Quantum Art

Using CV and Qiskit

Qiskit Summer Jam 2020

Alex Xu, Achilles Dabrowski, Franklin Boampong, Sam Lamba

Purpose

Quantum Art provides a fun interactive method to learn and experiment with the quantum world. It serves as an exciting tool for new learners to take the quantum world’s possibilities for a spin for the first time and create never-before-seen art.

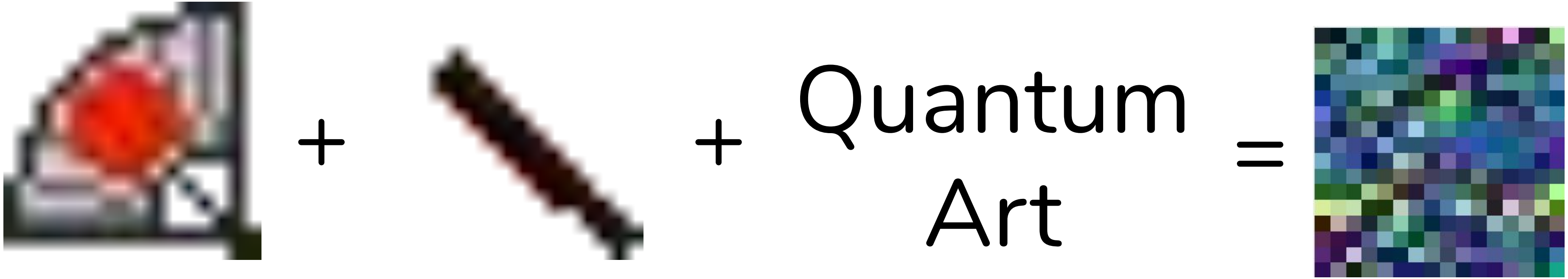
Using Quantum Computers

- Each of 2 images obtained from the user is converted to a 2 dimensional grid containing numbers representing the RGB value of each pixel
- Each cell in 2 n by m grid is accessed and the values are converted to binary
    - I.e 0b???? and 0b????
- The binary values are now entered into 15 quantum registers while maintaining a superposition between register 7 and 8 .i.e Based on the binary number, we set Quantum register to 1 or 0 using qc.x on quantum register.

Use Case

- Eg: If the two numbers from grid cells are 2 and 4 we convert the to 0000010(Base 2) and 0000100 (Base 2) respectively
- We then set exclusive values of quantum register 1 - 7 based on binary value of 2
- We also set exclusive values of quantum register 8 - 15 based on binary value of 4
- The quantum process then gives us new values with quantum registers after simulation
- Based on result we do a decoding process similar to encoding in binary to get results which is then stored in a new cell in a new n by m grid
- New 2 dimensional grid serves as new image generated by the process.

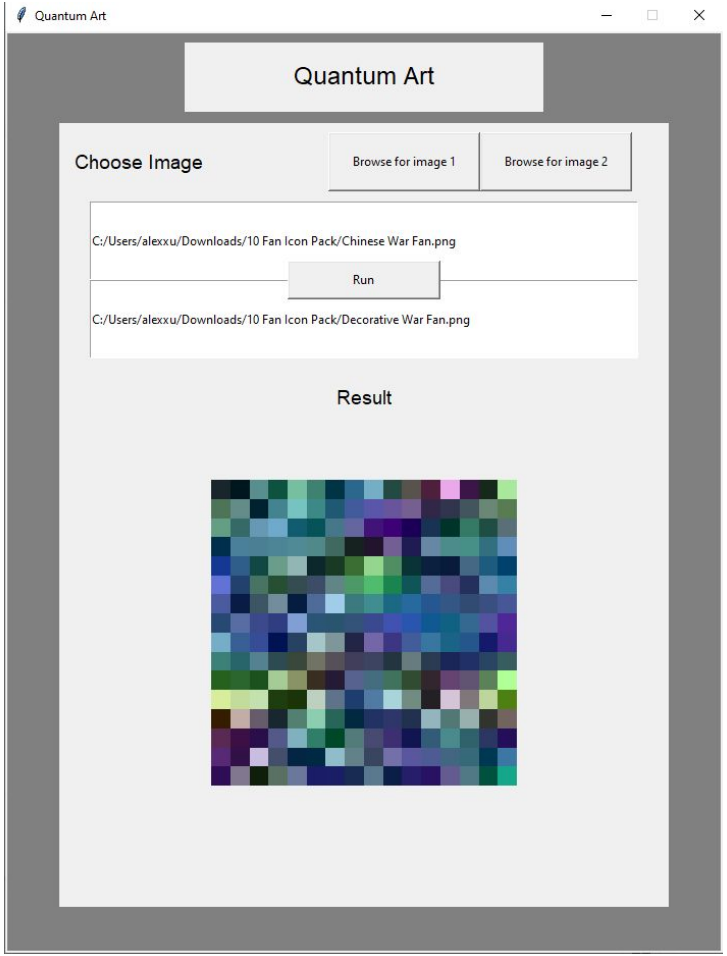
Workflow



Quick Usage Demo Video: https://youtu.be/_ItK1MHljmo

User Interface

- Allows user to select input images
- Easily view result
- Created using tkinter Python library



C:/Users/alexxu/Downloads/10 Fan Icon Pack/Chinese War Fan.png

C:/Users/alexxu/Downloads/10 Fan Icon Pack/Decorative War Fan.png

How We Built It

- Experimented using IBM Quantum Experience Notebooks (Jupyter) before porting to a local Python installation with Qiskit
- Four team members collaborated virtually over Zoom, GitHub, and CodeShare
- GUI was developed in parallel using tkinter
- Backend and Frontend Integration

Installation

Be sure to install and configure packages in an Anaconda Environment to support:

- Python 3.xx
- Pillow (PIL)
- Matplotlib
- Tkinter
- Numpy
- Sys
- Math
- Image_Merger
- Qiskit (Requires IBM Quantum Experience Key and setup)

Usage Instructions

1. Select 2 images (both 16x16 in dimension, PNG files)
2. Hit “Run”
3. Wait for results to show up!
4. Result image will be saved as “result.jpeg” in the root of the repository.

Team

Alex Xu (Duke ‘22)

Achilles Dabrowski (Duke ‘22)

Franklin Boampong (Duke ‘22)

Sam Lamba (Duke ‘22)