Generating Art using Quantum Convolutional Neural Networks and Bose-Einstein Condensates

Quantum Ordinals

June 26, 2023

Abstract

In this white paper we demonstrate a novel method for generating stylized images using Quantum Convolutional Neural Networks ran on IBM Quantum Computers and raw images from Bose-Einstein Condensates (BEC) obtained from experiments using the ColdQuanta Albert system.

1 Introduction

Within the last few years there has been a huge rise in the number of potential applications for Artificial Neural Networks. One of the potential applications is that of image processing and more specifically styling transfer. Styling transfer is where an image is created such that it has the content of one image but is in the style of another image. Convolutional Neural Networks (CNN) have shown much promise in this application given that they are designed to analyse pixel imagery. The power behind CNN's lay in their ability to determine particular features in a given image. This is due to the fact that they incorporate two types of layers known as the convolutional layer and pooling layer respectively.

1.1 Convolutional Neural Networks for styling transfer

Convolutional Neural Networks can be used to transfer the style of an image to the content of another image. This is done by feeding the CNN a style image and a content image, The style image contains the style that we wish to extract. The content image is the image that we wish to style. The CNN transfers the style by taking the images through a number of convolutional and pooling layers in a feed forward manner. Each layer acts as an image filter and extracts certain patterns and features using a number of image filters and feature maps. A layer generally has N_i filters and N_i feature maps of size M_i. The first layers will represent lower feature such as edges and textures while the final layers will represent higher level content such as windows of a house or leaves on a tree. The style from the style image is captured using a feature space that captures texture information rather than content.

This entire process can be generalized as loss function that is minimized:

$$L_{total}(\vec{b}, \vec{a}, \vec{x}) = \alpha L_{content}(\vec{b}, \vec{x}) + \beta L_{style}(\vec{a}, \vec{x}) \tag{1}$$

Where $\vec{a}$ and $\vec{b}$ are the styling and content images respectively. α and β are the weights for the content and style reconstructions.

1.2 Quantum Convolutional Neural Networks

Quantum computing has immense potential for a wide number of applications particularly in the development of neural network models. This is due to the fact that quantum computers can make use of quantum parallelism which can make neural network models more efficient and perform better. As such in our method we use the quantum analogue of a CNN known as a Quantum Convolutional Neural Network. These work in the same way as normal Convolutional Neural Networks except that images are encoded on to a quantum circuit in order to run on a quantum computer. This is done by encoding the image on to a quantum circuit using a feature map such as a ZZ feature map. Quantum feature maps encode features on to a Hilbert space where feature vectors are essentially quantum states.

For example consider a feature vector X that is encoded using a quantum feature map such that:

$$x \rightarrow |\phi(x)\rangle \tag{2}$$

This is normally implemented using a variational quantum circuit consisting of CNOT gates, RZ and RY gates where the rotation angles depend on the input image data.

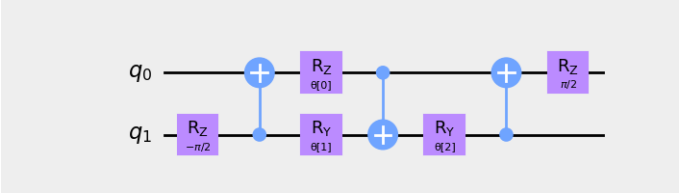


Figure 1: Example of a parameterized circuit within a QCNN

1.3 IBM Quantum Devices

For implementing the QCNN we used quantum computers provided by IBM. These quantum computers make use of a superconducting qubit known as a Transmon qubit. These are a type of qubit that make use of a superconducting metal that is cooled to very low temperatures (typically below 100nk). Specifically the Transmon qubit is a form of charge qubit that is designed to be less sensitive to charge noise. This form of qubit is made up of two main components. A gate capacitor and a Josephon junction. In between the capacitor and Josephon junction is a superconducting island. The ground and excited states of the qubit $|0\rangle$ and $|1\rangle$ are based upon the number of cooper pairs (pairs of electrons bound together at low temperatures) within the superconducting island.

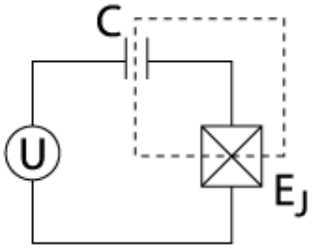


Figure 2: Diagram of a charge qubit

Qubits can be used to do computations using quantum logic gates, There are two main types of logic gates. Single qubit logic gates that act on a single qubit and multi-qubit gates that act on multiple qubits. Some of examples of single qubit gates are the Pauli gates (X,Y,Z) which each perform a rotation of π radians around a specific axis of the qubits hilbert space.

For example using matrix multiplication we can see how a Pauli-X gate operates on a qubits state. Each quantum logic gate will have a corresponding unitary matrix. These matrices are a compact way of describing the operation of the logic gate. For example the Pauli-X gate is described by the matrix:

$$X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \tag{3}$$

Using this matrix we can find out how this gate will affect the qubits state by multiplying the matrix by the column vector of the qubits state. For example if our qubits state is $|0\rangle$ then its column vector is $\begin{bmatrix} 1 \\ 0 \end{bmatrix}$ which we multiply by the Pauli-X matrix:

$$\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 0(1) + 1(0) \\ 1(1) + 0(0) \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix} = |1\rangle \tag{4}$$

From this we can see that the Pauli X gate will flip the qubit from $|0\rangle$ to $|1\rangle$.

If we first initialise the qubit to $|1\rangle$ and multiply by the Pauli X gate:

$$\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0(0) + 1(1) \\ 1(0) + 0(1) \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix} = |0\rangle \tag{5}$$

Which is correct as the qubit has flipped from $|1\rangle$ to $|0\rangle$.

Another important quantum logic gate is the Hadamard gate. This puts a qubit in to a superposition of states such that if the qubit is $|0\rangle$ then the state will become:

$$|0\rangle \longrightarrow \frac{|0\rangle + |1\rangle}{\sqrt{2}} \tag{6}$$

and if the state is $|1\rangle$ then the state will become:

$$|1\rangle \longrightarrow \frac{|0\rangle - |1\rangle}{\sqrt{2}} \tag{7}$$

The matrix for the Hadamard gate is as follows:

$$H = \tfrac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \tag{8}$$

If we initialise the qubit to $|0\rangle$ and apply a Hadamard gate:

$$\frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1(1) + 1(0) \\ 1(1) + -1(0) \end{bmatrix} \tag{9}$$

$$= \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ 1 \end{bmatrix} \tag{10}$$

Which has mapped $|0\rangle \longrightarrow \frac{|0\rangle + |1\rangle}{\sqrt{2}}$

Rotational gates such as the RZ and RY gates are very important for implementing QCNNs as they can be used to encode image data on to the qubit by encoding the data as specific rotation angles. An example of a multi-qubit gates is the CNOT gate. This flips a qubits state from $|0\rangle$ to $|1\rangle$ if a control qubits state is $|1\rangle$. This is also used to entangle qubits if one of the qubits is first put in to superposition using a Hadamard gate.

IBM quantum devices implement multiple qubits on to a chip (normally between 5 to 433 as of current). The qubits are controlled and measured using control and readout resonators. Multi-qubit gates are implemented using bus resonators. A typical device can be seen in the figure below:

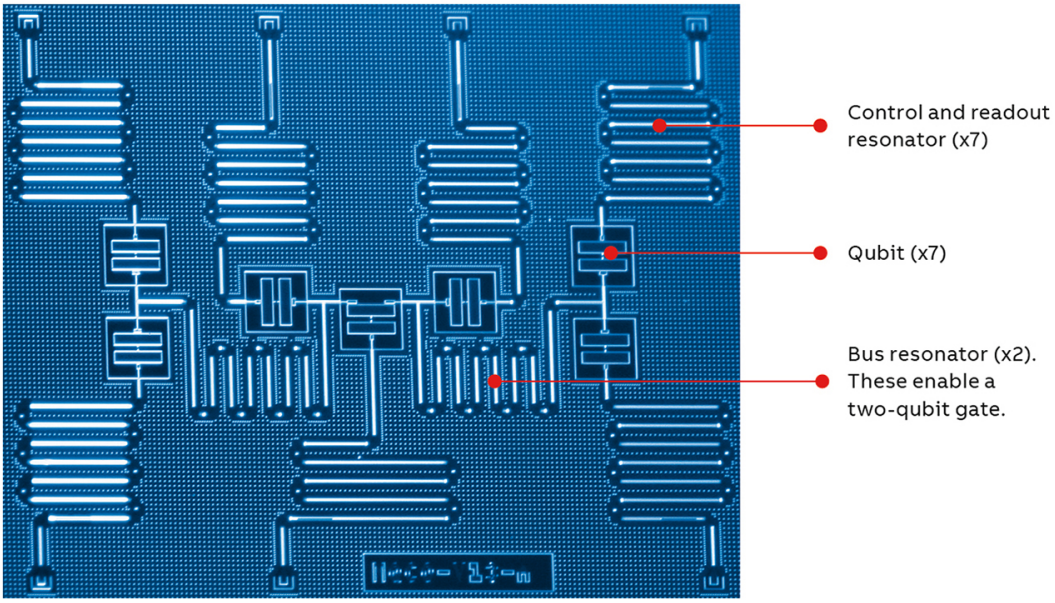


Figure 3: Figure showing a 7 qubit device

1.4 Bose-Einstein Condensates

Raw images for generating the art was obtained through experiments with Bose-Einstein condensates. A Bose-Einstein condensate is a phase of matter that is formed when a gas of bosons (such as Rubidium) is cooled to around $-273.15°C$. When a BEC is formed the bosons comprising it will occupy the lowest quanntum state and form a single quantum wave.

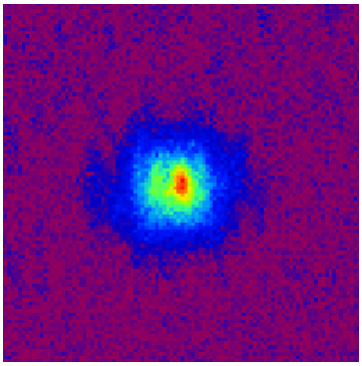


Figure 4: Time of Flight image showing a BEC obtained in one of our experiments. In this image a BEC was formed from 1309 condensed rubidum atoms in a magneto optical trap.

1.4.1 Magneto Optical Traps

The main method of generating a Bose-Einstein condensate is with a piece of hardware called a magneto optical trap. This uses laser cooling and a magnetic field to trap and generate ultracold matter. The main method of cooling is doppler cooling where atoms lose momentum and therefore energy as they scatter photons from the laser.

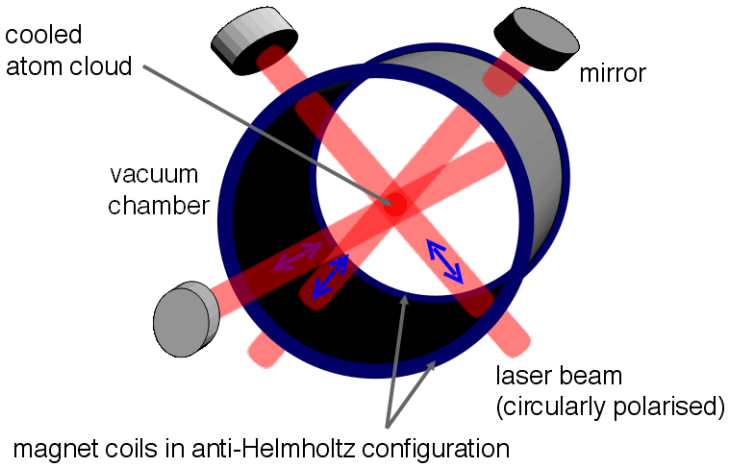


Figure 5: Diagram of Magneto Optical trap for generating Bose-Einstein condensates

2 Materials and Methods

This section pertains to the materials and methods for generating stylized images using QCNNs and raw images of Bose-Einstein condensates from experiments using the ColdQuanta Albert system.

2.1 Running BEC experiment and obtaining raw images

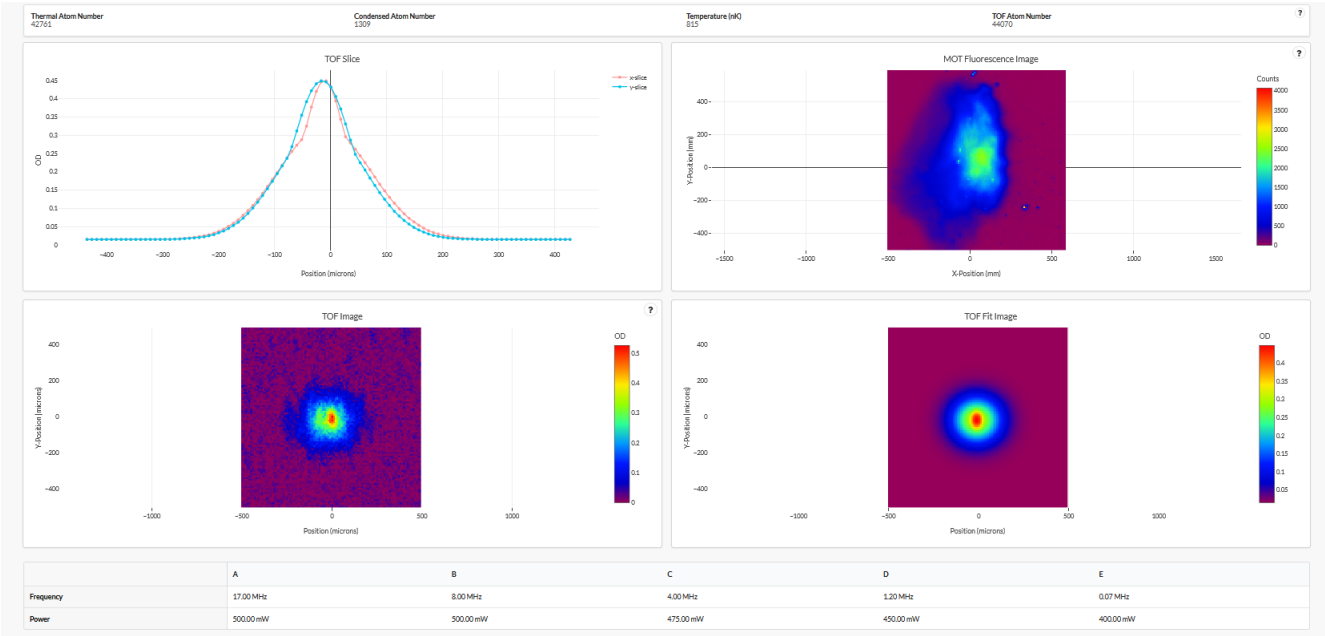
The first part of the method is to run a BEC experiment on the ColdQuanta Albert system. The Albert system is a cloud based ultracold matter experimentation platform allowing users create Bose-Einstein Condensates remotely. The system makes use of a tight magnetic trap on an atom chip which uses Radio frequency (RF) evaporation to get rid of the more thermally excited atoms from the cloud. This leaves the colder atoms which can occupy the lowest quantum state and thus form a BEC. In our experiments we formed a Bose-Einstein Condensate by setting the following key parameters on the Albert system:

- Time of flight (in milliseconds): This is the amount of time that the atom cloud is allowed to expand. Warmer atoms will move further away from their trapped positions than colder atoms.
- RF evaporation frequencies (MHz): These are the frequencies for each RF evaporation sequence (four in total each with a duration of 400 milliseconds). Each frequency will dictate which atoms are ejected from the trap. Higher frequencies will eject higher energy atoms while lower frequencies will eject lower energy atoms.
- RF power (mW): Higher power will remove more atoms during each evaporation sequence while a lower power will eject less.

During our experiments we found that the following values formed the best condensate:

	A	B	C	D	E
RF Frequency (MHz)	17.00	8.00	4.00	1.20	0.07
RF Power (mW)	500.00	500.00	475.00	450.00	400.00

Once the parameters have been tuned they are sent off to the Albert system where the BEC experiment is performed. After the experiment the results are sent back including the raw images as in the figure below:



	A	B	C	D	E
Frequency	17.00 MHz	8.00 MHz	4.00 MHz	1.20 MHz	0.07 MHz
Power	500.00 mW	500.00 mW	475.00 mW	450.00 mW	400.00 mW

Figure 6: Results obtained from a BEC experiment using the parameters in the table above

The Albert system will return three key images:

- MOT Fluorescence Image: This is an image of the atoms in the Magneto Optical Trap prior to a condensate being formed.
- Time of Flight (TOF) Image: This is an image of the atoms after a BEC has been formed. If the BEC has been successfully formed then a dense core should form within the middle of the image. This is the image that we use as the content image in later steps.
- Time of Flight (TOF) Fit Image: This is a 2D bi-modal fit of the raw TOF image. These are also used as content images if the raw TOF image contains too much noise.

Figure 7: ColdQuanta Albert System

2.2 Generating stylized image using QCNN

After the raw images have been obtained they can be used as content images for generating art using the QCNN. In this section we will go through all the steps required to generate a stylized image.

2.2.1 Step 1: Use QCNNLite to extract the style and content from the input images

The first step is to extract the content and style from the input images using the QCNN.

The QCNN used is a bespoke QCNN called QCNNLite. This is a hybrid model where the higher layers of the convolutional and pooling layers are classical but the lower layers are encoded on to qubits on an IBM quantum computer. The main reason for this hybrid approach is that quantum devices currently have a low qubit count (5 to 433 qubits) and thus cannot fully encode a large resolution image. However a CNN will reduce the spatial dimensions of the data and increase the depth after going through each pooling layer. Therefore the last few layers will have low spatial dimensions and as such can be realised as a quantum circuit with a high circuit depth (number of quantum gates). It is important to mention that the classical layers are executed locally using a classical computer while the quantum layers are executed on a backend quantum device remotely. Once the quantum circuit has been executed the results are returned back and the measurements decoded. QCNNLite was implemented in Python using Tensorflow for the classical layers of the QCNN. The quantum layers were implemented using a Python library called Qiskit. This library allows users to create quantum circuits and then run them on a backend quantum device. A full overview of QCNNLite can be seen in the figure below.

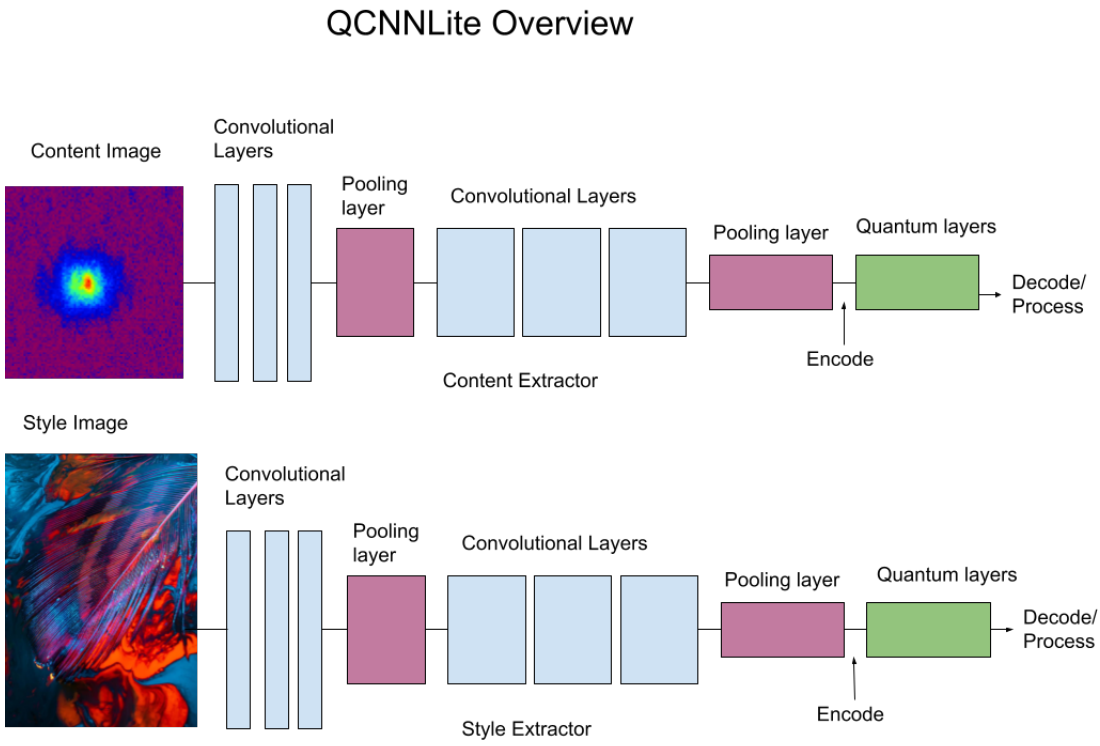


Figure 8: Diagram showing a basic overview of QCNNLite for style and content extraction

2.2.2 Step 2: Train the output image and apply gradient descent

Now that we have extracted the target values for the content and styles from QCNNlite we can now generate an output image with the content of the content image and the style of the styling image. This is done by first taking initializing a new image that is a copy of the content image. Once this is done we can begin training the output image as seen in the steps below:

1. Run the output image through the QCNNLite extractor
2. Obtain the mean-square error compared to the styling and content images
3. Compute gradients
4. Apply gradients using Adam optimizer
5. Repeat

Once the training steps are complete the output image should have the content of the content image but in the style of the styling image.

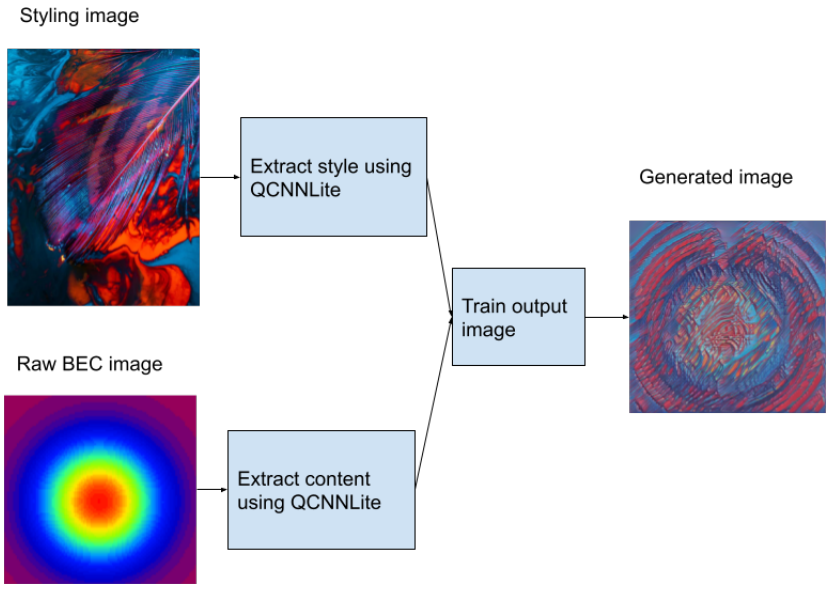


Figure 9: Diagram showing the basic process of generating a stylized image by extracting features from the input images using QCNNLite and then training an output image

3 Conclusion

In conclusion this white paper has shown how Quantum Convolusional Neural Networks can be used to generate styled images using styling images and time of flight images obtained from BEC experiments. It has also shown that despite the low qubit count of current NISQ quantum hardware it is still possible to use the devices to generate images. This is providing that the QCNN includes classical layers to handle higher data dimensions that can then be pooled to the quantum layers at the expense of high circuit depth.