DISSERTATION

*Abstract*—This study is to evaluate the performance of the state-of-the-art CNN models, namely Alexnet, VGGNet, Inception, ResNet and DenseNet on F-MNIST dataset and then try to improvise it through CapsNet. Also, this study aims to explore the possibilities of Quantum Computing in image recognition.

# Introduction

Modern CNN models are quite good to reconstruct the positional information of an image, but they are not quite accurate due the information loss endured in the pooling layers by retaining the noticeable features only and ignoring the less importance ones. These models give importance in relations of local orientations in an image, but they do not appear to create the global shape representation like a biological vision.

The capsule network which was introduced by Sabour et al. (2017), captures different positional information of the same feature in the image by replacing the scalar output of a classical CNN neuron with Capsules, which create a vector output. It also identifies the part to whole spatial relationship by implementing dynamic routing algorithm between capsules.

Quantum computing on other hand, defines a probabilistic approach of representing classical information using quantum theories. Much research is happening around image representation, image encoding and image processing by using QSobel edge extraction, quantum geometric transformation, quantum image segmentation etc. To get a more clearer picture of the mechanics of the quantum image model, the plan is to explicitly store the quantum states classically and measured the images to demonstrate the effect of the image processing operations.

# Exploring Classical Cnn models

## Model Selection

Given the vast amount of information available, uunderstanding various convolutional neural network (CNN) architectures can be tough. However, to test the performance of the classical CNN model, I intent to use the most popular models, used by the data scientists today.

**AlexNet** came up as the winner of the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) classification contest in the year of 2012.

**VGGNet** was the runner up of the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) classification contest held in the year of 2014.

**ResNet** secured 1st Position in ILSVRC and COCO 2015 competition with just error rate of 3.6% of error rate.

**Inception-v3** with 144 crops and 4 models ensembled, managed to obtain the top-5 error rate of 3.58% , and finally obtained 1st Runner Up (image classification) in ILSVRC 2015.

**DenseNet** implements dense concatenation of all subsequent layers to avoid using direct summation and preserves the features in preceding layers. When the parameters are less, DenseNet have proved to be better efficient than ResNet.

## Data Collection

Comparative study has been carried out in choosing the correct dataset for this work. ImageNet and Google Open Images though have many varieties and numbers, have excluded them because of the computation time and power needed. Being a part of the retail industry, I have ignored the CIFAR and COCO datasets and choose Fashion MNIST to carry out the work.

|  |  |  |  |
| --- | --- | --- | --- |
| Dataset Name | No. Of Images | Pixel Size | No. of Categories |
| FMNIST | 70,000 | 28X28 | 10 |
| ImageNet | 14,00000 | Varying | 20,000 |
| CIFAR | 50,000 | 32X32 | 10 |
| Open Images V5 | 90,00000 | Varying | 600 |
| COCO | 200000 | Varying | 80 |

*Table1 : Brief about popular datasets*

# Data Analysis

## AlexNet

The architecture of AlexNet have 8 layers and about 60,000 total parameters. The layers are - five convolutional layers, and three fully connected layers. The model used ReLU activation functions, which helped reduce the training time of the network, and was a current solution to the vanishing gradient problem. The pooling layers have a stride of length 4 pixels, which significantly reduced the error of their model.

Diagram, engineering drawing

Description automatically generated

*Fig1 : AlexNet architecture*

A picture containing text, crossword puzzle, receipt

Description automatically generated

*Fig2 : AlexNet network structure*

To reduce overfitting they have used runtime data augmentation as well as a regularization method called dropout [1]. In data augmentation, they have extracted translated and horizontally reflected 10 random patches of 224 × 224 images and also used principal component analysis (PCA) [2] for RGB channel shifting of training images.

As part of the study, an AlexNet model is built and trained with F-MNIST dataset. Below is the result obtained after completion of the training and followed by testing the model output.

|  |  |  |
| --- | --- | --- |
| **Epochs** | **Average Loss** | **Accuracy** |
| Epoch1 | 0.3817 | 82% |
| Epoch2 | 0.4204 | 84% |
| Epoch3 | 0.4007 | 85% |
| Epoch4 | 0.3922 | 86% |
| Epoch5 | 0.3765 | 86% |
| Epoch6 | 0.3603 | 87% |
| Epoch7 | 0.3559 | 87% |
| Epoch8 | 0.3250 | 88% |
| Epoch9 | 0.2959 | 89% |
| Epoch10 | 0.2900 | 89% |

*Table2 : Experiment output of AlexNet*

## VGGNet

The major problem with AlexNet is to deal with too many hyperparameters. VGG Net[9] proved more efficient in that respect by replacing large kernel-sized filters 11 and 5 in the first and second convolution layer, with multiple 3×3 kernel-sized filters one after another.

The structural details of a VGG16 architecture is given below.

Diagram

Description automatically generated

*Fig3 : VGGNet architecture*

A picture containing text, receipt

Description automatically generated *Fig4 : VGGNet network structure*

The model has been tested with F-MNIST dataset and below is the result.

|  |  |  |
| --- | --- | --- |
| **Epochs** | **Average Loss** | **Accuracy** |
| Epoch1 | 1.1091 | 61% |
| Epoch2 | 0.6679 | 76% |
| Epoch3 | 0.5013 | 82% |
| Epoch4 | 0.4141 | 86% |
| Epoch5 | 0.3351 | 89% |
| Epoch6 | 0.2704 | 91% |
| Epoch7 | 0.2711 | 91% |
| Epoch8 | 0.2273 | 92% |
| Epoch9 | 0.2175 | 92% |
| Epoch10 | 0.1864 | 93% |

*Table3 : Experiment output of VGGNet*

## Inception

#### All the previous CNN models focused to a fixed filter size while performing the convolution operation. Inception model talks about filters having different dimensions followed by max pooling.

**Diagram

Description automatically generated**

*Fig5: Inception model with 5x5 and 3x3 filters*

Inception modules[5][7] after the factorization of the n × n convolutions. In the below proposed architecture, value of n is considered as 7 for the 17 × 17 grid.

**Diagram

Description automatically generated**

*Fig6: after factorization of nxn* *convolutions*

Inception modules with expanded the filter bank outputs. The Inception architecture is used on the coarsest (8 × 8) grids to promote high dimensional representations

**Diagram

Description automatically generated**

*Fig7: with expanded filter bank outputs*

The inception model has 27 layers consisting of 9 inception blocks

A picture containing text

Description automatically generated

*Fig8: Inception model architecture*

The model has been tested with F-MNIST dataset and below is the result.

|  |  |  |
| --- | --- | --- |
| **Epochs** | **Average Loss** | **Accuracy** |
| Epoch1 | 0.3854 | 82% |
| Epoch2 | 0.3441 | 85% |
| Epoch3 | 0.3454 | 85% |
| Epoch4 | 0.3606 | 86% |
| Epoch5 | 0.3493 | 87% |
| Epoch6 | 0.3443 | 87% |
| Epoch7 | 0.3434 | 87% |
| Epoch8 | 0.3417 | 88% |
| Epoch9 | 0.3419 | 88% |
| Epoch10 | 0.3417 | 88% |

*Table4 : Experiment output of Inception*

## ResNet

The aim of ResNet is to remove the identity blocks by applying “skip connections”[4][8]. The skip connection ensures that learning the identity function will help to perform the higher layers as good as the lower layers.

Diagram

Description automatically generated

*Fig9: skip connection in ResNet*

The final output, H(x) = F(x)+x ……for the ResNet

And for standard CNN, it will be H(x) = x ……. A desired output.

So, to make the ResNet provide the desired output, F(x) must be equal to 0. The function F(x) =0 is called the residual function.

There are multiple versions of ResNet implemented over the years. For this study, I will be considering the ResNet50 architecture model.

Table

Description automatically generated

*Fig10: Network diagram for ResNet*

The below data has been captured by training and testing the F-MNIST dataset with ResNet50

|  |  |  |
| --- | --- | --- |
| **Epochs** | **Average Loss** | **Accuracy** |
| Epoch1 | 0.5024 | 81% |
| Epoch2 | 0.4579 | 83% |
| Epoch3 | 0.4451 | 83% |
| Epoch4 | 0.4285 | 84% |
| Epoch5 | 0.4242 | 84% |
| Epoch6 | 0.4078 | 85% |
| Epoch7 | 0.3967 | 85% |
| Epoch8 | 0.3901 | 86% |
| Epoch9 | 0.3823 | 86% |
| Epoch10 | 0.3774 | 86% |

*Table5 : Experiment output of ResNet*

## DenseNet

The CNN models designed previously failed to answer on how deep the network would be[8]. Experiments showed that more the deep network, doesn’t yield better the result. DenseNet [9]architecture ensures that information loss doesn’t happens as we go deeper the network. As the layers in this model receive feature maps from all the preceding layers, the network will be thinner and more compact[6].

Diagram

Description automatically generated with low confidence

*Fig11: DenseNet layers*

The layer between two adjacent dense blocks are referred to as transition layer and change feature map size via convolution and pooling. The DenseNet architecture is as below:

Table

Description automatically generated with low confidence

*Fig12: DenseNet network diagram*

The below data has been captured by training and testing the F-MNIST dataset with DenseNet.

|  |  |  |
| --- | --- | --- |
| **Epochs** | **Average Loss** | **Accuracy** |
| Epoch1 | 0.6335 | 77% |
| Epoch2 | 0.4017 | 86% |
| Epoch3 | 0.3503 | 88% |
| Epoch4 | 0.3164 | 89% |
| Epoch5 | 0.2889 | 90% |
| Epoch6 | 0.2666 | 90% |
| Epoch7 | 0.2441 | 91% |
| Epoch8 | 0.2354 | 91% |
| Epoch9 | 0.2191 | 92% |
| Epoch10 | 0.2083 | 93% |

*Table6 : Experiment output of DenseNet*

# findings and discussion on classical CNN models

It has been clear from the above tables, that most popular classical CNN models doesn’t fit well for image recognition due to ignoring spatial position of the image parts and due to the poling which leads to information loss. Data augmentation may produce some better result, but it never solves the main problem and rather the model becomes computational heavy. The aim of this study is to find out a method which will address the problems described above and provide a solution rather than a workaround. So, the problem statement for this study will be like –

|  |
| --- |
| Design a model which will provide accuracy more than 93% for the standard FMnist image database.  This we will try to achieve through CapsNet and Quantum computing. |

Quantum computing is mostly in research phase and not much industrial scale implementation done with the help of quantum algorithms. So, in this study, we will try to explore the opportunities it can bring in solving image classification problems.

# Exploring capsnet for image recognition

## Overview

To overcome the spatial relation and information loss problem occurs inn classical CNN models while performing the data augmentation, Hinton and Sabour drew inspiration from neuroscience. In their study, they tried to explain the functionality of the brain on how it is organized into multiple self-functioning modules, which can be think of as capsules, who internally implements dynamic routing algorithms to find out the features of objects like pose (position, orientation, size, deformation, velocity, albedo, hue, texture, and so on). This research was published in 2017, in their paper titled [Dynamic Routing Between Capsules](https://arxiv.org/abs/1710.09829)[11]

Diagram

Description automatically generated

*Fig13: Difference between a Capsule and a neuron*

Unlike classical CNN models, where information stored as a scalar, capsules store them as a vector. Capsules assumes that at each location in the image, there is at most one instance of the type of entity. This particular assumption, which was motivated by the perceptual phenomenon called "crowding" (Pelli et al. [2004]), removes the binding problem (Hinton [1981a]) and allows a capsule to use a distributed representation, which is its activity vector, to encode the instantiation parameters of the entity of that type at a given location[10]. Hence, this distributed representation in form of capsules, is exponentially efficient than encoding the instantiation parameters by activating a point on a high-dimensional grid and can then take full advantage of the idea that spatial relationships can be modelled by matrix multiplies.

## CapsNet Architecture

Diagram

Description automatically generated

*Fig14: CapsNet architecture*

This is a simple capsnet with 3 layers. Wij is a weight matrix between each ui; , ε (1, 32 ×6 ×6) in Primary Capsules and vj , j ε (1, 10).

Diagram

Description automatically generated

*Fig15: 3 layers CapsNet flow*

This is the capsnet architecture for MNIST, which is described by Hilton in his paper for dynamic routing in capsnet. Primary capsules, takes the input from the image and acts as the lowest level of multidimensional entities.

| **Layer Name** | **Apply** | **Output shape** |
| --- | --- | --- |
| Image | Raw image array | 28x28x1 |
| ReLU Conv1 | Convolution layer with 9x9 kernels output 256 channels, stride 1, no padding with ReLU | 20x20x256 |
| Primary Capsules | Convolution capsule layer with 9x9 kernel output 32x6x6 8-D capsule, stride 2, no padding | 6x6x32x8 |
| DigiCaps | Capsule output computed from a WijWij (16x8 matrix) between uiui and vjvj (ii from 1 to 32x6x6 and jj from 1 to 10). | 10x16 |
| FC1 | Fully connected with ReLU | 512 |
| FC2 | Fully connected with ReLU | 1024 |
| Output image | Fully connected with sigmoid | 784 (28x28) |

*Fig16: Network design for CapsNet*

## Building a CapsNet

Diagram

Description automatically generated

*Fig17: Steps to build a CapsNet model*

## Dynamic Routing in CapsNet[11]

Step1 : initialize the primary logits  as 0

Step2: using SoftMax function, calculating the capsule coefficient as

Text

Description automatically generated

Step3: calculate the total capsule inputs

Text

Description automatically generated

Step4: squash to get a normalized vector output

Diagram

Description automatically generated with medium confidence

Step5: To make  more accurate , it is updated iteratively in multiple iterations (typically in 3 iterations)

A picture containing diagram

Description automatically generated

## Testing FMNIST dataset with CapsNet

After successful run of 100 epochs, the accuracy of the model is 93.2%

Loss incurred during training is like

Chart

Description automatically generated

*Fig18: Loss information from the CapsNet model*

And the accuracy obtained for the model is

Chart

Description automatically generated

*Fig19: Accuracy information from the CapsNet model*

## Discussion

There are fundamental representational reasons for believing that CapsNet is a better approach than the classical CNN models, which needs attention in various small insights to make it an industry standard. It is worth exploring capsules system as it gives unparalleled performance at segmenting overlapping. People who are working in computer vision, their hope will be to leverage on the successes and failures of CapsNets to build high performing, robust machine vision algorithms through further research in this field.

# Quantum Computing and image recognition

## Introduction

The image recognition models running in classical computers are having a major drawback – excessive processing time. Some CNN models on ImageNet dataset can run for days or even weeks. This slowness might give a good accuracy but in the era where business is changing in moments, doesn’t provide an optimal solution to business needs. With the advance of quantum computation, there might be some compromisation in accuracy, but the processing will be super-fast, which in most of the cases will be desirable to business. In general, quantum image processing has three steps -

##### Diagram Description automatically generated

*Fig20: Steps involved in quantum image processing*

If F and G are the input and the output images, in classical a M x L image can be represented as a matrix which is encoded by 2n bits, where n = 

In contrast the same image in quantum computer can be expressed with n qubits, which is performed by unitary evaluation  under a suitable Hamiltonian.

Diagram

Description automatically generated

*Fig21: Operations involved in quantum image processing*

Here I will try to summarize the structure of QIP, and introduce some QIP algorithms which can be explored in this category.

## Superposition and Entanglement in Quantum Theory

Superposition[19] states that if some state vectors f1, f2 and f3

Solves the linear equation for ψ, then ψ = c1f1 + c2f2 + c3f3would also be a solution for the same equation.

An electron can have two spins – up and down. The superposition state of the electron can be denoted as –

Diagram

Description automatically generated with low confidence

Where c1 and c2 denotes the probability amplitudes of finding the up spin or down spin electron when an observation is done.

A quantum computer which is having n qubits can exist in a superposition of 2n states from  ∣000...0⟩ to ∣111...1⟩

Entanglement[19] is a quantum correlation between quantum systems. When quantum particles become entangled, they form a global system such that the quantum state of individual subsystems cannot be described independently.

If we consider any two arbitrary quantum systems A and B, and their Hilbert spaces are HA and HB, then the Hilbert space of the composite system of A and B is given by the tensor product A picture containing text, furniture, table, seat

Description automatically generated

If the first system is in state A picture containing text

Description automatically generated and the second in state  , then the state of the composite system is

A picture containing text

Description automatically generated

Let’s consider a basis vector Shape

Description automatically generated with low confidence for HA and A picture containing schematic

Description automatically generated for HB. The general state in HA ⊗ HB will be in the form of

A picture containing text, watch

Description automatically generated

The state is said to be separable if there exist two vectors

A picture containing logo

Description automatically generated

And the state is inseparable if for any vectors Diagram

Description automatically generated with low confidence at least for one pair of coordinates Text

Description automatically generated we have A picture containing diagram

Description automatically generated And only when a state is inseparable, it is called an 'entangled state'.

## Qubits

A Qubit is often called as the basic unit of quantum information. It is a superposition of two quantum state – 0 and 1 and is represented by a wave function

A picture containing text

Description automatically generated

Here α and β are two complex numbers denoting the probability amplitude. And Text

Description automatically generated

Qubits are vectors in mathematical terms and sometimes we need to change the direction of these vectors to perform certain operations. They can be represented through Bloch spheres.Diagram

Description automatically generated

*Fig22: Bloch sphere to represent a qubit*

The cartesian coordinates of x, y and z can be written as

x = r sinθ cosϕ

y = r sinθ sinϕ

z = r cosθ

the operation on this single qubit is represented by the rotation of the sphere

A close-up of a calculator

Description automatically generated with low confidence

So when  = 0,  = |0> and when  = 1,  = |1>

## Representing an Image in Quantum Computer

There exists strategies to capture the necessary information of the colors and corresponding position of every pixel in the image and convert them into quantum state[13][16]. FRQI (Flexible representation for quantum images), MCQI (RGB Multi-channel representation for quantum images), NEQR (Novel enhanced quantum representation of digital images),INEQR(Improved NEQR), and GQIR (Generalized quantum image representation) are few among those strategies.

#### FRQI

This strategy maps each pixel’s grayscale value to the amplitude, and an auxiliary qubit to denote the spatial position of each pixel. Then, the whole image is represented as a large quantum superposition state.



Diagram

Description automatically generated with medium confidence

Where θi represents the intensity of the i th pixel.

FRQI can be prepared by Hadamard and Controlled rotation operators[14]. For an 2 n ×2 n image representation, the first step is to initialize the n qubits as  and as a next step, it is to apply the Hadamard gate to each qubit and elevate them in their superposition:

Diagram, schematic

Description automatically generated

In next step, the controlled rotation operators to be implemented on each basis state resulting in the joint state as follows:

A picture containing text, gauge

Description automatically generated

Each of the Ri rotation corresponds to the respective pixel’s rotation. Therefore, by rotating these quantum states, encoding of the intensities are done.

A simple example for a 2×2 image is given below, with corresponding θ angles (colour encoding) and associated kets (position encoding) :

Shape

Description automatically generated with low confidence

And the equivalent quantum state is

Text

Description automatically generated

#### NEQR

NEQR uses the basis state of a qubit sequence to store the grey-scale value of each pixel available in the image[16], instead of the probability amplitude of a qubit, as in FRQI. This model is designed to use *q* quantum bits for representing colour information. q is the image colour depth and it can represent all colours in the grayscale range [0,2𝑞−1]

An image compiled through NEQR can be expressed as

Text, schematic

Description automatically generated

∣𝑐𝑖〉 and ∣𝑖〉 represents the colour and position information of the image and can be written as

Text

Description automatically generated

#### 

#### GQIR

In GQIR, both the position and the colour information are captured into normalized quantum states[20]:|0〉and|1〉.It uses h=  qubits for Y-coordinate and w= qubits for X-coordinate to represent a H×W image. General representation of a GQIR image is

A picture containing text, watch

Description automatically generated

Where

 and



|YX〉is the location information and |CYX〉is the colour information. Coordinates h and w are determined by-

Text

Description automatically generated with medium confidence

## Quantum Image Processing

#### Quantum Image Compression

Image compression is necessary such that one can create, store, retrieve images by using less quantum resources[20]. A 216 image needs to 232 simple gates for reconstruction. Hence to minimize the number of gates i.e complexity by simplifying the boolean expressions, compression has to be carried out. An algorithm for compression could be like –

Text

Description automatically generated

*Fig23: Quantum image compression steps*

#### Geometric Transformation of Quantum Image

Geometric transformation[21] deals with spatial change in pixel position like image rotation, zoom in or zoom out.

Geometric transformation includes two-point swapping, flip, coordinate swapping, orthogonal rotations and their variants. This can be achieved by using NOT, CNOT, Toffoli gates. Circuit design for geometric transformation is like-

Chart, histogram

Description automatically generated

*Fig24: Circuit design for geometric transformation*

#### Colour Transformation of Quantum Image

Colour transformation refers to changing the image’s pixel value. To perform colour transformation, the quantum density-stratified algorithm is applied on quantum image GQIR over quantum colormap QCR. Based on the outcome, the quantum realization in the form of quantum circuits is prepared.

In 2003, Venegas-Andraca proposed that if an apparatus could detect electromagnetic frequencies and produce a quantum state as output, it could store colour in a qubit by translating given frequencies to quantum states. A full image could then be stored in a qubit lattice by updating the indices to specify pixels in the image.

A QHSL model[12] defined by chromaticity (further divided into hue and saturation) and lightness is proposed for colour representation. This model can be represented as

Text

Description automatically generated with medium confidence

where θ ∈ [0, π] indicates saturation, φ ∈ [0, 2π) indicates hue, and L 0 , L1 , · · · , Lq−1 ∈ [0, 2 q − 1], where Li ∈ {0, 1} denotes lightness.

A preparation circuit diagram for the QHSL model -

Table

Description automatically generated

*Fig25: Circuit design for colour transformation*

#### Quantum Image Scrambling

The popular traditional image scram-bling algorithms[17], i.e., Hilbert transform, Arnold trans-form, and Fibonacci transform, are extended to the quantum field to convert an image to its distort forms.

Suppose that |b7b6 . . . b0> stores a grayscale value , then we can design the below circuit[11] to change the grey scale value.

Diagram, engineering drawing

Description automatically generated

*Fig26: Circuit design for image scrambling*

Step 1 use quantum shuffle permutation to produce |b7>. . . |b1> |b0> → |b0> |b1>. . . |b7>.

Step 2 produce the scrambled output as



Where  and  denotes XOR

#### Quantum Image Segmentation

In the last two decades, a wide variety of segmentation techniques have been developed, which conventionally fall into the following two categories: layer-based and block-based segmentation methods. In both cases, thresholding methods have proven better result than other strategies. Thresholding methods can be differentiated between - parametric and nonparametric methods. Parametric approaches assume that the intensity distributions of images obey the Gaussian mixture

Nonparametric approaches try to find the threshold based on discriminating criteria such as the between-class variance, cluster distance, entropy, etc.[14]

After a set of thresholds are selected from past experience, image is divided into series of classes-

A picture containing text

Description automatically generated

where 𝑙∈[0,𝐿−1] represents the intensity level of image pixels, {𝑡ℎ𝑖=1,2,⋯,𝑀−1} is the set of thresholds and {𝐶𝑖=1,2,⋯,𝑀} are classes labelling different groups of pixels.

After which optimal thresholds are identified by maximising the below objective function

Text

Description automatically generated

Here, i and j are the index of the intensity classes, and 𝜔𝑖 and 𝜇𝑖 are the occurrence probability and the mean of a class, respectively. Such values are obtained as:

Text

Description automatically generated

where 𝑝𝑗denotes the probability distribution of pixels and 𝑞𝑗=𝑝𝑗/𝜔𝑖

#### Feature Extraction of Quantum Image

Feature extraction of an image is largely depends on the edge detection as edge detection can lead to image segmentation, image mosaic and target detection. In recent years, classic image edge extraction algorithms such as Prewitt, Kirsch, Sobel and Canny have been evaluated to find a suitable fix for the QIP.

The workflow of an image edge extraction is

Diagram

Description automatically generated

*Fig27: Steps involved in quantum feature extraction*

Compare to classical 3x3 convolutional operation, QSobel proposed a 5x5 convolution in 8 directions.

A picture containing calendar

Description automatically generated

*Fig28: Feature extraction from 8 different angles*

During the detection process, the grey pixels are weighted and averaged to provide continuous edge information. Taking the 0o direction, the algorithm calculate the gradient descent of the image brightness.

Text, letter

Description automatically generated

|XY> is the position vector of the pixel p.

S0 will create a pixel neighbourhood window as

Table

Description automatically generated

*Table7 : Pixel neighbourhood window for 0o direction*

Similarly, the gradient for all other direction is calculated and maximum value between the 8 is considered.



Si represents the grey value of pixel p.

Then the image gradient values are classified according to the threshold function Ut and a threshold T is set 

After filtering by threshold function, when TYX = 1,

The pixel point is considered as image edge.

## Measuring Quantum Image

Measuring a superposition state breaks the quantum nature of the system and falls back to classical one. Only the systems that are closed and are described by unitary time evolution by a Hamiltonian can be measured.

If two quantum states are entangled, measurement of one register affects or reveals the state of the other register by partially or entirely collapsing its state too. This effect can be used for computation, and is used in many algorithms[24].

There are quite a few theories regarding measurement. One of them is “weak measurement”, where the measuring system is weakly coupled with measuring device and doesn’t gives up quantum properties after measurement. There is also “non-destructive measurement”, where th integrity of the system and value of the measured observable is preserved.

Quantum measurements are described by a collection of operators {Mm} which hold below properties:

* Index m refers to the possible outcome of the measurement
* If the state of the system is  before the measurement, then the probability of m as outcome is given by p(m) = M+m Mmand the state of the system after measurement is given by Mm/Sqrt(M+m Mm)
* The measurement operators satisfy the completeness equation  m (M+m Mm) = I

The common reasons for the instability of the quantum system, which is called as “gate noise” are

#### The complementary problem

Where more noise gets introduced in the circuit while trying to minimize the existing noise. In case of FRQI, the image is represented by

A picture containing text

Description automatically generated

Where  encodes the color pixel and  represents the location in Bloch sphere. |I (θ)> has projections on more than one axis of the Bloch’s sphere. Consequently, the single procedure of measuring along a single axis of the Bloch’s sphere, for a technique that is expressed on more than one axis, introduces noise.

#### Decoherence

The decoherence theoryis reverting a quantum system back to classical through interactions with the environment (Radiation, light, sound, vibrations, heat, magnetic fields) which decay and eliminate quantum behaviour of particles. Coherence Lengthis denoted by the total time in which a qubit can maintain its quantum properties.

#### Quantum Error Correction and Quantum Gates

The set of allowable single qubit operations consists of unitary transformations corresponding to 2 × 2 complex matrices *U* such that *U*†*U =* 1. The corresponding action on a single qubit is represented in a circuit as[22]

A picture containing text, clock

Description automatically generated

The quantum logic gates, which are the heart of quantum processing, can be described as unitary matrices. A matrix, U, is unitary if and only if its inverse equals its conjugate transpose, i.e., if and only if U−1 = U†. If U is unitary the following facts hold:

• U† is unitary.

• U−1 is unitary.

• U−1 = U† (which is the criterion for determining unitarity).

• U†U = 1

• | det(U)| = 1.

Text

Description automatically generated

Some well-known quantum gates are given below in the table.

Table

Description automatically generated

*Table8 : Frequently used quantum gates*

Various strategies has been explored to minimise the impact of decoherence by increasing the coherence length. The aim of correcting error of a qubit system lies in detecting the error and counter defend it rather than disturbing the information flow[23]. The prominent operations used to handle the errors are

|  |  |
| --- | --- |
| Bit Flip | Prevents the qubit’s computation state flip from 0 to 1 and vice versa. |
| Sign Flip | Prevents the qubit’s computation state flip from + to - and vice versa. |
| Shor Code | combination of bit flip and sign flip codes |
| Bosonic Codes | QEC in a bosonic mode benefits from the infinite-dimensional Hilbert space of a harmonic oscillator for redundant information encoding and only one error syndrome that needs to be monitored |
| General Codes | In general, a quantum code for the [quantum channel](https://en.wikipedia.org/wiki/Quantum_channel) is given by subspace where  is the state in Hilbert space, such that there exists another quantum channel with  where  is the orthogonal projection onto .  is referred as the correction operation. |

*Table9 : Frequently used quantum operations*

## Circuit Diagram for Quantum Image Matching

The model described by Nan Jiang, Yijie Dang, Jian Wang [18] could find out a small image B with size 2m × 2m from a larger image A having size as 2n × 2n, where

Diagram, text, schematic

Description automatically generated

and n > m, k is position, and I is colour.

Additionally two extra auxiliary qubits f and g are needed 

The initial form of the algorithm is given by

Logo

Description automatically generated with low confidence

And the circuit diagram for the same is given by

Chart, diagram

Description automatically generated *Fig29: Circuit diagram for image matching*

Step 1 uses q CNOT gates to find out the matched area, i.e., the pixels that IA = IB.

Step 2 uses a transform U2 to find out the upper left corner of the matched area. U2 : | f = 0 → | f = 1, if |IA(kA) = |0 ⊗q and |kB = |0 ⊗2m, where U2 is a (q + 2m)-CNOT gate.

Step 3 changes the probability of subspace |kA> and increases the probability of |kA0 >, which corresponds to the upper left corner of the matched area based on Grover’s algorithm

Step 4 highlights the implementation of the projective measurements for measuring the state of the previous step, comparing the output in the upper left corner of the matched area.

# Discussion and future work

In 2019, Google claimed its 54-qubit quantum computer which could solve a calculation in minutes, that would take a classical machine 10,000 years. Though there is enormous potential for quantum processing, the same is very hard to achieve due to the occurrence of random errors.

Microsoft, IBM and Google have all created tools — [Q#](https://docs.microsoft.com/en-us/azure/quantum/overview-what-is-qsharp-and-qdk), [Qiskit](https://qiskit.org/textbook/what-is-quantum.html) and [Cirq](https://quantumai.google/cirq), respectively for the developers who can practise quantum coding on their platforms; but “The stage of quantum computers now is something like classical computing in the late 1980s” and needs lots of improvement till it can be used in large scale.

It is unlikely to happen that quantum computing will replace classical computers, instead, both will be embedded to resolve real-world problems in a much faster and more efficient way.

A recent study by Gartner informs that by 2023, 20% of organizations will be having separate budgeting for quantum computing projects.

There are huge potentials in the research area of quantum computing[15][16], as this is in a very nascent stage, and one can build a promising career in this field.

The main purpose of this section is to gather the current mainstream studies happening over various Quantum image representations, processing and measurement. All these efforts are very much needed for the materialisation of smooth, effective, faster and secure QIMP technologies, which will help developers to take full advantage of quantum computing and quantum information processing.

##### References

1. G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, “Improving neural networks by preventing coadaptation of feature detectors,”
2. I. Jolliffe, Principal Component Analysis. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011
3. M. D. Zeiler, D. Krishnan, G. W. Taylor, and R. Fergus, “Deconvolutional networks,”
4. Alex Krizhevsky, Ilya Sutskever, Geoffrey E. Hinton“ImageNet Classification with Deep Convolutional Neural Networks”
5. Karen Simonyan & Andrew Zisserman “Very Deep Convolutional Networks for Large-scale Image Recognition”
6. Ciresan, D., Meier, U., Schmidhuber, J.: Multi-column deep neural networks for image classification. In: Computer Vision and Pattern Recognition (CVPR), 2012
7. Christian Szegedy, [Vincent Vanhoucke](https://arxiv.org/search/cs?searchtype=author&query=Vanhoucke%2C+V), [Sergey Ioffe](https://arxiv.org/search/cs?searchtype=author&query=Ioffe%2C+S), [Jonathon Shlens](https://arxiv.org/search/cs?searchtype=author&query=Shlens%2C+J), [Zbigniew Wojna](https://arxiv.org/search/cs?searchtype=author&query=Wojna%2C+Z), “Rethinking the Inception Architecture for Computer Vision”
8. K. He, X. Zhang, and S. Ren, “Deep residual learning for image recognition”
9. G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, “Densely connected convolutional networks.”
10. A. Jaiswal, W. AbdAlmageed, Y. Wu, and P. Natarajan, “Capsulegan: Generative adversarial capsule network
11. S. Sabour, N. Frosst, and G. E. Hinton, “Dynamic routing between capsules,”
12. Fei Yan, Nianqiao Li, Kaoru Hirota, “QHSL: a quantum hue, saturation, and lightness colour model”
13. Fei Yan · Salvador E. Venegas-Andraca · Kaoru Hirota, “A Critical and Moving-Forward View on Quantum Image Processing
14. Aditya Dendukuri and Khoa Luu, “IMAGE PROCESSING IN QUANTUM COMPUTERS”
15. Yue Ruan , Xiling Xue , and Yuanxia Shen, “Quantum Image Processing: Opportunities and Challenges
16. Mario Mastriani , “Quantum Image Processing: the truth, the whole truth, and nothing but the truth about its problems on internal image representation and outcomes recovering”
17. Li, Chen, Song, Liao and Fang, “A Block-Based Quantum Image Scrambling for GNEQR
18. Nan Jiang, Yijie Dang, Jian Wang, “Quantum Image Matching”
19. Dong-Sheng Wang, “Superposition and Entanglement from Quantum Scope
20. Jian Wang, Nan Jiang, Na Zhao, “A Novel Quantum Image Compression Method Based on JPEG”
21. Ping Fan, Ri-Gui Zhou, Naihuan Jing, Hai-Sheng Li,” Geometric transformations of multidimensional colour images based on NASS”
22. Timothy Proctor, Kenneth Rudinger, Kevin Young, Erik Nielsen, Robin Blume-Kohout, “Measuring the Capabilities of Quantum Computers”
23. H. J. Briegel, D. E. Browne, W. Dür, R. Raussendorf, M. Van den Nest,” Measurement-based quantum computation”
24. Peter Morgan, “Classical and Quantum Measurement Theory”