machine Learning
Quantum
Computing
Machine
Learning


qDerm

Appendix