Quantum Information Processing (2023) 22:179
https://doi.org/10.1007/s11128-023-03930-5

QSurfNet: a hybrid quantum convolutional neural network for surface defect recognition

Shraddha Mishra[1] · Chi-Yi Tsai[1]

Received: 16 June 2022 / Accepted: 3 April 2023 / Published online: 27 April 2023

Abstract
In this paper, we propose a novel hybrid quantum–classical convolutional neural network named QSurfNet, inspired by an efficient surface defect recognition model called SurfNetv2. SurfNetv2 is an established high-speed classical convolution neural network (CNN) model for image recognition, and QSurfNet further inherits the legacy by introducing quantum CNN (QCNN) layers, reducing the number of convolution blocks in the model architecture and the image size required for recognition. The QSurfNet architecture consists of a QCNN module, a feature extraction module, and a surface defect recognition module. The algorithm works on end-to-end supervised quantum machine learning and deep learning techniques to classify the surface defect categories of the surface defect image datasets. For this research, we used the 8×8-pixel and 12×12-pixel resolution RGB image information from the public Northeastern University dataset, and an industry-sourced calcium silicate board private dataset. We used principal component analysis for image dimensionality reduction across the R, G, and B channels, individually. We compare the performance of QSurfNet with six state-of-the-art methods on these datasets upon recognition results on test accuracy, recall, precision, and $F1$-Measure performance metrics. QSurfNet is novel in terms of the algorithm design methodology that can turn any classical CNN algorithm into state-of-the-art QCNN. Hence, the proposed methodology contributes to the practical feasibility of developing novel convolutional architecture designs of hybrid quantum–classical algorithms.

Keywords Quantum convolutional neural network · Quantum machine learning · Surface defect recognition · Parameterized quantum circuit · Industry 4.0

✉ Chi-Yi Tsai
chiyi_tsai@gms.tku.edu.tw

1 Department of Electrical and Computer Engineering, Tamkang University, 151 Yingzhuan Road, Tamshui District, New Taipei City 251, Taiwan R.O.C.

1 Introduction

Advances in computing technology are based on the keystone objective of matching and ultimately surpassing the human brain's capability to perform complex logical tasks. Neurologically, human brains are pattern-mapping machines, and we are social creatures and pattern-recognition machines [1]. Precisely this is why replicating how humans process visual information through computers has been a crucial problem to be solved. After the successful contribution of artificial intelligence, machine learning, and deep learning [2] to computer vision, scientists are now simulating how nature processes visual information by incorporating the laws of physics [3] with computer vision. This new paradigm is quantum information processing.

Our research explores quantum image processing for multi-class multichannel image classification. Processing machine vision through quantum computing is logical, because light propagates in the "quanta-of-electromagnetic-energy" of photons, which has dual nature as a propagating wave of electric and magnetic fields [4], and hence, is a quantum phenomenon [5].

Computer vision in industrial automation contributes to the overall enhancement of efficiency, reduction of errors, and labour costs. Surface defect recognition is a very important computer vision automation application in the manufacturing industry. The development of high-performance defect recognition algorithms can not only accelerate the upgrading of automatic inspection technology, but also improve the production efficiency of related industrial applications [6–8].

In this paper, we propose a novel surface defect recognition algorithm based on a hybrid quantum–classical convolution neural network (QCNN), where parameterized quantum circuit (PQC) is designed as a 2-dimensional quantum convolutional neural network (2-D QCNN) layer. The parameterized quantum circuit design is based on the authors' previous work [9], using TensorFlow Quantum [10], and convolutional neural network (CNN) architecture is inspired by SurfNetv2 [11]. The main contributions of this paper are as follows:

1. We propose a new method of hybrid quantum–classical algorithm architecture design that allows the implementation of 2-D QCNN layer design integration with any state-of-the-art algorithm, while maintaining the customizability and the modularity of the algorithm, thereby enhancing its performance.
2. The proposed QSurfNet model provides improved image recognition performance over Surfnetv2 [11].
3. The proposed QSurfNet model is the first hybrid quantum–classical algorithm using QCNN for surface defect recognition applications.
4. Our proposed algorithm supports the computer algorithm adage "size doesn't matter" and uses only two blocks of SurfNetv2 [11], delivering state-of-the-art results as described in Sect. 4.2. Hence, QSurfNet is a high-performance industrial-grade computer vision application.

The QSurfNet is trained, validated, tested, and benchmarked against SurfNetv2 [11], SurfNet [12], ResNet18 [13], DenseNet [14], VGG16 [15], and MobileNetv2 [16]. The experimental results demonstrate that QSurfNet outperformed the latter six state-of-the-art algorithms on both our industry-sourced private calcium silicate

(CSB) dataset and the public Northeastern University (NEU) surface defect datasets for the very small image resolutions of $8 \times 8 \times 3$ and $12 \times 12 \times 3$. In terms of applications, QSurfNet serves the industrial use case as an efficient, high-performance, and lightweight hybrid quantum–classical CNN algorithm.

The above brief introduction is a broad and brief glimpse into the details of the rest of this paper, which are arranged as follows. Section 2 discusses the literature review on the important past developments briefly in surface defect recognition, and in detail on quantum image classification leading to the present advancements. Section 3 elaborately describes the underlying principles of quantum machine learning (QML) in Sect. 3.1, the details of QCNN layer development in Sect. 3.2 and Sect. 3.3 and the proposed QSurfNet model in Sect. 3.4. Section 4 discusses the experiments and performance evaluation of the proposed algorithm. Finally, Sect. 5 summarizes the crucial findings of the research and concludes with suggestions for possible future directions.

2 Literature review

2.1 Surface defect recognition

In the Industry 4.0 Era, success of any manufacturing industry and company is dependent on the quality and flawlessness of their mass-produced end-products where defect detection of the end-products is automated through robotics and machine vision algorithms. In industrial real-time surface defect recognition setups, defect recognition systems are deployed comprising of the high-speed image-capture hardware setup, and a high-accuracy defect detection module working on specific image recognition algorithms.

There are four categories of surface defect recognition methods: conversion to the frequency domain, colour space conversion, machine learning and deep learning-based approach, and hybrid quantum-classical method as shown in Fig. 1.

Conversion of images to the frequency domain aids in the detection of hard-to-spot defects. When Fourier transform (FT) pre-processing is applied to the spatial image,

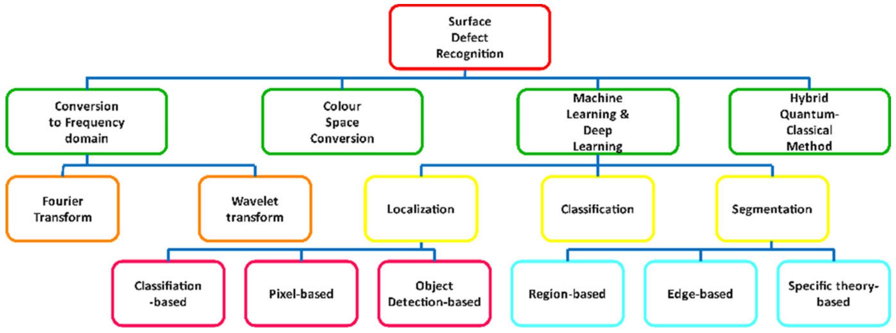


Fig. 1 Surface Defect Recognition Methodologies

the image data gets converted into the frequency domain, and the defect detection module detects the changes in the image spectrum [17]. Frequency domain filtering aids defect detection capability and after CNN feature extraction, inverse transform is applied to the image to get the processed result.

In the steel manufacturing industry, implementation of FT-based defect detection algorithms is a common application for the defect detection of hot-rolled strips. Reference [18] first proposed the surface defect detection of the rolling strips by processing input features by fast Fourier transform (FFT) followed by the Learning Vector Quantization (LVQ). Similarly, [19, 20] discusses the FFT-based defect detection method for small magnet rings. FFT-based Surface defect detection can also be applied on soft surfaces of textiles and fabrics [21–23]. FFT-based surface defect recognition methods yield high-performance detection results for even the smallest of the defects, with the trade-off of high operation costs. Hence, its alternates are more popular.

Wavelet Transform is another powerful tool to detect surface defects in the frequency domain [24, 25]. It performs the localization analysis of the spatial image's time or space frequency, and gradually refines the signal by translation and scaling. It is mostly implemented where images require noise reduction [26] and need to overcome the discontinuity of the hard threshold function [27]. This method is deployed by the metal manufacturing and metal fabrication industry for surface defect recognition for cracks, rust, etc. [26, 28, 29].

Some surface defect detection methods perform colour space conversion of the spatial image to aid feature extraction. This category works best for the applications where colour serves as one of the information features and converting to some other colour spaces than grayscale will better enhance the defect detection. Certain industries encounter surface defects in different irregular forms. Reference [30] proposes to distinguish the defect-detected regions of a food image by dividing the image into many small pieces and clustering the defective and flawless parts in segregation, followed by HSV colour space conversion that detects the defective regions of the image.

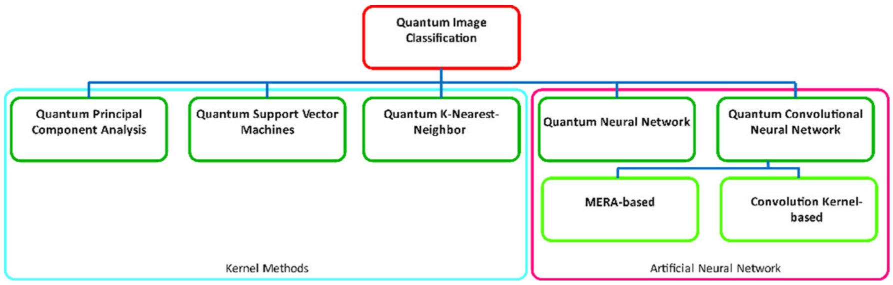
Machine learning and deep learning-based defect recognition algorithms are the most widely adopted by the industry due to their fast and efficient performance. Nowadays, these methods deliver state-of-the-art detection results, including classification, segmentation, and localization strategies for defect detection. Defect classification methods detect the type of defect by inspecting the texture, shape, and size features of the image data. These classification algorithms deploy deep CNN for defect feature extraction. For instance, existing SurfNetv2 [11] classifies the CSB dataset images into crash, dirty, uneven, and normal and NEU dataset images into rolled-in scale, patches, crazing, pitted surface, inclusion, and scratches categories. Reference [31] performs iron welding defect classification based on CNN and Gaussian kernel, and classifies the subject based on good quality welds, over splatter, porosity and undercut. Localization algorithms detect the position of the defect and indicate it with the bounding box, classify its defect type, identify which pixels carry the defect, and even differentiate different defects from each other as object detection. Reference [32] proposes a DCNN classification architecture that can automatically localize defect regions in reflectometric recordings on shiny surfaces. Reference [33] proposes to first localize the object in the image, and then classify its defect through CNN-based algorithm. Reference [34] performs pixel-perfect defect recognition that localizes the defect, passes the input to CrackNet and then classifies the pavement crack defects of the picture. Reference [35] proposes Faster R-CNN [36]-based multiscale defect-detection network to locate aluminium profile surface defects. Segmentation-based defect recognition indicates the extent and severity of the defects and damages. For efficiency, region-based defect recognition algorithms implement threshold segmentation [37, 38], regional growth method [39], clustering [40, 41], and splitting and merging [42] in their algorithms. Edge-based segmentation methods [43–45] are used for applications where defect position is not that important, but speed and memory efficiency is the focus. Some segmentation methods are based on a specific theory, such as wavelet transform-based algorithms [46, 47].

Hybrid quantum-classical methods [10, 48–50] have proven to be widely successful in image recognition applications. Surface defect classification is one of the industrial implementations of quantum image processing algorithms. In the Quantum Computing Challenge 2021 held by BMW and AWS, the use-case challenges were the surface defect classification [51] and automated quality assessment of concrete surface against the presence of cracks [52] based on the quantum neural network (QNN) implementation of quantum kernel alignment (QKA) [53] algorithm. However, not much research and development are available on the QCNN implementation in the surface defect recognition application.

2.2 Quantum image classification

In the noisy intermediate-scale quantum (NISQ) era [54] quantum computing paradigm, a major part of information processing is delegated to classical computing. Hence, quantum computing techniques are currently aiding classical machine learning and deep learning algorithms as subroutines for improving information processing performance [55].

Classification is one of the generalization targets of machine learning and deep learning methods. Quantum computing methods process information in superposition through entanglements. It carries sampling and probabilistic computational similarities with the machine learning methods, hence QML can also be implemented on classical image processing techniques [2, 56, 57]. In this section, we briefly introduce works done based on the quantum image classification methodologies as shown in Fig. 2.



Quantum Image Classification
Quantum Principal Component Analysis
Quantum Support Vector Machines
Quantum K-Nearest-Neighbor
Quantum Neural Network
Quantum Convolutional Neural Network
MERA-based
Convolution Kernel-based
Kernel Methods
Artificial Neural Network

Fig. 2 Quantum Image Classification Methodologies

Reference [58] proposes the algorithm for grayscale image classification based on classical principal component analysis (PCA) [59] followed by quantum measurement. PCA divides the image signal space into two orthogonal subspaces, and the leading principal components are projected onto the subspace of quantum states. Quantum classification is performed after the measurement of quantum states that represent image data as encoded input computational basis states. However, classical PCA is not computationally and cost-effectively efficient when the problem contains a large number of high-dimensional variable parameters [60]. Lloyd et al. proposed quantum principal component analysis (qPCA) [61], which can analyse an unknown low-rank density matrix by extracting its eigenvectors with the largest eigenvalues in a short time, giving exponential speedup over existing algorithms. However, qPCA has the computational requirements of a large number of ancillary register qubits for phase estimation algorithm (PEA) [62] analysis and error-free quantum operations, hence qPCA is applicable in future quantum hardware. For NISQ-era hardware, resonance-based quantum PCA (RqPCA) [63] is feasible which requires only one ancillary qubit and also executes simpler quantum algorithms.

Rebentrost et al. proposed a quantum support vector machine (qSVM) algorithm [64] that can be implemented in $O(\log GH)$ run time with both training and classification stages, in comparison to $O(\log(\epsilon^{-1})\text{poly}(G,H))$ run time taken by classical SVM, where G is the dimension of the feature space, H is the number of training vectors, and ϵ is the accuracy. Delilbasic et al. compare the computation performance between circuit-based qSVM and quantum annealing-based qSVM for remote sensing multispectral image data [65] and they observe that quantum annealing-based qSVM is feasible on current NISQ hardware, and the circuit-based qSVM is more suitable for the future hardware capabilities. The limitation of circuit-based qSVM for the classification application in NISQ devices due to computational resource requirements is also confirmed by [66–68]. However, these researches also establish the superiority of qSVM over classical SVM in the classification results.

K-nearest neighbour (KNN) is a simple and high-accuracy performance algorithm [69]. The quantum K-nearest neighbour (qKNN) method is used to classify data output from the quantum simulator that cannot be efficiently simulated classically by averaging the vectors in set A and B and computing a representative vector known as centroid [70] based on computing Euclidean distance. The qKNN offers significant query complexity reduction in comparison to its classical counterpart. However, qKNN based on Hamming distance gives time complexity $O(\sqrt{T}\log_2 T)$ run time which is quadratic speedup in comparison to that based on the Euclidean distance, where T is the number of times the computation of distance must be performed [71].

Zak et al. were the first to propose the theory of linear reversible QNN implemented in the role of nonlinear irreversible neural networks as a tandem of hybrid quantum and classical computation [72]. Schuld et al. proposed the QML model [73] where classical McCulloch–Pitts neuron [74] is replaced by a "quron" embedding the values $|x\rangle$ in the 2-D Hilbert space with bases $\{|0\rangle, |1\rangle\}$ [75]. This "quron" is to be designed as a qubit embedding in the unitary variational quantum circuit or PQC, where the correlations of the data are captured by the entanglement connections of quantum circuits [76, 77]. The features of high-dimensional image data are pre-processed with the state-of-the-art classical network before being passed on to the quantum neural network to be processed for recognition and classification [78]. Circuit-based image classifiers have been highly successful in binary and multi-class image recognition and classification, implemented as hyperspectral image classifier [79], medical image classifier [80], etc. For more efficient classification results, image dimensionality reduction is performed with PCA or convolutional autoencoder (CAE) before implementing QNN [81], and feature extraction is performed through transfer learning[9, 78, 82–87].

Cong et al. proposed a novel NISQ device-compatible QCNN architecture similar to the multiscale entanglement renormalization ansatz (MERA) architecture, involving alternate quantum convolution and pooling quantum layers for quantum phase recognition requiring only $O(\log(n))$ variational parameters for input sizes of n qubits for recognising 1D phases [88]. For binary image classification, Reference [89] introduces a ternary unitary circuit feature extractor based on MERA encoding as QCNN. Reference [90] proposes a resource and depth-efficient QCNN feature-extracting model for large input image data using the quantum random access memory (QRAM) algorithm [90] for image classification. Reference [91] proposes a hybrid quantum graph convolutional neural network (QGCNN), and Ref [92] proposes a kernel-based QCNN algorithm for processing voluminous and sparse data High Energy Physics datasets.

Recently, new QCNN architectures similar to CNN architectures were designed, where the quantum kernel convolves over an image to extract image features, and the variational quantum circuit learns the input features, updates the learning parameters during training and executes measurement before passing the information to the subsequent quantum pooling layer and the tandem classical architecture for image recognition and classification. For instance, Ref [93] proposes QCNN consisting of quantum convolutional layer, quantum pooling layer and fully connected quantum layer for MNIST dataset image classification while also implementing spatial filtering, image smoothing, sharpening and edge detection. Hur et al. in [94] proposed a novel methodology of QCNN in which they construct and benchmark quantum convolution neural networks from parameterized quantum circuits by using only two-qubit interactions. They identify best-performing PQCs and encoding methods for QCNN with shallow-depth unitary circuit designs with a small number of quantum gates while establishing the better performance of QCNN than its CNN counterpart. Apart from image recognition, QCNN has also been implemented in speech recognition and classification via quantum convolutional filter [95].

3 The proposed method

In this section, we describe the QCNN component and the QSurfNet architecture in detail.

3.1 Parameterized quantum circuit

A gate-centric quantum computer is controlled by operating the gate commands through programming on the qubits isolated from the universe. This isolated physical system in physics is represented by the Hilbert space $\mathcal{H}$, where the vector state

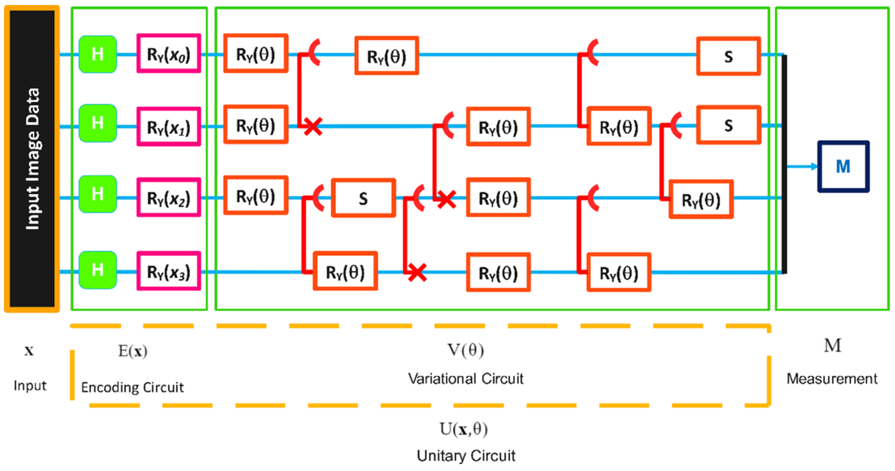


Fig. 3 Parameterized Quantum Circuit (PQC) 1

components are represented with a complex vector space with an inner product, which serves as a feature space for the quantum data [96]. For NISQ-era quantum computation, such a system can be built with a few qubits forming a quantum circuit. A quantum circuit is called parameterized if its gates consist of undetermined trainable parameter expressions. The PQC is the building block of circuit-based QML algorithms. The PQC design comprises the data encoding, information processing and measurement part of the unitary circuit. We developed specialized PQCs for image classification in [9], from which we choose PQC 1 as shown in Fig. 3 for the development of QCNN.

The PQCs we have implemented are better suited for NISQ-era applications and consist only of Hadamard gates (H), controlled-NOT gates (CNOT), rotational Y gates $R(\theta)$, controlled rotational Y gates $CR_y(\theta)$, and S gate, which is the squared root of Z gate. The Y-gate is represented as a combination of X and Z gates as represented in (1). We chose (θ) gates in the development of our PQCs for quantum image processing because $R_y(\theta)$ gate rotates the parameter θ radians anticlockwise about the $\widehat{y}$ axis of the Bloch sphere without introducing complex amplitudes. These variational parameters θ update during the QCNN model training and their values optimize as per the overall effect of the hyperparameter optimization. At a time, each qubit holds 1 pixel, and every individual pixel value is encoded by a quantum gate.

$$Y = -iZX \tag{1}$$

The PQC updates its parameters by optimizing the expectation value of the cost function M also known as observable as explained in (2) during the algorithm training process [10].

$$f(x,\theta) = \left\langle 0 \middle| U^{\dagger}(\boldsymbol{x},\theta) M U(\boldsymbol{x},\theta) \middle| 0 \right\rangle \tag{2}$$

The input data x is encoded as the initial state $|0\rangle^{\oplus n}$ in an n-qubit quantum circuit. The input to the PQC is initiated at $|0\rangle$ by the Hadamard gates because preparing the input state as a superposition state allows the encoding of multiple inputs as a single initial state. Consider a unitary circuit $U(\mathbf{x}, \theta)$, that implements unitary operation and is the function of the data input $\mathbf{x}$ and a set of parameters θ that act as weights in the QNN. It is a system of qubits connected in a specific sequence, where qubits individually hold a one-directional sequence of quantum gates. These quantum gates apply unitary operations on the qubits to process input information $\mathbf{x}$ by transforming the quantum state of the system [97]. This arrangement of gate operations upon the qubits is known as a circuit in analogy to the classical analogue circuits where the information passes unidirectionally through the electric wires in all the interconnections and analogue components only once. Similar to analogue circuits, the quantum circuit is composed of a collection of quantum wires, each connected to a qubit. The information in the quantum circuit passes through each component and inter-qubit connections unidirectionally, by convention graphically represented from left to right, changing the values of the parameters of the quantum components.

The unitary circuit $U(\mathbf{x}, \theta)$ comprises of the encoding circuit $E(\mathbf{x})$ and variational circuit $V(\theta)$. $E(\mathbf{x})$ is a nonparametric unitary quantum circuit responsible for encoding input pixels represented by the image data $\mathcal{X} = [\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \ldots, \mathbf{x}_N] \in \Re^{n\times N}$ and transforming the classical input data of domain $\mathcal{X}$ into the quantum input for the quantum circuit parameters θ. The variational circuit $V(\theta)$ is a variational unitary quantum circuit comprising learnable variational parameters responsible for learning of the parameterized weights. Since the input is given to PQCs through the encoding circuit first as $E(\mathbf{x})$, then the quantum data is passed to the parameters θ of $V(\theta)$, thus $U(\mathbf{x}, \theta)$ is mathematically represented by Eq. (3), as shown in Fig. 3 and explained below [78, 83, 98, 99].

$$U((\mathbf{x}, \theta) = V(\theta)E((\mathbf{x}) \tag{3}$$

Any classical data that undergoes underlying quantum mechanical processing, either resulting from quantum circuit processing measurements or the data generated by a quantum device and then fed into the next algorithm as input, is to be considered quantum data. Hence, the classical input data $\mathcal{X}$ after being processed by encoding circuit $E(\mathbf{x})$ becomes quantum data [10].

A. Encoding Circuit

The encoding circuit maps the data input $\mathbf{x} \in \Re^n$ into 2^n-dimensional Hilbert space $\mathcal{H}$ created by quantum circuit $E(\mathbf{x})$ of n qubits with the help of quantum encoding function $\phi_Q(\mathbf{x})$ upon the unitary circuit $U(\mathbf{x}, \theta)$ as represented by (4).

$$E : \mathbf{x} \mapsto U_{\phi_{\mathrm{Q}}}(\mathbf{x})|0\rangle^{\oplus n} \tag{4}$$

Each ith input from $\mathcal{X}$, which is fed inside the quantum circuit $E(\mathbf{x})$, is first processed by the Hadamard gates H into its superposition state $|\psi_i\rangle$ and then processed by the parameters of unitary gates hence implementing the quantum feature map ϕ_{Q}: $\mathcal{X} \mapsto \mathcal{H}$ where x_i is the ith element of the data input $\mathbf{x}$ having angular value $x_i \in [0, \pi]$

and θ being the angular rotation value of $R_y(\theta)$ as explained by (5). The encoded information processed by U_{ϕ_Q} becomes the quantum data $\left|\phi_Q(\mathbf{x})\right\rangle$ and exists as the Kronecker product of superposition states of all the inputs as explained by (6) [10, 94, 98, 100]. The quantum feature Hilbert space $\mathcal{H}$ is hence defined as the space of the complex vectors $\left|\phi_Q(\mathbf{x})\right\rangle$ where ϕ_Q is an encoding function that transforms the elements of $\mathcal{X}$ into the circuit parameters and $\mathfrak{R}^n$ denotes input value x from domain of real numbers $\mathfrak{R}$ of dimension n [101].

$$|\psi_i\rangle = \left(\cos\left(\frac{x_i}{2}\right)|0\rangle + \sin\left(\frac{x_i}{2}\right)|1\rangle\right) \tag{5}$$

$$U_{\phi_Q} : \mathbf{x} \in \mathfrak{R}^n \mapsto \left|\phi_Q(\mathbf{x})\right\rangle = \otimes_{i=1}^{n}|\psi_i\rangle \tag{6}$$

B. Variational Circuit

The variational circuit $V(\theta)$ defines the learnable structure of the PQC consisting of the free parameters θ which act as weights. In the gate-centric architecture, variational circuit is composed of a certain circuit depth (explained in i) of the rotational and entangled gates constituting the main circuit block with the qubits entangled as per entanglement strategy (explained in v) followed by the measurement of the processed values.

Since our PQC [9] is a gate-centric architecture [76], it consists of the following constructional parameters:

i. Circuit Depth

The number of times the primary circuit block is repeated or the total number of the quantum gates in the single layer $\updownarrow_Q$ of the quantum unitary circuit determines the circuit depth. For NISQ-era quantum algorithm development, we implement shallow-depth quantum circuits with small number of qubits to manage the qubit noise error [76].

ii. Layer

An n-qubit quantum unitary circuit $U(\mathbf{x}, \theta)$ may also be a multilayer unitary circuit. By controlling the number of layers and the quantum gate operations implemented within them, the information processing is controlled.

iii. Number of Qubits

Each qubit in the quantum circuit holds the image data feature. Hence, the number of qubits implemented in the PQC affects the size of the image that the PQC circuit can process, and the time it takes to complete algorithm training. Due to the inherent qubit noise of the NISQ-era qubits, the use of fewer qubits in quantum circuit designs is promoted [76].

iv. Qubit Arrangement

Qubit arrangement determines how each qubit in the quantum circuit is arranged as a circuit. There are two methods of qubit arrangement: Linear and Grid arrangement. We have adopted the grid arrangement of qubits to implement the 2D QCNN layers.

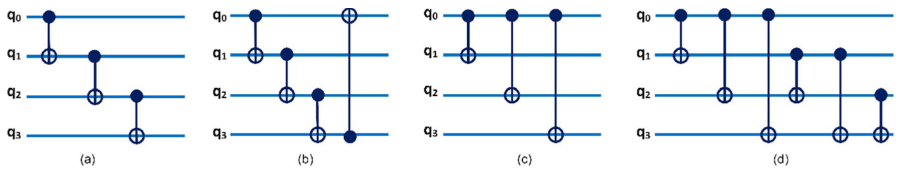


Fig. 4 Inter-qubit Entanglement Connection Strategies. **a** Linear; **b** Cyclic; **c** Star-Connected; **d** Fully Connected

v. Entanglement Strategy

The entanglement strategy determines the inter-qubit connection by the two-qubit gate operation in the quantum circuit. In the gate-centric PQC architecture, all qubits implement gate operation at least once. Entanglement strategy commands which qubits connect with each other. There are four strategies of entanglement: linear, circular, star-connected and fully connected as represented in Fig. 4.

3.2 Quantum convolutional neural network

In classical machine learning, the feature extraction model learns the non-redundant meaningful information from the image data $\mathcal{X} = [\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \ldots, \mathbf{x}_N]$ through convolution layers, pooling layers and fully connected layers. The classical convolutional layer ($\updownarrow_C$) extracts feature values $x_{ij}^{\updownarrow_C}$ from the pixel values of the domain $\mathcal{X}$, where the classical filter scans across the local regions in the position (i, j) of the image and computes the dot product of kernel with the pixel values of the local region and the weights $w_{a,b}$ form $k_C \times k_C$ dimension classical convolutional kernel or classical filter as explained by (7), where k_C denotes the convolutional kernel size. The extracted features are mapped to the classical feature space ($\mathcal{F}$) represented by the classical feature map $\phi_C : \mathcal{X} \mapsto \mathcal{F}$.

$$x_{ij}^{\ell_C} = \sum_{a,b=1}^{k_C} w_{a,b} x_{i+a-k_C/2,\, j+b-k_C/2}^{\ell_C - 1} + \text{bias}, \tag{7}$$

The pooling layer downsizes the feature map and the fully connected layers flatten the data and compile the data extracted by the previous layers for classification. The weights in the filter are updated and optimized during the training process.

Let k_Q denote the quantum kernel size. Similarly, in the gate-centric quantum setting, the PQCs can be designed in the form of a QCNN layer comprising quantum filters for performing convolution inner product between image matrix $\mathcal{X}$, and the quantum kernel matrix $\mathcal{K} : k_Q \times k_Q$, implemented through quantum kernel circuit [9]. Encoding circuit $E(\mathbf{x})$ implements the quantum feature map encoding function ϕ_Q: $\mathcal{X} \mapsto \mathcal{H}$ upon the input image data $\mathcal{X}$, and maps the input data into Hilbert space $\mathcal{H}$ by encoding the input image pixels from $\mathcal{X}$ into quantum states ϕ_Q: $x \mapsto |\phi_Q(\mathbf{x})\rangle$. Variational circuit $V(\theta)$ is responsible for learning the features of the image by performing unitary gate operations before executing multiple measurements of the quantum state

$|\phi_{\mathrm{Q}}(\mathbf{x})\rangle$, and passing the values for the higher-level feature extraction to the classical CNN architecture. In the following sections, we discuss these underlying QML processes in detail.

3.3 Components of proposed quantum convolutional neural network layer

In the proposed QSurfNet, we implement 2 PQCs to perform quantum convolutional operations on the input image-Quantum Kernel Circuit and Quantum Unitary Circuit, together they form the QCNN layer. The following sections explain the building blocks of the proposed QCNN layer design method as depicted in Fig. 5.

The components of the proposed QCNN layer are explained as follows:

A. Quantum Kernel Circuit

The convolution operation in the QCNN layer is implemented by a PQC designed to function as a quantum convolution kernel. The PQC quantum filter scans across the receptive field of the image, just like a classical CNN filter. We implement a 4-qubit EfficientSU2 [102] PQC as a 2×2 quantum kernel filter in the grid-qubit arrangement. EfficientSU2 is a hardware-efficient special unitary quantum circuit of degree 2 with 2×2 unitary matrix element of determinant $+ 1$. It consists of the multi-depth parameterized rotational Pauli-Y and Rotational Pauli-Z gates at the beginning and end of the circuit separated with CX (Control NOT) Gates in the various entanglement strategies. We implement the circular entanglement strategy where each qubit q_0, q_1, q_2 and q_3 is connected to its next nearest neighbour in a circular arrangement as represented in Fig. 6.

In the quantum convolution layer $\updownarrow_{\mathrm{Q}}$, the kernel $\mathcal{K}$ scans across the image $\mathcal{X}$ and performs the convolution operation extracting the quantum inner product, represented as $\langle \mathcal{X}^{\updownarrow_{\mathrm{Q}}} | \mathcal{K}^{\updownarrow_{\mathrm{Q}}} \rangle$ in (8), between the kernel and the corresponding overlapping image region, transforming the pixel values in the local region of image into the quantum

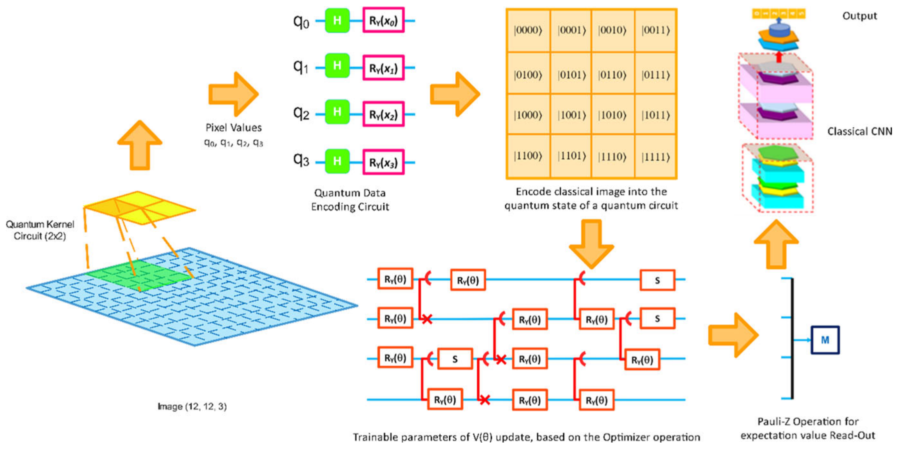

Output
Classical CNN
q_0
q_1
q_2
q_3
H
$R_Y(x_0)$
$R_Y(x_1)$
$R_Y(x_2)$
$R_Y(x_3)$
Pixel Values
q_0, q_1, q_2, q_3
Quantum Data Encoding Circuit
$|0000\rangle$ $|0001\rangle$ $|0010\rangle$ $|0011\rangle$
$|0100\rangle$ $|0101\rangle$ $|0110\rangle$ $|0111\rangle$
$|1000\rangle$ $|1001\rangle$ $|1010\rangle$ $|1011\rangle$
$|1100\rangle$ $|1101\rangle$ $|1110\rangle$ $|1111\rangle$
Encode classical image into the quantum state of a quantum circuit
Quantum Kernel Circuit (2x2)
Image (12, 12, 3)
$R_Y(\theta)$
S
M
Trainable parameters of V(θ) update, based on the Optimizer operation
Pauli-Z Operation for expectation value Read-Out

Fig. 5 Quantum Convolution Neural Network Operation

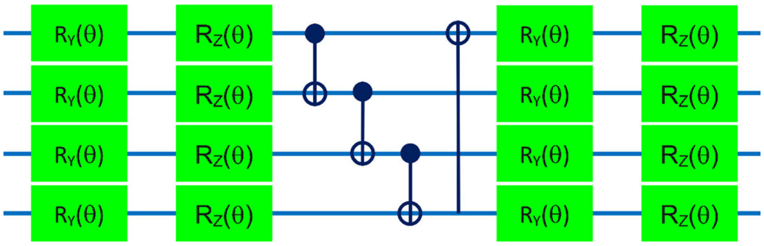


Fig. 6 Quantum Kernel Circuit

states of its data circuit [9, 101, 103].

$$f\left(\mathcal{X}^{\ell_{Q}+1}\right) := f\left(\mathcal{X}^{\ell_{Q}} * \mathcal{K}^{\ell_{Q}}\right) \tag{8}$$

B. Quantum Convolution Layer

The convoluted input $\langle \mathcal{X}^{\updownarrow_{Q}} | \mathcal{K}^{\updownarrow_{Q}} \rangle$ is then processed by the unitary PQC [9] as explained in Sect. 3.1 and shown in Fig. 5. The QCNN layer $\updownarrow_{Q}$ performs convolution operation based on (9) upon the information held in the quantum kernel and passes the inner product values to quantum unitary circuit $U(\mathbf{x}, \theta)$ before obtaining the final readout measurement data as explained through (9–12) [103].

$$D^{\ell_{Q}+1} = \frac{\left(D^{\ell_{Q}} - k_{Q} + 2P\right)}{S} + 1, \tag{9}$$

In (9), $D^{\ell_{Q}+1}$ is the output of the convolution layer, $D^{\updownarrow_{Q}}$ is the input to the convolutional layer $\updownarrow_{Q}$, P is the padding or margin pixels applied on the image to prevent image shrinking, and S is the kernel stride on the image during convolution.

Within $U(\mathbf{x}, \theta)$, the input information is processed by the gate operations. To optimize the parameters of $V(\theta)$, the expectation value of cost function M from (2) is optimized. Finally, the quantum state in the computational basis is reduced to binary value m through measurement by performing Pauli-Z operation to obtain final value $\mathcal{F} \in \{0, 1\}^n$ and linearly combining the results as expressed in (10) to obtain measurement binary value M where $m(\cdot) \in \{-1, 1\}$. The $U(\mathbf{x}, \theta)$ circuit is executed for a specific number of times or "shots", and the measurement results are averaged to obtain the final expectation value. The probability of observing z is expressed as (11) [104].

$$M = \sum_{z \in \{0,1\}^n} m(z) |z\rangle\langle z| \tag{10}$$

$$|\langle z|V(\theta)||\phi(\mathbf{x})\rangle|^2 = \left\langle \phi(\mathbf{x}) \middle| V^{\dagger}(\theta) \middle| z \right\rangle \langle z|V(\theta)|\phi(\mathbf{x})\rangle \tag{11}$$

Similar to Eq. (7), the quantum unitary circuit $U(\mathbf{x}, \theta)$ consisting of trainable parameters in variational circuit $V(\theta)$ optimizes the weights during the training process as expressed in (12) [104].

$$f(\mathbf{x}, \theta) = \phi(\mathbf{x}) \Big| U^{\dagger}(\mathbf{x}, \theta) M U(\mathbf{x}, \theta) \Big| \phi(\mathbf{x})\rangle + \text{bias} \tag{12}$$

We observed in our experiments that existing quantum hardware can reliably implement a few or up to 10 physical qubits. A large number of qubits in the PQC do not give coherent results because the noise of NISQ qubits overwhelms the signal in the circuit and slows down the computation for QCNN. The available literature confirms that implementing quantum circuits as convolutional operators is computationally expensive, as it can also be inferred from (6). Also, since k_Q^2 qubits form quantum kernel $\mathcal{K}$ of size $k_\mathrm{Q} \times k_\mathrm{Q}$, and, we decided to use only 4 qubits for the quantum kernel and quantum data processing circuits each. We developed a 2×2 quantum kernel $\mathcal{K}$ and a 4×4 unitary circuit $U(\mathbf{x}, \theta)$ in the grid qubit arrangement. The 4 qubits of $U(\mathbf{x}, \theta)$ and $\mathcal{K}$ when arranged in grid qubit arrangement, quantum circuit gates are arranged on the 2D square lattice of 4 quantum wires. In Fig. 5, quantum unitary sections $|0000\rangle \ldots |1111\rangle$ and in Fig. 6, quantum kernel sections (q_0, q_2), (q_0, q_3), (q_1, q_3) and (q_1, q_2) represent the identity of the individual quantum grid blocks. Although each qubit is initiated with $|0\rangle$, the complex matrix operations within the circuit make it impossible to ascertain the value held by any parameter at any moment, hence these PQCs are performing black box quantum neural network computations just like classical CNN networks. In the next section, we discuss QSurfNet architecture in detail.

3.4 QSurfNet model

QSurfNet integrates the QCNN layer developed in the proposed method with the CNN architecture based on the existing SurfNetv2 [11] architecture to perform surface defect recognition on the input 3-channel RGB images of four classes of the CSB dataset and six classes of NEU dataset. The following sections explain the working of the QSurfNet model in detail.

A. Image Pre-processing

The image data pre-processing module resizes the input image to the specified dimensions mentioned in Tables 4 and 5, then splits the RGB channels of the image and applies PCA dimensionality reduction in the individual channels to further diminish the redundant features and then stacks those channels back as a whole image. The images are then pre-processed with pixel normalization and the subsequent elements of the label set Y are one-hot encoded for the image classes. These images are then fetched by the feature extraction module as the dataset $\mathcal{X}$ with the corresponding targets Y. The resultant reduced dimensionality RGB images are used as input data for the QSurfNet .

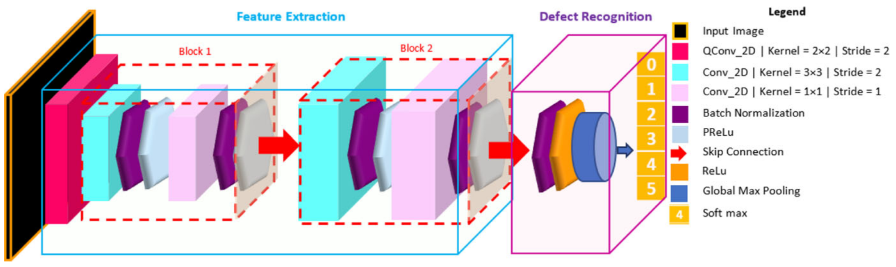


Fig. 7 QSurfNet architecture

B. Model Architecture

The QCNN layer was tested to be implemented for extracting low-level features and high-level features. But the experimental results demonstrate that the 2-D quantum convolution layer (QConv_2D) must not be implemented for extracting high-level features in the deep neural networks because the QConv_2D layer is not capable of processing a large number of learned parameters. Hence, we directly feed the pre-processed image data to the QConv_2D layer and let it process the low-level features. The output data from QConv_2D passes to the classical CNN architecture as represented by Fig. 7.

The QCNN module is followed by the CNN part of QSurfNet, which is the VGG-based residual learning architecture. It consists of two feature extracting blocks and a defect recognition module.

Each feature extracting block contains two specific convolutional layers as shown in Fig. 7. The first 2-D convolutional layer (Conv_2D) uses 3×3 kernel size to gain receptive fields without missing important features with stride step of 2, without padding. The convolution output is passed through the batch normalization layer to stabilize the learning process, and then through the nonlinearity function which allows matrix operations that facilitate deep neural networks, implementing progressive learning action in the deep neural networks across the CNN.

We apply parametric rectified linear unit (PReLU) expressed in (13) as the activation function of choice in the feature extraction layer between subsequent convolutional layers, where x is the input of the nonlinear activation function, and α is the slope controlling trainable coefficient for the slope for negative values.

$$\mathrm{PReLU}(x) = \begin{cases} x, if\ x > 0 \\ \alpha x, if\ x \le 0 \end{cases} \tag{13}$$

The nonlinear output product is then passed through the second Conv_2D layer without padding with 1×1 kernel size and stride step of 1 for down-sampling, again followed by the batch normalization and PReLU. The second convolution layer realizes the addition of the input and output feature maps of only the second Conv_2D layer by a skip connection before passing the output on to the next block. The feature extraction module has the following unique features:

1. No drop-out layer.

2. No pooling operation. Hence, we did not implement quantum pooling circuit either in the QCNN module.

The same process is also adopted in the second feature-extracting block and its output is then processed by the defect detection module.

To impede overfitting, we implement the minimalist defect detection module with the Global Max Pooling operation to compensate for the absence of pooling operation in the feature extracting module, replace the additional dense layers needed, and implement dimensionality reduction in the feature maps while utilizing only one fully connected dense layer to reduce the amount of parameter calculations. The Global Max Pooling operation processes the input tensor of dimension $width \times height \times channels$ to compute the maximum value out of all the values across the $width \times height$ matrix for each of the input *channels* to output a 1-dimensional tensor of size (*channels*).

In the QSurfNet architecture, the defect recognition module accepts input from the second feature extraction block and then implements batch normalization and the nonlinearity activation function ReLU as expressed in (14) on the input.

$$\mathrm{ReLU}(x) = \begin{cases} x, \; if & x > 0 \\ 0, \; if & x \le 0 \end{cases} \tag{14}$$

The activation function product is then passed to the global max pooling layer before the classification operation by the last fully connected dense layer using the Softmax activation function expressed by (15) to identify the defect category based on the features learned, where $\mathbf{x}$ is the 1D tensor of size incoming (*channels*) from the global max pooling layer, $\mathbf{w}_i$ is the ith weight vector of the fully connected layer output, and C is the number of output categories. The Softmax operation is read as the conditional probability of occurrence of the jth category given the input tensor vector $\mathbf{x}$ as expressed in (15).

$$p(y = j|\mathbf{x}) = \frac{e^{\mathbf{x}^T \mathbf{w}_j}}{\sum_{i=1}^{C} e^{\mathbf{x}^T \mathbf{w}_i}} \tag{15}$$

The output dimension of the Softmax layer can be set as per the categories defined in the training dataset as shown in the pictorial representation of QSurfNet architecture in Fig. 7.

The last fully connected layer classifies the surface defect category based on the feature detection of the QCNN and CNN feature extracting module.

C. Model Training

SurfNetv2 uses the RMSprop as the optimizer for the feature of suppressing the gradient oscillations but when implemented on QSurfNet, it gave unprecedented overfitting. Hence, we implement the Nesterov accelerated stochastic gradient descent (SGD) [99, 105] as a hyperparameter setting since SGD yields state-of-the-art results with significantly fewer circuit executions and measurements, while reducing the chances of overfitting with better generalization performance for training example pair $\mathcal{X}_i$ and label Y_i with the learning rates 10^{-3} and 10^{-6} for the CSB dataset, and the NEU dataset, respectively.

To further avoid overfitting, we implemented K-fold cross-validation and divided the training data into 10 sets to train and verify our data. For the parameter update, we implement categorical Cross-Entropy loss function $\mathcal{L}$ for classification through the Softmax activation function explained in (15), as expressed by

$$\mathcal{L}(\mathbf{t}, \mathbf{x}) = -\sum_{j=1}^{C} t_j \log(p(y = j|\mathbf{x})), \tag{16}$$

where $\mathbf{t} = [t_1, t_2, \ldots, t_C]$ denotes the desired target, which is the 1D one-hot encoded vector, and t_j is the jth output of the target $\mathbf{t}$.

4 Results and discussion

The quantitative results, generated through the computation infrastructure specifications represented in Table 1, are charted in the following sections. Note that according to [10], the TensorFlow-Quantum framework allows users to simulate quantum processing units (QPUs) while designing, training, and testing hybrid quantum-classical models, and finally run the quantum portions of these models on an actual QPU when testing online. Therefore, the proposed hybrid quantum-classical model can be run on a classical computer by simulating the QPU, while the proposed PQC design can be run on a real QPU.

1. Dataset

We train, test and benchmark QSurfNet on the two datasets namely CSB and NEU. NEU is a 6-category public dataset constituting hot-rolled steel strip RGB images commonly used in the surface defect recognition research, and CSB is the 4-category private RGB image dataset sourced from Jia Dah Chemical Industrial Co. Ltd., Taiwan, whose extraction and development is explained in detail in [11]. The datasets have the equal number of samples across their categories as represented in Table 2.

Table 1 Computational Infrastructure

Part	Item	Specification
Hardware	CPU	Intel® Core™ i9-12950HX
	RAM	32 GB
	GPU	NVIDIA® GeForce RTX™ 3070 Ti
Software	System	Ubuntu OS; TensorFlow-Quantum circuit simulator
	Programming Language	Python 3
	Backend	TensorFlow, TensorFlow-Quantum [10]

Table 2 Surface Defect Datasets

Dataset		Class/category	Sample number
Private	Calcium Silicate Board (CSB)	Crash	4960
		Dirty	4960
		Uneven	4960
		Normal	4960
Public	Northeastern University (NEU)	Rolled-in Scale	300
		Patches	300
		Crazing	300
		Pitted Surface	300
		Inclusion	300
		Scratches	300

2. Performance Evaluation

In addition to the K-Fold cross-validation, and training upon two distinguished datasets to establish its applicability across different types of defect surfaces on different materials, QSurfNet is also benchmarked for its performance against SurfNetv2 [11], SurfNet [12], ResNet18 [13], DenseNet [14], VGG16 [15], and MobileNetv2 [16] under the same datasets for the training epoch settings represented in Table 3.

We train QSurfNet under the model training settings described in Section 3.4.C and Table 3, and monitor the algorithm performance through the classification metrics described by Accuracy, which measures the total correct predictions under all cases as expressed in (17); Recall, which monitors the percentage of the true positives detected as expressed in (18); Precision, which measures the cases when the class prediction is correct and the algorithm could flag it as true positives as expressed in (19) and $F1$-Measure as represented in (20).

Table 3 Training Epoch Number for different datasets

Model	Epoch Number	
	CSB Dataset	NEU Dataset
QSurfNet	200	400
SurfNetv2		
SurfNet	300	500
Resnet	100	150
DenseNet	150	150
VGG16	200	300
MobileNetv2	200	300

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \tag{17}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{18}$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{19}$$

$$F1 - \text{Measure} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{20}$$

where TP = True Positive, the case is true and the predicted result is also true; TN = True Negative, the case is negative and the predicted result is also negative; FP = False Positive, the case is negative and the predicted result is True; FN = False Negative, the case is true and the predicted result is also negative. Note that equations (17–29) are four important evaluation metrics used to assess the performance of image recognition models. Accuracy is a measure of how often the model correctly classifies the positive and negative samples. Precision is a measure of the model's ability to correctly identify a particular class. Recall is a measure of the model's ability to correctly identify all instances of a particular class. The $F1$-Measure is a metric that combines precision and recall to provide an overall assessment of the model's performance. A better image recognition model should be able to achieve high accuracy, high precision, high recall and high $F1$-measure simultaneously.

The experimental test results based on the classification metrics described by the equations (17–20) of the CSB dataset and NEU dataset are represented in Tables 4 and 5, respectively. In addition, we also recorded the frame rate (FPS) of each model during model testing to compare the processing speed of each model during real-time testing.

All the experiments for benchmarking were trained with 1 quantum convolutional layer and two classical convolutional blocks as discussed in Sect. 3.4, under all the same hyperparameters and data except for the different image sizes. Classical algorithms were capable of processing bigger images under standard computational cost, but QSurfNet could only process small-size images because the size of encoding circuits and the quantum kernel is dependent on the number of qubits. Also, since in 1 QCNN layer, 2 quantum circuits are processing the image data, it takes a lot of computation resources and time to train the data. Hence, for the sake of computational efficiency and the optimum image size permissible by the quantum variational circuit $V(\theta)$, we train QSurfNet on the image sizes $8 \times 8 \times 3$ and $12 \times 12 \times 3$.

The experimental results shown in Tables 4 and 5 can be summarised as follows:

1. For the smallest total parameters, compared with the other state-of-the-art algorithms, QSurfNet performs the best in terms of the classification metrics defined in (17–20).
2. The QSurfNet model performed better on the CSB dataset than the NEU dataset for the image sizes $8 \times 8 \times 3$ and $12 \times 12 \times 3$.
3. During model training, the QSurfNet model was trained using the computational resources detailed in Sect. 4, along with the datasets specified in Table 2 and the

Table 4 Experimental results of the CSB dataset

Model	Image Size	Accuracy	Recall	Precision	F1-Measure	FPS	Parameters
QSurfNet	8 × 8	98.31%	97.86%	97.87%	97.86%	200.03	235,606
	12 × 12	99.99%	99.99%	100.00%	99.99%	199.91	
SurfNetv2	128 × 128	99.90%	99.89%	99.90%	99.90%	199.38	8.2 M
	256 × 256	99.83%	99.83%	99.84%	99.84%	157.91	
SurfNet	128 × 128	99.68%	99.65%	99.70%	99.68%	198.93	2.4 M
	256 × 256	99.33%	99.22%	99.39%	99.31%	154.21	
ResNet18	128 × 128	99.79%	99.79%	99.80%	99.79%	142.36	11.1 M
	256 × 256	99.82%	99.81%	99.82%	99.82%	124.21	
DenseNet	128 × 128	99.71%	99.71%	99.71%	99.71%	42.77	7.0 M
	224 × 224	99.37%	99.37%	99.38%	99.38%	40.20	
VGG16	128 × 128	83.45%	83.38%	83.53%	83.45%	230.07	14.7 M
	256 × 256	85.88%	85.77%	85.92%	85.84%	128.05	
MobileNetv2	128 × 128	97.28%	97.22%	97.34%	97.28%	97.42	2.2 M
	256 × 256	98.37%	98.35%	98.39%	98.37%	89.64	

Table 5 Experimental results of the NEU dataset

Model	Image Size	Accuracy	Recall	Precision	F1-Measure	FPS	Parameters
QSurfNet	8 × 8	97.23%	97.23%	97.31%	97.23%	146.34	235,606
	12 × 12	99.79%	99.79%	99.89%	99.93%	165.86	
SurfNetv2	64 × 64	99.37%	99.37%	99.44%	99.40%	125.45	8.2 M
	128 × 128	99.75%	99.75%	99.75%	99.75%	134.74	
SurfNet	64 × 64	99.37%	99.25%	99.44%	99.34%	128.83	2.4 M
	128 × 128	99.31%	99.25%	99.44%	99.34%	134.21	
ResNet18	64 × 64	99.50%	99.31%	99.55%	99.43%	112.31	11.1 M
	128 × 128	99.62%	99.62%	99.69%	99.66%	104.29	
DenseNet	64 × 64	99.06%	98.94%	99.43%	99.17%	23.85	7.0 M
	128 × 128	99.62%	99.62%	99.75%	99.69%	20.77	
VGG16	64 × 64	95.94%	95.19%	96.36%	95.76%	112.95	14.7 M
	128 × 128	98.00%	97.81%	98.17%	97.99%	134.90	
MobileNetv2	64 × 64	94.38%	93.19%	94.84%	93.94%	33.86	2.2 M
	128 × 128	96.94%	96.62%	97.24%	96.92%	54.67	

training epoch settings outlined in Table 3. When training for an image size of 12 $\times$ 12 $\times$ 3, the CSB dataset took 5 s per step while the NEU dataset took 3 s per step. For the 8 $\times$ 8 $\times$ 3 image size, it took 3 s per step for the CSB dataset and 2 s per step for the NEU dataset.

4. During model testing, the real-time recognition speed, measured in frames-per-second (FPS), of QSurfNet was only slightly faster for the CSB dataset and fairly faster for the NEU dataset compared to the other state-of-the-art algorithms. Nevertheless, QSurfNet was the fastest among all the other state-of-the-art models in both datasets. Hence, the experimental results in Tables 4 and 5 show that, despite its superior performance, QSurfNet is a computationally expensive algorithm to train, but very efficient to deploy.
5. In general, the frame rate of a real-time image recognition system should be at least 20 FPS. In the model testing, although all compared methods meet this requirement, the proposed QSurfNet not only achieves higher than 100 FPS, but also provides faster frame rates than SurfNet and SurfNetv2.
6. Decreasing the input image size, the training speed of QSurfNet increases considerably, but performance in terms of the classification metrics defined in (17–20) depreciates.
7. The test results establish the competitive edge of QML.

QML aids classical machine learning and deep learning to enhance the end results [75, 106, 107]. QCNN layer, even at the low-level feature extraction implements global feature correlation in comparison to local region feature correlation implemented by classical CNN [101], which is why adding even a few QCNN layers to the classical CNN can contribute to considerable performance enhancement of the algorithm [90, 93, 94, 103]

One of the limitations of classical deep convolutional neural networks is gradient vanishing due to loss of information as the depth of the network increases which increases learning time [108]. However, gradient scaling in the QCNN does not encounter barren plateau due to vanishing gradient [109]. It gives QCNNs the advantage of efficient learning capability and lesser information loss.

5 Conclusion and future work

Through this research, we were successful in developing a methodology to convert parameterized quantum circuits into the 2-D QCNN layer and implemented it with the classical CNN Surfnetv2 to develop our proposed hybrid quantum-classical neural network QSurfNet.

We implemented a 4-qubit EfficientSU2 as the quantum kernel, and the proposed PQC design [9] as 4-qubit encoding circuit and variational circuit for global feature correlation and convolution operation on the image dimensions sizes $8\times8\times3$ and $12\times12\times3$. The experimental results are compared with purely classical state-of-the-art CNN algorithms. Consequently, QSurfNet proves that despite quantum image processing is limited by the number of qubits which can be coherently implemented in the quantum circuits and the size of the image it may process, yet, it may generate competitive results in comparison to classical deep learning methods.

The proposed algorithm is capable of customization of the PQC circuit, QCNN layer and the algorithm parameters. QSurfNet is first of its kind QCNN algorithm that has 2-D QCNN layers with just like classical architecture modularity, giving it the flexibility to adjust to any application use case and transform any state-of-the-art classical algorithm into an enhanced performance hybrid quantum-classical convolutional neural network.

The algorithm is successfully tested on real-time surface defect recognition of small-size RGB spatial images of the public NEU dataset and the private CSB dataset and applies to a wide range of defect detection use cases.

In the future, we will try to design new PQC circuit designs and implement it as QCNN layer on different state-of-the-art algorithms to test and expand the implementation possibilities of various applications of surface defect recognition and image processing algorithms. Also, for our results, we implemented only one quantum convolution layer and two classical convolutional blocks through the quantum simulator due to computational budget considerations. The experiment results for the CSB dataset in Table 4 reveal that QSurfNet consistently gave zero False Positives for the image size 12×12, which is a rare phenomenon in deep learning studies. Chances are it might be silently skipping wrong predictions, which we would like to research in the future QCNN study. For future research, we will also experiment with different combinations of the quantum and classical layers to benchmark the best QSurfNet model version and generate results using quantum devices. For the image recognition of the bigger size images and improving the time cost of training the algorithm, the design of bigger convolution kernels out of PQCs having more than 4 qubits is a research-worthy direction of future development.

Acknowledgements This research was supported by the Ministry of Science and Technology of Taiwan, R.O.C., under grant MOST 111-2221-E-032-031.

Declarations

Conflict of interest The authors have no conflict of interest regarding the research method, results and funding. All data presented in this study are available in this article.

References

1. du Castel, B.: Pattern activation/recognition theory of mind. Front. Comput. Neurosci. **9**, 90 (2015)
2. Yann, L., Yoshua, B., Geoffrey, E.H.: Deep learning. Nature (2015). https://doi.org/10.1038/nature14539
3. Piron, C.: On the foundations of quantum physics. In: Quantum Mechanics, Determinism, Causality, and Particles, pp. 105–116. Springer (1976)
4. Maxwell, J.C.: VIII. A dynamical theory of the electromagnetic field. Philos. Trans. R. Soc. Lond. **155**, 459–512 (1865). https://doi.org/10.1098/rstl.1865.0008
5. Nelkon, M.M., Parker, R.: Advanced Level Physics. Heinemann Educational Books Ltd (1970)
6. Benbarrad, T., Salhaoui, M., Kenitar, S.B., Arioua, M.: Intelligent machine vision model for defective product inspection based on machine learning. J. Sens. Actuator Netw. **10**(1), 7 (2021)

7. De Stefano, V.: “Negotiating the algorithm”: automation, artificial intelligence, and labor protection. Comp. Labor Law Policy J. **41**, 15 (2019). https://doi.org/10.2139/ssrn.3178233
8. Le, H.N., Bao, T.V., Debnath, N.C.: Computer vision–based system for automation and industrial applications. In: Artificial Intelligence and the Fourth Industrial Revolution, pp. 3–43. Jenny Stanford Publishing (2022)
9. Mishra, S., Tsai, C.-Y.: Design of superior parameterized quantum circuits for quantum image classification. Presented at the 2022 14th international conference on computer and automation engineering (ICCAE), Brisbane, Australia (2022)
10. Broughton, M., et al.: TensorFlow quantum: a software framework for quantum machine learning. arXiv preprint arXiv:2003.02989, pp. 56 (2020). https://doi.org/10.48550/arXiv.2003.02989
11. Tsai, C.-Y., Chen, H.-W.: SurfNetv2: an improved real-time SurfNet and its applications to defect recognition of calcium silicate boards. Sensors **20**(16), 4356 (2020). https://doi.org/10.3390/s20164356
12. Arikan, S., Varanasi, K., Stricker, D.: Surface defect classification in real-time using convolutional neural networks. arXiv: Image and Video Processing 2019. https://doi.org/10.48550/arXiv.1904.04671
13. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016 (2016), pp. 770–778. https://doi.org/10.1109/CVPR.2016.90
14. Iandola, F., Moskewicz, M., Karayev, S., Girshick, R., Darrell, T., Keutzer, K.: DenseNet: implementing efficient convnet descriptor pyramids. arXiv : Computer Vision and Pattern Recognition 2014. https://doi.org/10.48550/arXiv.1404.1869
15. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. In: International Conference on Learning Representations (ICLR 2015), San Diego, CA (2015), Ed., 7–9 May 2015, p. 14. https://doi.org/10.48550/arXiv.1409.1556
16. Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., Chen, L.-C.: "Mobilenetv2: inverted residuals and linear bottlenecks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA (2018), pp. 4510–4520. https://doi.org/10.48550/arXiv.1801.04381
17. Brigham, E.O.: The Fast Fourier Transform and Its Applications. Prentice-Hall Inc, NJ, United States (1988)
18. Guifang, W., Haichao, Z., Xiuming, S., Jinwu, X., Ke, X.: A bran-new feature extraction method and its application to surface defect recognition of hot rolled strips (2007). https://doi.org/10.1109/ical.2007.4338916
19. Zhanjiang, Y., Xiaozhou, L., Huadong, Y., Dan, X., Aimei, L., Hao, L.: Research on surface defect inspection for small magnetic rings (2009). https://doi.org/10.1109/icma.2009.5246333
20. Wang, F.-L., Zuo, B.: Detection of surface cutting defect on magnet using Fourier image reconstruction. J. Cent. South Univ. **23**(5), 1123–1131 (2016). https://doi.org/10.1007/s11771-016-0362-y
21. Chan, C.-H., Pang, G.K.: Fabric defect detection by Fourier analysis. IEEE Trans. Ind. Appl. **36**(5), 1267–1276 (2000). https://doi.org/10.1109/28.871274
22. Hu, G.-H., Wang, Q.-H., Zhang, G.-H.: Unsupervised defect detection in textiles based on Fourier analysis and wavelet shrinkage. Appl. Opt. **54**(10), 2963–2980 (2015). https://doi.org/10.1364/AO.54.002963
23. Mahajan, P., Kolhe, S., Patil, P.: A review of automatic fabric defect detection techniques. Adv. Comput. Res. **1**(2), 18–29 (2009)
24. Boujelbene, R., Jemaa, Y.B., Zribi, M.: A comparative study of recent improvements in wavelet-based image coding schemes. Multimed. Tools Appl. **78**(2), 1649–1683 (2019). https://doi.org/10.1007/s11042-018-6262-4
25. Shensa, M.J.: The discrete wavelet transform: wedding the a trous and Mallat algorithms. IEEE Trans. Signal Process. **40**(10), 2464–2482 (1992). https://doi.org/10.1109/78.157290
26. Abbate, A., Koay, J., Frankel, J., Schroeder, S.C., Das, P.: Signal detection and noise suppression using a wavelet transform signal processor: application to ultrasonic flaw detection. IEEE Trans. Ultrason. Ferroelectr. Freq. Control **44**(1), 14–26 (1997). https://doi.org/10.1109/58.585186
27. Rosenboom, L., Kreis, T., Jüptner, W.: Surface description and defect detection by wavelet analysis. Meas. Sci. Technol. **22**(4), 045102 (2011). https://doi.org/10.1088/0957-0233/22/4/045102

28. Wu, X.-Y., Xu, K., Xu, J.-W.: Application of undecimated wavelet transform to surface defect detection of hot rolled steel plates. In: 2008 Congress on Image and Signal Processing, vol. 4, pp. 528–532. IEEE (2008). https://doi.org/10.1109/CISP.2008.278
29. Jeon, Y.-J., Yun, J.P., Choi, D.-C., Kim, S.W.: Defect detection algorithm for corner cracks in steel billet using discrete wavelet transform. In: 2009 ICCAS-SICE, Fukuoka, Japan, 18–21 Aug 2009, pp. 2769–2773. IEEE (2009)
30. Chang, Q., Zhang, Y., Sun, Z.: Research on surface defect detection algorithm of ice-cream bars based on clustering. In: 2019 IEEE 3rd Information Technology, Networking, Electronic and Automation Control Conference (ITNEC), pp. 537–541. IEEE (2019)
31. Khumaidi, A., Yuniarno, E.M., Purnomo, M.H.: Welding defect classification based on convolution neural network (CNN) and Gaussian kernel. In: 2017 International Seminar on Intelligent Technology and Its Applications (ISITIA), pp. 261–265. IEEE (2017). https://doi.org/10.1109/ISITIA.2017.8124091
32. Maestro-Watson, D., Balzategui, J., Eciolaza, L., Arana-Arexolaleiba, N.: Deep learning for deflectometric inspection of specular surfaces. In: The 13th International Conference on Soft Computing Models in Industrial and Environmental Applications, pp. 280-289. Springer (2018)
33. Song, L., Li, X., Yang, Y., Zhu, X., Guo, Q., Yang, H.: Detection of micro-defects on metal screw surfaces based on deep convolutional neural networks. Sensors **18**(11), 3709 (2018). https://doi.org/10.3390/s18113709
34. Zhang, A., et al.: Automated pixel-level pavement crack detection on 3D asphalt surfaces using a deep-learning network. Comput. Aided Civ. Infrastruct. Eng. **32**(10), 805–819 (2017). https://doi.org/10.1111/mice.12297
35. Wei, R., Bi, Y.: Research on recognition technology of aluminum profile surface defects based on deep learning. Materials **12**(10), 1681 (2019). https://doi.org/10.3390/ma12101681
36. Ren, S., He, K., Girshick, R., Sun, J.: Faster r-cnn: towards real-time object detection with region proposal networks. In: Advances in Neural Information Processing Systems, vol. 28 (2015)
37. Sison, H., Konghuayrob, P., Kaitwanidvilai, S.: A convolutional neural network for segmentation of background texture and defect on copper clad lamination surface. In: 2018 International Conference on Engineering, Applied Sciences, and Technology (ICEAST), pp. 1–4. IEEE (2018)
38. Cuevas, E., Zaldivar, D., Pérez-Cisneros, M.: A novel multi-threshold segmentation approach based on differential evolution optimization. Expert Syst. Appl. **37**(7), 5265–5271 (2010). https://doi.org/10.1016/j.eswa.2010.01.013
39. Tang, J.: A color image segmentation algorithm based on region growing. In: 2010 2nd International Conference on Computer Engineering and Technology, vol. 6, pp. V6–634-V6–637. IEEE (2010). https://doi.org/10.1109/ICCET.2010.5486012
40. Chuang, K.-S., Tzeng, H.-L., Chen, S., Wu, J., Chen, T.-J.: Fuzzy c-means clustering with spatial information for image segmentation. Comput. Med. Imaging Graph. **30**(1), 9–15 (2006). https://doi.org/10.1016/j.compmedimag.2005.10.001
41. Dhanachandra, N., Manglem, K., Chanu, Y.J.: Image segmentation using K-means clustering algorithm and subtractive clustering algorithm. Procedia Comput. Sci. **54**, 764–771 (2015)
42. Mia, S., Rahman, M.M.: An efficient image segmentation method based on linear discriminant analysis and K-means algorithm with automatically splitting and merging clusters. Int. J. Imaging Robot. **18**(1), 62–72 (2018)
43. Xu, Y., Li, D., Xie, Q., Wu, Q., Wang, J.: Automatic defect detection and segmentation of tunnel surface using modified Mask R-CNN. Measurement **178**, 109316 (2021). https://doi.org/10.1016/j.measurement.2021.109316
44. Xu, R., Hao, R., Huang, B.: Efficient surface defect detection using self-supervised learning strategy and segmentation network. Adv. Eng. Inform. **52**, 101566 (2022). https://doi.org/10.1016/j.aei.2022.101566
45. Funck, J., Zhong, Y., Butler, D., Brunner, C., Forrer, J.: Image segmentation algorithms applied to wood defect detection. Comput. Electron. Agric. **41**(1–3), 157–179 (2003). https://doi.org/10.1016/S0168-1699(03)00049-8
46. Celik, T., Tjahjadi, T.: Unsupervised colour image segmentation using dual-tree complex wavelet transform. Comput. Vis. Image Underst. **114**(7), 813–826 (2010). https://doi.org/10.1016/j.cviu.2010.03.002

47. Lo, E.H.S., Pickering, M.R., Frater, M.R., Arnold, J.F.: Image segmentation from scale and rotation invariant texture features from the double dyadic dual-tree complex wavelet transform. Image Vis. Comput. **29**(1), 15–28 (2011). https://doi.org/10.1016/j.imavis.2010.08.004
48. Bergholm, V., et al.: Pennylane: automatic differentiation of hybrid quantum-classical computations. arXiv preprint arXiv:1811.04968 (2018)
49. Gonzalez, C.: Cloud based QC with Amazon braket. Digit. Welt **5**(2), 14–17 (2021). https://doi.org/10.1007/s42354-021-0330-z
50. Guerreschi, G.G., Hogaboam, J., Baruffa, F., Sawaya, N.P.: Intel quantum simulator: a cloud-ready high-performance simulator of quantum circuits. Quantum Sci. Technol. **5**(3), 034007 (2020)
51. Das, M., Miguel, A.D., Mehrotra, A., Sahgal, V.: Quantum defect analyser. In: BMW Group Quantumm Computing Challenge (2021). [Online]. Available: https://github.com/iotaisolutions/BMWQuantumChallenge2021/blob/main/Reports/BMW_Quantum_Computing_Challenge_Solution%20Description.pdf
52. Schuetz, M., Shishenina, E., Klepsch, J., Luckow, A.: Use case insights from the BMW Group quantum challenge. In: AWS Invent, Las Vegas, Nevada (2021), p. 23. [Online]. Available: https://d1.awsstatic.com/events/reinvent/2021/Use_case_insights_from_the_BMW_Group_quantum_challenge_QTC304.pdf
53. Glick, J.R., et al.: Covariant quantum kernels for data with group structure. arXiv : Quantum Physics, p. 18 (2021). https://doi.org/10.48550/arXiv.2105.03406
54. John, P.: Quantum computing in the NISQ era and beyond. Quantum **2**, 79 (2018). https://doi.org/10.22331/q-2018-08-06-79
55. Scott, A.: Read the fine print. Nat. Phys. (2015). https://doi.org/10.1038/nphys3272
56. Ian, G., Yoshua, B., Aaron, C.: Deep learning (2016). [Online]. Available: https://www.deeplearningbook.org/
57. Fei, Y., Salvador, E.V.-A.: Quantum image processing. Int. J. Quantum Inf. (2020). https://doi.org/10.1007/978-981-32-9331-1
58. Mateusz, O., Przemysław, S., Piotr, G.: Quantum image classification using principal component analysis. Theor. Appl. Inform. **27**(1), 1–12 (2015). https://doi.org/10.20904/271001
59. Jolliffe, I.: Principal component analysis. Springer Series in Statistics, pp. 338–372 (1986). https://doi.org/10.1002/9781118445112.stat06472
60. Liberty, E., Woolfe, F., Martinsson, P.-G., Rokhlin, V., Tygert, M.: Randomized algorithms for the low-rank approximation of matrices. Proc. Natl. Acad. Sci. **104**(51), 20167–20172 (2007). https://doi.org/10.1073/pnas.0709640104
61. Seth, L., Seth, L., Masoud, M., Patrick, R.: Quantum principal component analysis. Nat. Phys. **10**(9), 631–633 (2014). https://doi.org/10.1038/nphys3029
62. Ha, J., Heo, J.: Performance comparison of quantum phase estimation algorithm with different number of register qubits on noisy quantum processor. Presented at the IEEE Region 10 Symposium (TENSYMP), Jeju, Republic of Korea, 23–25 Aug 2021 (2021)
63. Zhaokai, L., et al.: Resonant quantum principal component analysis. Sci. Adv. **7**(34), 25–29 (2021). https://doi.org/10.1126/sciadv.abg2589
64. Rebentrost, P., Mohseni, M., Lloyd, S.: Quantum support vector machine for big data classification. Phys. Rev. Lett. **113**(13), 130503 (2014). https://doi.org/10.1103/PhysRevLett.113.130503
65. Delilbasic, A., Cavallaro, G., Willsch, M., Melgani, F., Riedel, M., Michielsen, K.: Quantum support vector machine algorithms for remote sensing data classification. Presented at the IEEE International Geoscience and Remote Sensing Symposium IGARSS, Brussels, Belgium, 11–16 July 2021, pp. 2153-7003 (2021)
66. Alina, O.S., Shapovalova, N.: Classification problem solving using quantum machine learning mechanisms. In: 4th Workshop for Young Scientists in Computer Science & Software Engineering, Kryvyi Rih, Ukraine, vol. 6, no. 7, CEUR Workshop Proceedings, p. 8 (2022). [Online]. Available: https://ceur-ws.org/Vol-3077/paper06.pdf. [Online]. Available: https://ceur-ws.org/Vol-3077/paper06.pdf
67. Kariya, A., Behera, B.K.: Investigation of Quantum Support Vector Machine for Classification in NISQ era. arXiv: Quantum Physics (2021). https://doi.org/10.48550/arXiv.2112.06912
68. Rana, A., Vaidya, P., Gupta, G.: A comparative study of quantum support vector machine algorithm for handwritten recognition with support vector machine algorithm. Mater. Today Proc. **56**, 2025–2030 (2022). https://doi.org/10.1016/j.matpr.2021.11.350
69. Cover, T., Hart, P.: Nearest neighbor pattern classification. IEEE Trans. Inf. Theory **13**(1), 21–27 (1967). https://doi.org/10.1109/TIT.1967.1053964

70. Lloyd, S., Mohseni, M., Rebentrost, P.: Quantum algorithms for supervised and unsupervised machine learning. arXiv preprint arXiv:1307.0411 (2013)
71. Li, J., Lin, S., Yu, K., Guo, G.: Quantum K-nearest neighbor classification algorithm based on Hamming distance. Quantum Inf. Process. **21**(1), 1–17 (2022). https://doi.org/10.1007/s11128-021-03361-0
72. Zak, M., Williams, C.P.: Quantum neural nets. Int. J. Theor. Phys. **37**(2), 651–684 (1998). https://doi.org/10.1023/A:1026656110699
73. Schuld, M., Sinayskiy, I., Petruccione, F.: An introduction to quantum machine learning. Contemp. Phys. **56**(2), 172–185 (2015). https://doi.org/10.1080/00107514.2014.964942
74. McCulloch, W.S., Pitts, W.: A logical calculus of the ideas immanent in nervous activity. Bull. Math. Biol. **5**(4), 115–133 (1943)
75. Maria, S., Sinayskiy, I., Ilya, S., Francesco, P.: The quest for a quantum neural network. Quantum Inf. Process. **13**(11), 2567–2586 (2014)
76. Schuld, M., Bocharov, A., Svore, K.M., Wiebe, N.: Circuit-centric quantum classifiers. Phys. Rev. A **101**(3), 032308 (2020)
77. Schuld, M., Petruccione, F.: Supervised Learning with Quantum Computers, p. 287. Springer, Berlin (2018)
78. Mari, A., Bromley, T., Izaac, J., Schuld, M., Killoran, N.: Transfer learning in hybrid classical-quantum neural networks. Quantum (2020). https://doi.org/10.22331/q-2020-10-09-340
79. Gawron, P., Lewiński, S.: Multi-spectral image classification with quantum neural network. In: IGARSS 2020–2020 IEEE International Geoscience and Remote Sensing Symposium, 26 Sep -2 Oct 2020, pp. 3513–3516 (2020). https://doi.org/10.1109/IGARSS39084.2020.9323065
80. Natansh, M., et al.: Medical image classification via quantum neural networks. arXiv : Quantum Physics (2021). https://doi.org/10.48550/arXiv.2109.01831
81. Alam, M., Satwik, K., Topaloglu, R., Swaroop, G.: ICCAD special session paper: quantum-classical hybrid machine learning for image classification. Presented at the 2021 IEEE/ACM International Conference On Computer Aided Design (ICCAD), Munich, Germany, 01–04 Nov 2021 (2021)
82. Kaya, M., Hajimirza, S.: Using a novel transfer learning method for designing thin film solar cells with enhanced quantum efficiencies. Sci. Rep. **9**(1), 1–10 (2019). https://doi.org/10.1038/s41598-019-41316-9
83. Azevedo, V., Silva, C., Dutra, I.: Quantum transfer learning for breast cancer detection. Quantum Mach. Intell. **4**(5), 1–14 (2022). https://doi.org/10.1007/s42484-022-00062-4
84. Mittal, S., Dana, S.K.: Gender recognition from facial images using hybrid classical-quantum neural network. In: 2020 IEEE Students Conference on Engineering & Systems (SCES), pp. 1–6. IEEE (2020) https://doi.org/10.1109/SCES50439.2020.9236711
85. Zhou, J., Gan, Q., Krzyżak, A., Suen, C.Y.: Recognition of handwritten numerals by quantum neural network with fuzzy features. Int. J. Doc. Anal. Recognit. (IJDAR) **2**(1), 30–36 (1999). https://doi.org/10.1007/s100320050034
86. Mittal, H., Saraswat, M., Bansal, J.C., Nagar, A.: Fake-face image classification using improved quantum-inspired evolutionary-based feature selection method. In: 2020 IEEE Symposium Series on Computational Intelligence (SSCI), pp. 989–995. IEEE (2020). https://doi.org/10.1109/SSCI47803.2020.9308337
87. Xu, Y., Zhang, X., Gai, H.: Quantum neural networks for face recognition classifier. Procedia Eng. **15**, 1319–1323 (2011)
88. Cong, I., Choi, S., Lukin, M.D.: Quantum convolutional neural networks. Nat. Phys. **15**(2), 1273–1278 (2019)
89. Guoming, C., et al.: Quantum convolutional neural network for image classification. Presented at the 2020 8th International Conference on Digital Home (ICDH), Dalian, China (2020)
90. Seunghyeok, O., Jaeho, C., Jong-Kook, K., Joongheon, K.: Quantum convolutional neural network for resource-efficient image classification: a quantum random access memory (QRAM) approach. In: 2021 International Conference on Information Networking (ICOIN) (2021). https://doi.org/10.1109/icoin50884.2021.9333906
91. Chen, S.Y.-C., Wei, T.-C., Zhang, C., Yu, H., Yoo, S.: Hybrid Quantum-Classical Graph Convolutional Network. arXiv: Machine Learning, no. arXiv:2101.06189 (2021). https://doi.org/10.48550/arXiv.2101.06189

92. Chen, S.Y.-C., Wei, T.-C., Zhang, C., Yu, H., Yoo, S.: Quantum convolutional neural networks for high energy physics data analysis. Phys. Rev. Res. **4**(1), 013231 (2022). https://doi.org/10.1103/PhysRevResearch.4.013231
93. Shijie, W., YanHu, C., ZengRong, Z., Gui Lu, L.: A quantum convolutional neural network on NISQ devices. AAPPS Bull. **32**(1), 1–11 (2022). https://doi.org/10.1007/s43673-021-00030-3
94. Hur, T., Kim, L., Park, D.K.: Quantum convolutional neural network for classical data classification. Quantum Mach. Intell. **4**(3), 18 (2022). https://doi.org/10.1007/s42484-021-00061-x
95. Chao-Han Huck, Y., et al.: Decentralizing feature extraction with quantum convolutional neural network for automatic speech recognition. Presented at the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 6523–6527 (2021)
96. Nielsen, M.A., Chuang, I.L.: Quantum Computation and Quantum Information, 10th Anniversary Edition. Cambridge University Press (2010). https://doi.org/10.1017/CBO9780511976667
97. Willsch, M., Willsch, D., Michielsen, K.: Lecture Notes: Programming Quantum Computers. arXiv :Quantum Physics (2022). https://doi.org/10.48550/arXiv.2201.02051
98. Schuld, M., Sinayskiy, I., Petruccione, F.: The quest for a quantum neural network. Quantum Inf. Process. **13**(11), 2567–2586 (2014). https://doi.org/10.1007/s11128-014-0809-8
99. James, S., Josh, I., Nathan, K., Giuseppe, C.: Quantum natural gradient. Quantum **4**(269), 15 (2020). https://doi.org/10.22331/q-2020-05-25-269
100. Ville, B., et al.: PennyLane: automatic differentiation of hybrid quantum-classical computations. arXiv: Quantum Physics (2018). https://doi.org/10.48550/arXiv.1811.04968
101. Bartkiewicz, K., Gneiting, C., Černoch, A., Jiráková, K., Lemr, K., Nori, F.: Experimental kernel-based quantum machine learning in finite feature space. Sci. Rep. **10**(1), 12356 (2020). https://doi.org/10.1038/s41598-020-68911-5
102. Sainadh, U.S.: An efficient quantum algorithm and circuit to generate eigenstates of SU (2) and SU (3) representations. arXiv: Quantum Physics 1309.2736, pp. 121 (2013). https://doi.org/10.48550/arXiv.1309.2736
103. Iordanis, K., Jonas, L., Anupam, P.: Quantum algorithms for deep convolutional neural networks. In: Eighth International Conference on Learning Representations. (2020). https://doi.org/10.48550/arXiv.1911.01117.
104. Egger, D.J., et al.: Quantum computing for finance: State-of-the-art and future prospects. IEEE Trans. Quantum Eng. **1**, 1–24 (2020). https://doi.org/10.1109/TQE.2020.3030314
105. Ryan, S., et al.: Stochastic gradient descent for hybrid quantum-classical optimization. Quantum (2020). https://doi.org/10.22331/q-2020-08-31-314
106. Szegedy, C., Toshev, A., Erhan, D.: Deep neural networks for object detection. Presented at the Proceedings of the 26th International Conference on Neural Information Processing Systems, vol. 2, Lake Tahoe, Nevada (2013)
107. Vedran, D., Jacob, M.T., Hans, J.B.: Quantum-enhanced machine learning. Phys. Rev. Lett. **117**(13), 130501 (2016). https://doi.org/10.1103/PhysRevLett.117.130501
108. Hochreiter, S.: The vanishing gradient problem during learning recurrent neural nets and problem solutions. Int. J. Uncertain. Fuzziness Knowl. Based Syst. **6**(02), 107–116 (1998). https://doi.org/10.1142/S0218488598000094
109. Arthur, P., Marco, C., Samson, W., Tyler, V., Andrew, T.S., Patrick, J.C.: Absence of barren plateaus in quantum convolutional neural networks. Phys. Rev. X (2021). https://doi.org/10.1103/physrevx.11.041011