WILLKOMMEN!

ADVANCING IMAGE CLASSIFICATION: INTEGRATING NAQSS ENCODING WITH HYBRID QUANTUM-CLASSICAL PQC MODELS VIA INTEL QUANTUM SDK

TECHNISCHE HOCHSCHULE DEGGENDORF

CONTENTS

TEAM

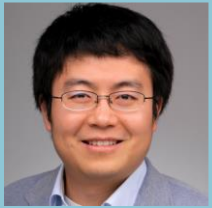
Digvijaysinh Ajarekar
Student, HPC-QC, Deggendorf Institute of Technology

2nd Rank Intel Quantum Challenge
Best Poster Presentation @FUT3CH Symposium 2024
Abstract Accepted @Quantum Matter 2024, San Sabastian, Spain

Prof. Dr. Helena Liebelt
Director
Institute of Future Technologies,
Deggendorf Institute of Technology

Suhaib Al-Rousan
Student, HPC-QC, Deggendorf Institute of Technology
Research Assistant, Quantum Applications
and Research Laboratory (QAR-Lab), LMU, Munich

2nd Rank Intel Quantum Challenge
Best Poster Presentation @FUT3CH Symposium 2024
Abstract Accepted @Quantum Matter 2024, San Sabastian, Spain



Prof. Dr. Rui Li
Co-Founder
Institute of Future Technologies,
Deggendorf Institute of Technology

Research

- Quantum Image Classification for Fake Art Identification
- Quantum Computational Fluid Dynamics with Intel Quantum SDK

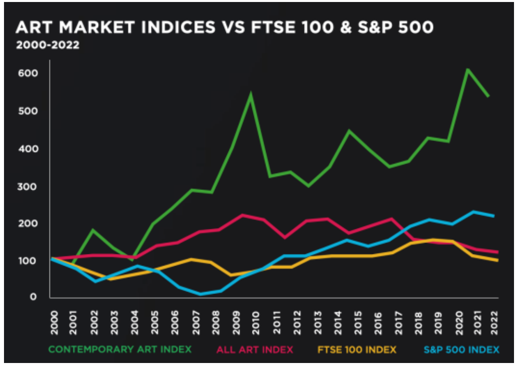
QUANTUM IN FAKE ART

The art world grapples with the issue of **art forgery**, where replicas are created and passed off as genuine works.
This can lead to significant **financial losses** and **erosion of trust** in the art market.

- Growing Luxury Asset
- Fake Art threat to Global Art Market

Traditional Methods,
- Require expertise
- Subjective
- Time Consuming

AI to the Rescue,
- Increased Accuracy
- Efficient



Model	No. of Layer	Parameters (Million)	Size
AlexNet	8	60	-
VGGNet-16	23	138	528 MB
VGGNet-19	26	143	549 MB
Inception-V1	27	7	-
Inception-V3	42	27	93 MB
ResNet-152	152	50	132 MB
ResNet-101	101	44	171 MB
InceptionResNetV2	572	55	215 MB
MobileNet-V1	28	4.2	16 MB
MobileNet-V2	28	3.37	14 MB
EfficientNet B0	-	5	-

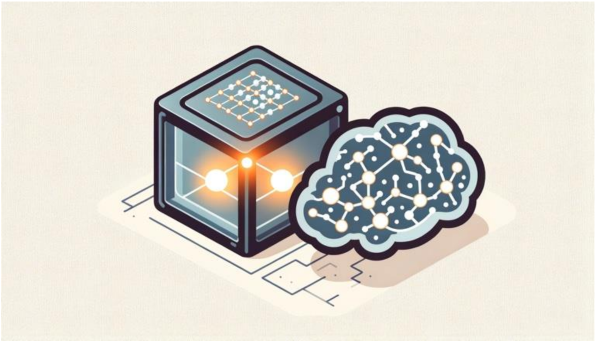
These concepts can be
Applied to Fake Currency Detection and other authenticity detection tasks.

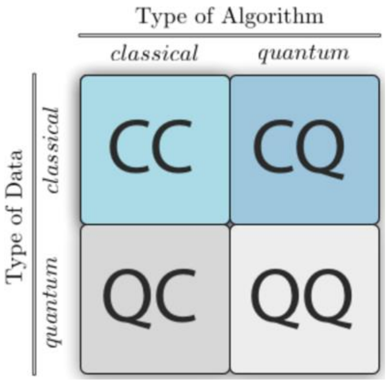
Quantum Computing provides exponential speed up with linear combination of all the possible basis states in superposition.

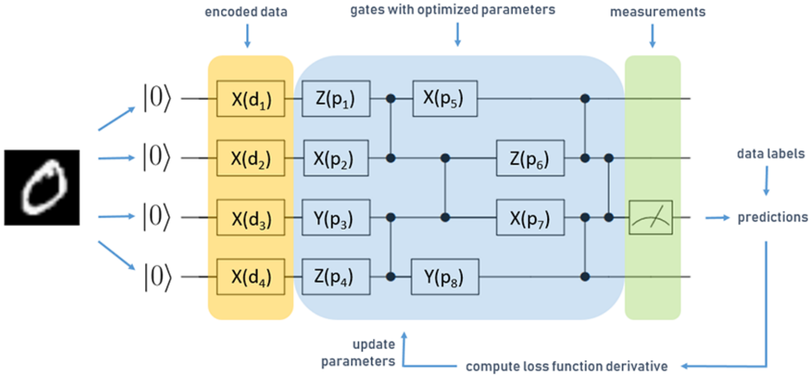
QML: MARRAIGE OF POWER HOUSES

Quantum machine learning represents the potent amalgamation of quantum computing and machine learning.

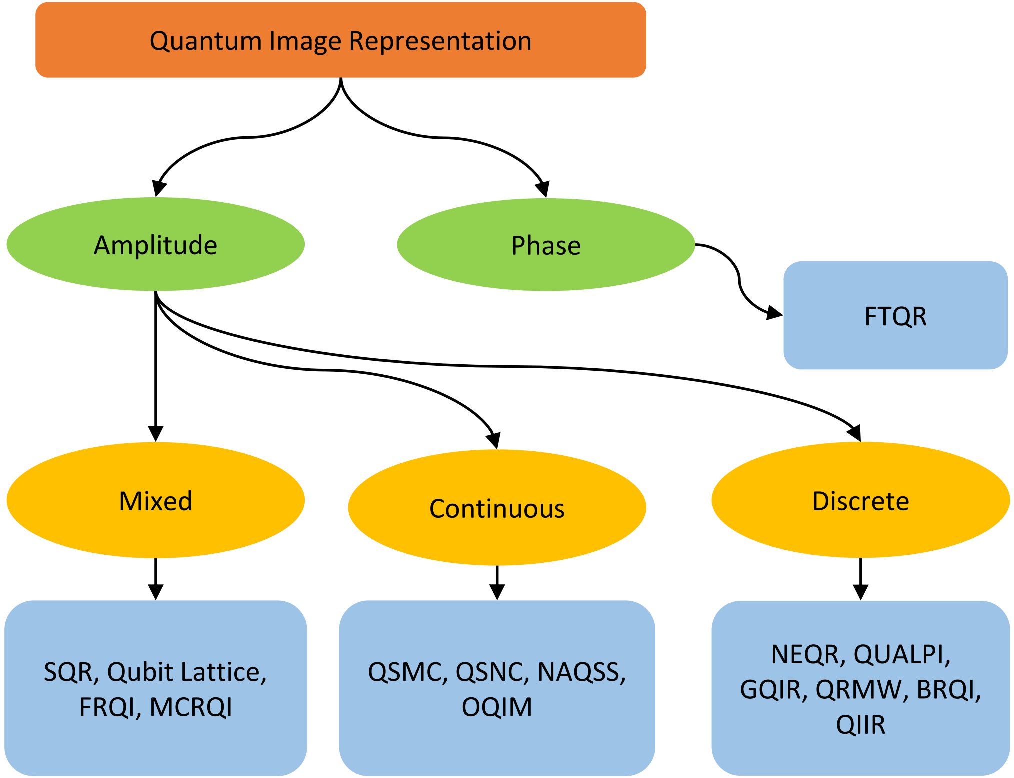
Quantum computers possess the ability to solve problems that are too intricate for classical computers, while machine learning algorithms can leverage the unique properties of quantum computers to achieve superior performance.





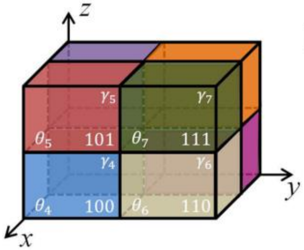


QUANTUM IMAGE REPRESENTION



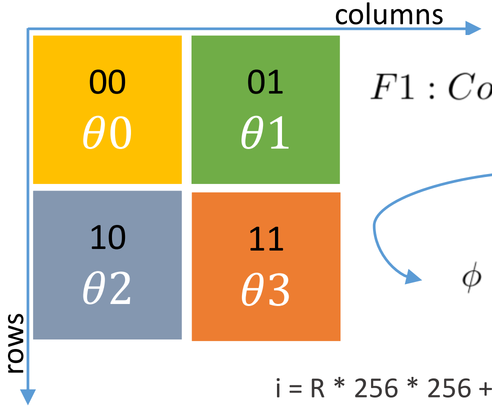
QIR	Required qubits	QIR	Required qubits
FRQI	$2n+1$	GNEQR	$n+8$ [G]/ $n+10$ [C]
NAQSS	$n+1$	NCQI	$2n+3q$
QSMC& QSNC	$4n$	BRQI	$n+3$ [G]/ $3n+9$ [C]
SQR	$2n$	QRCI	$2n+3$
QUALPI	$2n+q$	QRMMI	$2n+t+q$
NEQR		QRMW	$2n+b+q$
CQIR	$2n+m$	OCQR	$2n+4q$
MCQI	$2n+3$	FRQCI	$6n$
INEQR	$2n+8$ [G]/ $2n+10$ [C]	QMCR	$2n+3q$
IFRQI	$2n+p$	QBIR	$2n$
OQIM	$2n+2$	QIIR	$2n+2q+c$
DQRCI	$2n+9$	-	-

Note: "G" represents grayscale image, "C" represents color image.



$$
\begin{aligned}
|I\rangle = &\theta_0|000\rangle \otimes (\cos\gamma_0|0\rangle + \sin\gamma_0|1\rangle) \\
&+\theta_1|001\rangle \otimes (\cos\gamma_1|0\rangle + \sin\gamma_1|1\rangle) \\
&+\theta_2|010\rangle \otimes (\cos\gamma_2|0\rangle + \sin\gamma_2|1\rangle) \\
&+\theta_3|011\rangle \otimes (\cos\gamma_3|0\rangle + \sin\gamma_3|1\rangle) \\
&+\theta_4|100\rangle \otimes (\cos\gamma_4|0\rangle + \sin\gamma_4|1\rangle) \\
&+\theta_5|101\rangle \otimes (\cos\gamma_5|0\rangle + \sin\gamma_5|1\rangle) \\
&+\theta_6|110\rangle \otimes (\cos\gamma_6|0\rangle + \sin\gamma_6|1\rangle) \\
&+\theta_7|111\rangle \otimes (\cos\gamma_7|0\rangle + \sin\gamma_7|1\rangle)
\end{aligned}
$$

NAQSS



$$F1 : Color \longleftrightarrow \phi$$

$$\phi = \frac{\pi(i-1)}{2(M-1)}$$

i = R * 256 * 256 + G * 256 + B + 1

$$\theta_i = \frac{a_i}{\sqrt{\sum_{y=0}^{2^n-1} a_y^2}}$$

$$|\alpha_1 >= arctan\sqrt{\frac{\sum_{i=n}^{n} |\theta_{1i_{2..n}}|^2}{\sum_{i=n}^{n} |\theta_{0i_{2..n}}|^2}}$$

$$\alpha_{j,i_j,...,i_{j-1}} >= arctan\sqrt{\frac{\sum_{i=j+1}^{n} |\theta_{i_1...i_{j-1}1_{i_1...i_{j-1}}}|^2}{\sum_{i=j+1}^{n} |\theta_{i_1...i_{j-1}0_{i_1...i_{j-1}}}|^2}}$$

$$||\ \psi|| = \sqrt{\sum_{i=0}^{2^n-1} \theta_i^2 (cos^2 y_i + sin^2 y_i)} = 1$$

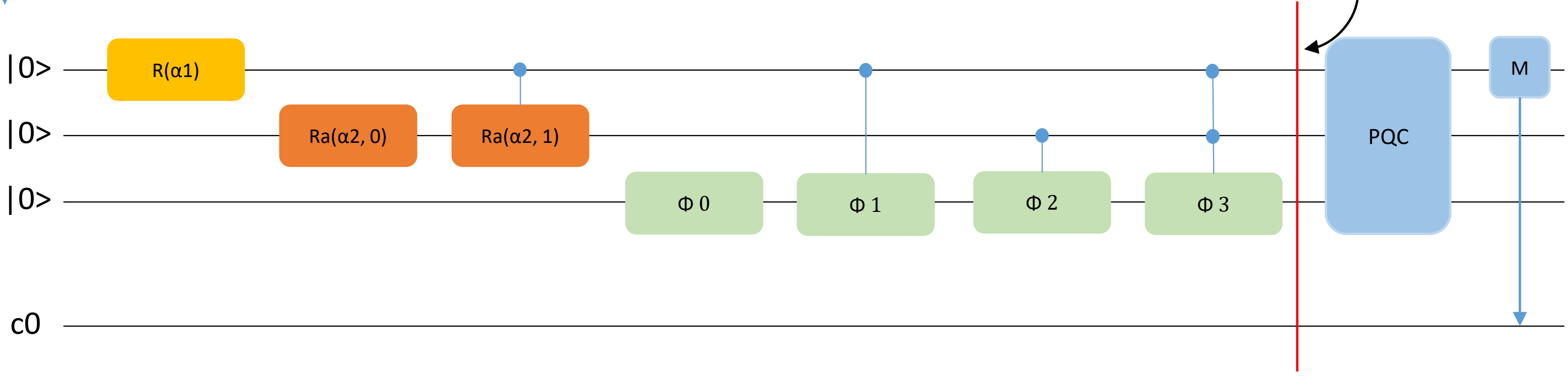
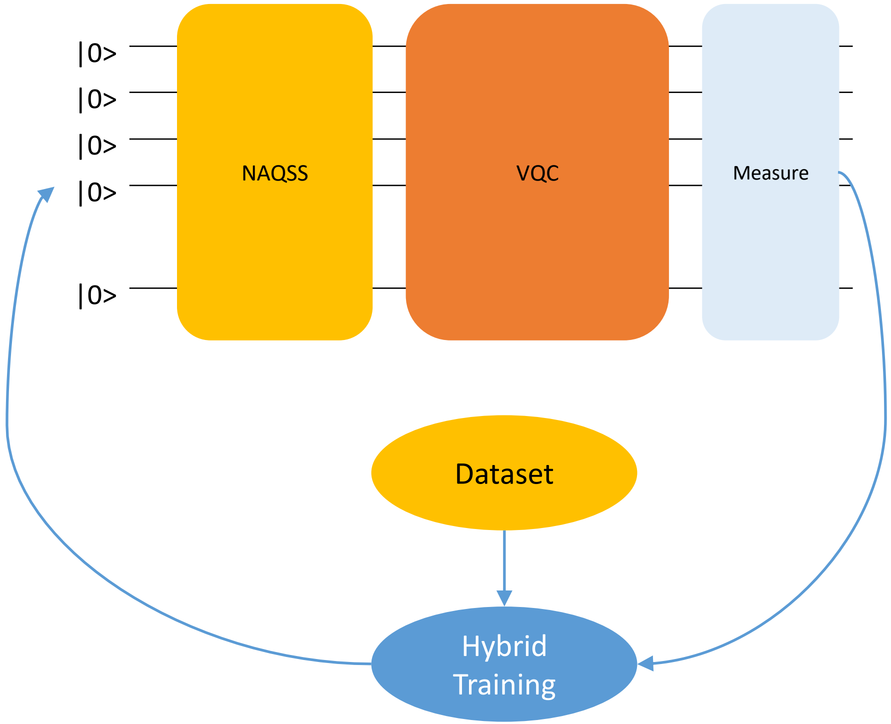


IMAGE CLASSIFICATION WITH NAQSS AND QNN

Algorithm

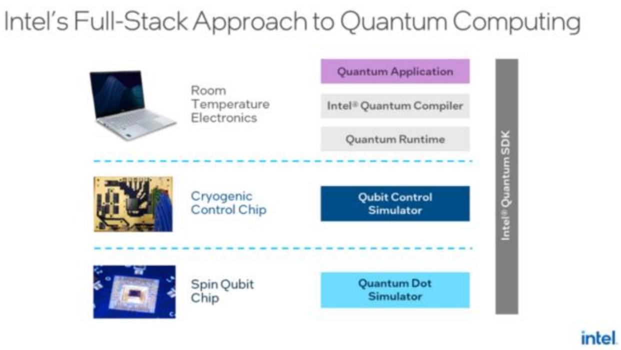
1. Data Preprocessing and transformations on the Images.
2. Encode Image into Quantum Circuit using NAQSS implementation.
3. Pass the state to QNN
4. Measure the Qubits
5. Get Loss and Gradients with Parameter Shift Rule
6. Update the Parameters
7. Repeat step 2 – 6 for all the images in dataset.



INTEL QUANTUM SDK

The Intel® Quantum SDK is a complete quantum computing stack in simulation. It includes:

- An intuitive user interface
- A compiler toolchain
- A quantum runtime environment optimized for running hybrid quantum-classical algorithms
- A choice of qubit simulation back ends



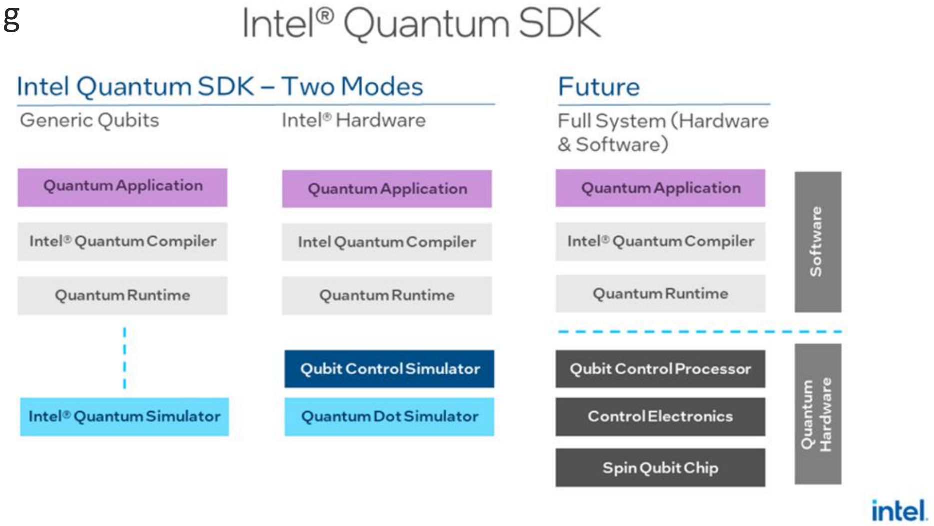
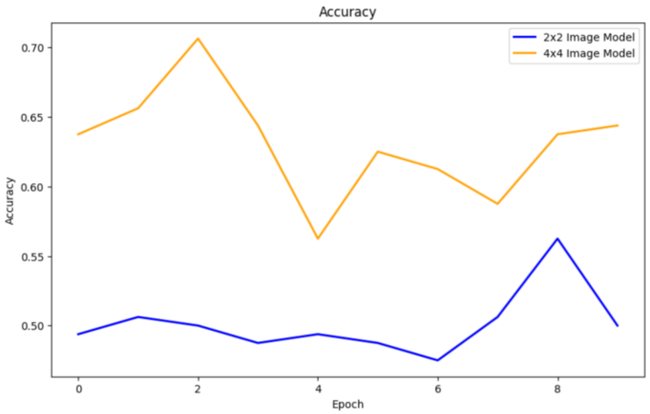
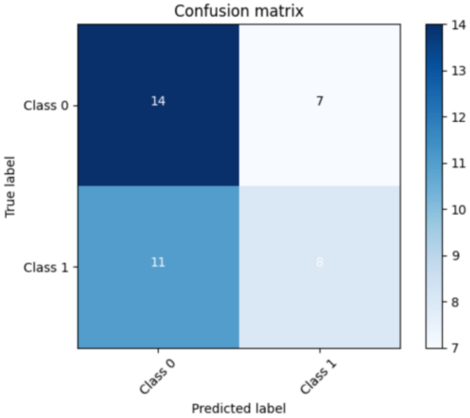


IMAGE CLASSIFICATION WITH NAQSS AND QNN

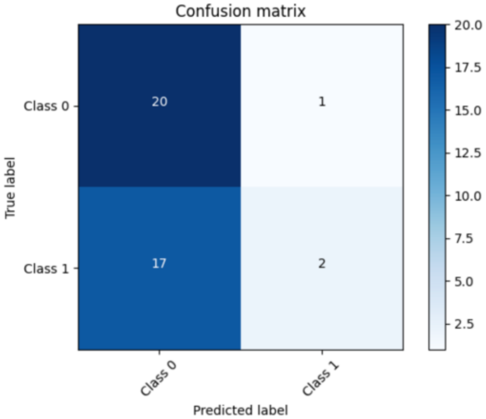
```
 Circuit for QBB in quantum_kernel - 'arbitaryState'

|qid_0> : - RX(alpha) ------- o ------------------------------------------- o ---------------------- o ---
                              |                                             |                        |
|qid_1> : - RX(alpha[1]) ---- Phase(alpha[2]) ---- o ---------------------- | ---------------------- o ---
                                                   |                        |                        |
|qid_2> : - RX(alpha[3]) ---- RX(alpha[4]) ------- X ---- RX(alpha[5]) ---- X ---- RX(alpha[6]) ---- X ---


 Circuit for QBB in quantum_kernel - 'qnn'

|qid_0> : - RX(weights[0]) ---- RY(weights[0]) ---- o ----------- X ---
                                                    |             |
|qid_1> : - RX(weights[1]) ---- RY(weights[1]) ---- X ---- o ---- | ---
                                                           |      |
|qid_2> : - RX(weights[2]) ---- RY(weights[2]) ----------- X ---- o ---


 Circuit for QBB in quantum_kernel - 'prepare'

|QReg_0> : - PrepZ ---- PrepZ ---- PrepZ ---

|QReg_1> : - PrepZ ---- PrepZ ---- PrepZ ---

|QReg_2> : - PrepZ ---- PrepZ ---- PrepZ ---


 Circuit for QBB in quantum_kernel - 'measure'

|QReg_0> : - MeasZ(CReg[0]) ---
```

RESULTS





2x2 Image



4x4 Image

Model	Image Size	Epochs	Parameters	Accuracy
Quantum	2x2	10	90	51%
Quantum	4x4	10	70	64.5%

VIELEN DANK

CONTACT:

digvijaysinh.ajarerekar@stud.th-deg.de

suhaib.al-rousan@stud.th-deg.de

helena.liebelt@th-deg.de

rui.li@th-deg.de