Contents lists available at ScienceDirect

NDT and E International

journal homepage: www.elsevier.com/locate/ndteint

Research paper

Quantum machine learning for recognition of defects in ultrasonic imaging

Anurag Dubey [a,*], Thulsiram Gantala [b], Anupama Ray [c], Anil Prabhakar [d], Prabhu Rajagopal [a]

[a] Centre for Non-Destructive Evaluation, Department of Mechanical Engineering, Indian Institute of Technology Madras, Chennai, 600036, India
[b] Non-Destructive X Laboratory, Mechanical and Aerospace Engineering, Indian Institute of Technology Hyderabad, Telangana, 502285, India
[c] International Business Machines, New Delhi, 110044, India
[d] Department of Electrical Engineering, Indian Institute of Technology Madras, Chennai, 600036, India

ARTICLE INFO

Keywords:
Weld defects
Quantum computing
DCGAN
VQC
TFM
SelectKBest

ABSTRACT

The paper discusses a new paradigm of employing a quantum machine learning (QML) algorithm for automated weld defect recognition. A variational quantum classifier (VQC) using ultrasonic phased arrays is proposed to extract weld defect features in the atomic state to improve the classification accuracy and achieve high-speed calculation due to simultaneous qubits. The VQC is trained using a simulation-assisted weld dataset generated using finite element (FE) models and deep convolution generative adversarial networks (DCGAN). The total focusing method (TFM) weld images of porosity and slag are generated using time-transmitted signals received by performing full matrix capture, modeling various defect morphologies using FE simulations. These datasets are fed to train the DCGAN to generate synthetic TFM images. We use the feature selection method to obtain the best results with a quantum circuit with minimal qubits. Prominent features so obtained are supplied to the encoder circuit of the VQC to convert it to a quantum domain, thereby passing to an ansatz circuit to train quantum hyperparameters. The loss is computed for every iteration by optimizing the learnable parameters of the VQC. The VQC is trained by varying quantities of datasets to improve the reliability and efficiency of the weld defect classifications. It is found that VQC outperforms some of the classical machine learning algorithms with an accuracy of 96%.

1. Introduction

Phased array ultrasonic imaging is a widely used non-destructive evaluation technique that plays an important role in the performance qualification of critical components across industries [1,2]. Particularly, phased array ultrasonics is vital for the quality assessment of welds, and human domain experts typically evaluate the images generated. [3–6]. However, this process is cumbersome, time-consuming, and error prone. Therefore, artificial intelligence (AI) algorithms have been proposed for automated weld defect recognition to eliminate human error and help in quicker decision-making [7,8]. AI is known to perform better in feature extraction and image classification [9,10]. Convolutional neural networks (CNN) are used to analyze and classify weld images obtained from ultrasound phased array technique [8,11,12]. The CNN learns the pattern and features from the training dataset during training [13,14]. These trained networks are deployed for online monitoring of welds to perform automated defect recognition (ADR) [12].

In this paper, we present an original research study that explores the application of Variational Quantum Classifiers (VQC) for automated weld defect detection. We specifically compare VQC with three classical algorithms: multi-layer perceptron (MLP), autoencoders, and region-based convolutional neural networks (RCNN). MLPs are the fundamental type of feedforward artificial neural network where information flows from the input layers through hidden layers to output layers without any loops or feedback connections [15]. Autoencoders are neural networks used for classification tasks by utilizing the latent representations learned by the encoder. They can be trained to reconstruct data and the bottleneck layer in the encoder, which captures the compressed representation of the input and can be utilized as features for classification using a separate classifier layer [16]. By using autoencoders for classification, the network can learn to extract relevant features from the input data and reduce its dimensionality, leading to potentially improved classification accuracy, especially when dealing with high-dimensional or complex datasets. RCNN is a class of CNN that focuses on object detection and localization tasks by combining region proposals with deep learning models, typically using a two-stage architecture. The RCNN model consists of the selective search for region proposals, feature extraction using a pre-trained CNN, and a final classifier to detect objects within the proposed regions [17]. These algorithms are widely used for pattern recognition, classification, and

* Corresponding author.
E-mail address: anuragdubey569@gmail.com (A. Dubey).

https://doi.org/10.1016/j.ndteint.2024.103262
Received 15 February 2024; Received in revised form 3 September 2024; Accepted 20 October 2024
Available online 22 November 2024

regression tasks. However, they have their limitation as the training time increases with the increasing quantity of the training dataset. Generally, since the network is used for inspecting crucial components, the predicted accuracy should not be falsely positive, leading to catastrophic failures.

In order to develop a robust ADR system, a new data-driven methodology is necessary to speed up the computation and improve the efficiency in detecting and classifying weld defects. Unlike classical computers, which store and process information using bits (binary digits) that can have only the values of 0 or 1, quantum computers use qubits (short for quantum bits) that can exist in multiple states simultaneously. One of the key concepts in quantum computing is superposition, which refers to a qubit's ability to live simultaneously in multiple states [18]. For example, a qubit can exist in the forms '0' and '1' simultaneously, and the qubit's state is determined by a complex probability amplitude associated with each state. This allows a qubit to represent multiple pieces of information at once. Another key concept in quantum computing is entanglement, which refers to the phenomenon where the state of two or more qubits becomes correlated, such that the state of one qubit is dependent on the state of the other. This allows quantum computers to perform certain operations more efficiently than classical computers [19].

QML is a research area that explores the interplay of ideas from quantum computing and machine learning. QML combines the principles of QC with machine learning techniques. QML uses statistical methods and algorithms without explicit instructions to analyze the feature space of empirical data to identify and generalize patterns and insights. Here, we propose the use of VQC [20] for automated weld defect detection and classification. In this work, we have used a hybrid quantum–classical approach for training the VQC network. The proposed VQC network is trained on simulated TFM images of welds with defects such as slag and porosity. TFM is a technique of using the data from FMC to produce an image that is focused at every specified point in the image. TFM is implemented by applying an algorithm to the full data set collected by FMC. The training dataset is generated using two different approaches. In the first approach, full matrix capture (FMC) [7] configuration of phased array ultrasonics is performed in an FE solver to record the A-scan signals by modeling the weld's porosity and slag defect features. The TFM is employed on the received A-scan to generate the images [21]. In the second approach, the FE weld defect TFM images are fed to train the AI algorithm for generating faster TFM images as output [22]. The selectKBest methodology (explained in Section 3-A) is employed to extract dominated features from the weld TFM images supplied to the VQC network for training [21,23]. To obtain the optimal number of training datasets and improve the prediction accuracy, the VQC network is trained multiple times by increasing the quantity of the training dataset.

The objective of this research is to investigate the effectiveness of Variational Quantum Classifiers (VQC) in classifying weld defects, with a focus on identifying the advantages and limitations of quantum machine learning in comparison to classical methods. The study aims to make original contributions to the field of QML and its potential applications in non-destructive testing. Firstly, the effectiveness of Variational Quantum Classifiers (VQC) in classifying weld defects is assessed. This is an important area of study as weld defects can lead to structural failures and pose a risk to safety in various industries. By utilizing VQC, the accuracy and efficiency of weld defect classification can be improved. Secondly, the study investigates the impact of noise on the classification accuracy of VQC. The noisy and non-noisy weld datasets are analyzed separately to determine the potential limitations of VQC in noisy environments. This is a crucial aspect to consider, as many industrial environments can be noisy and may affect the performance of VQC. Thirdly, the results of quantum classification are compared with classical machine learning results to determine the potential advantages of quantum computing in industrial applications. The comparison provides insights into the strengths and limitations of each approach and highlights the areas where quantum computing can offer superior performance. Finally, the applicability of VQC deployment for industrial usage is checked. This involves evaluating the feasibility of implementing VQC in real-world industrial environments. Our research contributes to the advancement of QML in non-destructive testing by providing new insights into the application of quantum algorithms for weld defect detection. This work lays the groundwork for further exploration of quantum computing's potential in industrial applications, beyond the scope of a review.

The overall brief process to classify the weld dataset using VQC is shown in Fig. 1, where initially, the image goes through a feature selection method to extract the best three features, which then goes through VQC for classification.

This paper is organized as follows: Section 2 describes the model structure of VQC and the data generation process. Section 3 discusses training the VQC classifier. Results and analysis are given in Section 4, after which the paper concludes an outlook on further work in Section 5.

2. Background - quantum machine learning model used for training

This section explains how weld defects prominent features are extracted using the selectKBest methodology, quantum, and classical classifiers used for training.

2.1. Feature selection using SelectKBest

SelectKBest transforms data from a high-dimensional space to a lower-dimensional space while retaining as much of the original information as possible [24]. To obtain the best results with a minimum number of qubits, we reduced the feature size from 784 to 3, which equals the number of qubits we used for training the VQC. This feature selection algorithm uses the F-test to select the top K features most relevant to predicting the target variable in a classification problem. The F-test is a statistical test that evaluates the linear relationship between the feature and the target variable. It computes the ratio of the between-group variability to the within-group variability. The between-group variability is the variability between the means of the target variable for each class, and the within-group variability is the variability within each class. The algorithm evaluates each feature individually using the F-test and then ranks the features based on their F-scores. The higher the F-score, the more relevant the feature is to the target variable.

SelectKBest can be used for various purposes, such as reducing the dimensionality of data, identifying patterns in data, and visualizing high-dimensional data. In machine learning, it is often used as a preprocessing step to reduce the number of features in a dataset and improve models' performance.

2.2. Classification with variational quantum classifier model

We imported the weld dataset of porosity and slag from the images generated through FE and DCGAN. The image is divided into squares of 784 × 784 pixels converted to comma-separated values. The output pixels are converted into NUMPY values, and the selectKBest feature extraction method is applied to select the best three features. This feature-extracted dataset is supplied to our VQC model. The unitary gates U(θ) are first applied to the initial states of the quantum circuit, $|0>^n$ in a VQC. A variational circuit block W(θ) is added, tuned, tested, and optimized. The circuit is then measured on a Z-basis. The result is a string of bits, z ϵ $[0,1]^n$. The bit string is then mapped to the cost function C. The labels from the circuit are compared to the actual labels. The optimizer feeds a fresh set of parameters to W(θ). The expectation value $P_z = \langle\phi(x)|W^{\dagger}(\theta)M_z W^{\dagger}(\theta)|\phi(x)\rangle$ is estimated by re-running the circuit again. These values are again mapped to C ϵ [+1,

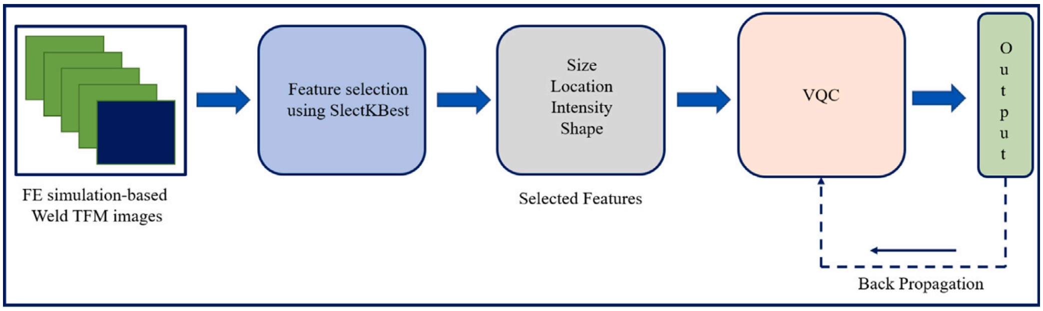

Fig. 1. Schematic diagram illustrating an overall weld defect classification process using VQC.

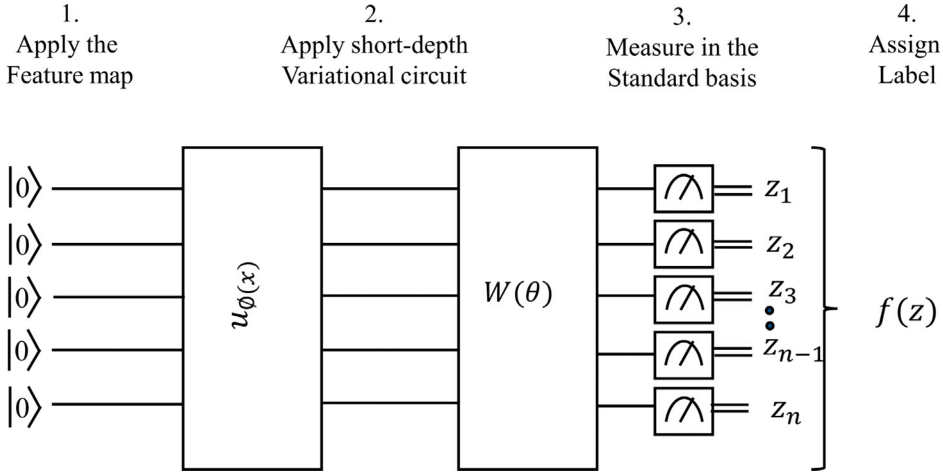

Fig. 2. Schematic diagram showing the proposed VQC Circuit after feature extraction of our dataset.

−1] until the ideal hyperparameters are discovered. To evaluate the algorithm's performance, test data is applied by fixing the parameters of W(θ). We have utilized the statevector simulator provided by IBM's Qiskit framework. The statevector simulator is a high-performance classical simulator that accurately models the behavior of a quantum system by representing the quantum state as a complex vector in a high-dimensional Hilbert space. This allows us to simulate the quantum circuit operations as if they were running on actual quantum hardware, enabling us to evaluate the performance of VQC in the absence of real quantum hardware. Given the current limitations of quantum hardware, including issues such as qubit coherence time, gate errors, and limited qubit counts, the use of a high-fidelity simulator is crucial for developing and testing quantum algorithms. The statevector simulator allows us to explore the potential of VQC without being constrained by the hardware's current limitations. The schematic diagram of this proposed VQC model is shown in Fig. 2 [25].

2.2.1. Advantages of using variational quantum classifiers (VQC) over classical methods

In our study, we propose the use of Variational Quantum Classifiers (VQC) for the automated detection and classification of weld defects, and we compare its performance with traditional machine learning methods like multi-layer perceptrons (MLP), autoencoders, and region-based convolutional neural networks (RCNN). Below, we outline the key advantages of using VQC over these classical methods:

- **Quantum Superposition and Parallelism:** Classical algorithms process information in a sequential or parallel manner, but they are fundamentally limited by the linearity of their processing units. VQC leverages the principle of quantum superposition, allowing a qubit to exist in multiple states simultaneously. This means that VQC can explore multiple solutions or patterns at once, potentially leading to faster convergence and a more efficient exploration of the feature space. This parallelism can be particularly advantageous in high-dimensional datasets, where classical methods may struggle with computational complexity.
- **Entanglement and Feature Correlation:** Classical models like MLP and autoencoders rely on manually engineered features or simple feedforward architectures that may not fully capture complex correlations between features. VQC uses quantum entanglement to create correlations between qubits that represent different features. This allows the quantum circuit to naturally capture and exploit intricate relationships within the data, which might be difficult for classical models to learn without extensive tuning or deeper architectures.
- **Robustness to Overfitting:** Classical models are prone to overfitting, especially when dealing with small datasets or noisy data, which often requires the use of regularization techniques and careful model selection. The inherent randomness and probabilistic nature of quantum measurements can introduce a form of regularization, potentially making VQC more robust to overfitting. Additionally, quantum circuits may inherently generalize better due to their ability to explore a broader solution space, even with fewer parameters than their classical counterparts.
- **Scalability for Complex Industrial Applications:** Scaling classical models to meet the demands of complex industrial applications often involves increasing model complexity, which can lead to diminishing returns and higher computational costs. VQC's ability to scale with qubit resources, without necessarily increasing model depth or complexity in a linear fashion, offers a promising avenue for tackling large-scale, complex industrial problems like weld defect detection. As quantum hardware improves, the scalability of VQC could surpass classical methods in both performance and efficiency.

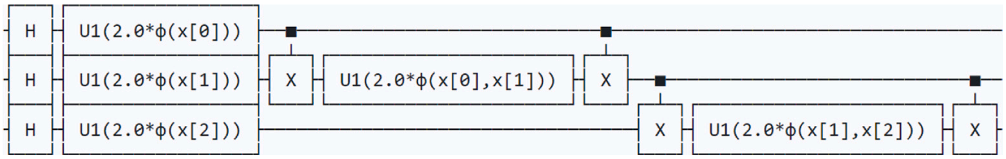

Fig. 3. Schematic diagram showing ZZ Feature Map Circuit.

2.2.2. Challenges and limitations of training VQC

Current quantum hardware is limited by factors such as qubit coherence time, gate fidelity, and noise. These factors can affect the stability and reliability of the VQC training process. The hybrid quantum–classical optimization process is sensitive to the initial conditions and can suffer from issues such as vanishing gradients and local minima. This requires careful tuning of the optimization parameters and the choice of classical optimizers. As the number of qubits increases, the complexity of the quantum circuit and the corresponding classical optimization process also increase. This scalability issue is a significant challenge that we aim to address in future iterations of our research, particularly as quantum hardware continues to evolve.

Using FE simulation and DCGAN, we have 1000 training weld images with porosity and slag-type defects. The optimal number of training datasets is essential for any machine learning network because the model training time partially depends upon the quantity of the images. Therefore, to minimize the training time and improve the prediction accuracy, we randomly selected two batches of 200 and 400 different images from 1000 images to train various classifier models.

3. Methodology

3.1. Model structure of VQC

The VQC pre-processes the weld classification data to prepare it in the quantum state using an encoder circuit. The preprocessed data is then fed to an ansatz layer for training, after which the variational parameters are optimized to minimize the classification error. The detailed components of VQC to train the model are model below:

3.1.1. Encoder circuit

An encoder circuit is a part of the variational circuit that prepares the input data in a quantum state, which the remaining layers can then process [26,27]. The encoder circuit typically takes classical input data, such as a vector of numerical features, and maps it to a quantum state. This mapping can be done using a variety of quantum gates, such as rotation gates, entangling gates, or other gates designed to transform the input data into a form that the quantum circuit can more easily process. The choice of gates used in the encoder circuit depends on the problem being solved and the available computational resources. The goal of the encoder circuit is to map the input data to a quantum state that can be used as the input to the remaining layers of the variational circuit. The parameters of the encoder circuit are typically learned during the training of the VQC, along with the parameters of the other layers in the circuit, using classical optimization techniques.

The numerical experiments described in this paper run with the ZZ FeatureMap encoder, a second-order Pauli-Z evolution circuit [26]. It is effective in various problems, including image classification and financial data analysis. A schematic diagram of the three-qubit, single repetition, and linearly entangled ZZ FeatureMap circuit used is shown in Fig. 3.

3.1.2. Ansatz

The ‘ansatz’ is a part of the variational circuit responsible for performing a series of quantum operations on the input data in order to enable better classification [28]. This circuit in VQC is typically composed of layers of quantum gates, such as single-qubit rotations and two-qubit entangling gates, which can be repeated multiple times to increase the depth and complexity of the circuit. During the training phase, the parameters of the ansatz are optimized using classical optimization techniques, such as gradient descent, to minimize the cost function, which is a measure of the difference between the predicted and the actual class labels of the input data [29]. The optimization is typically performed using a classical computer, which interacts with the quantum computer to evaluate the circuit and update the parameters. The schematic diagram of the parameterized ansatz circuit illustrated in Fig. 4 is used in the experiments performed in the current analysis.

3.1.3. Optimizer

Supervised learning involves finding the parameters that minimize the difference between the predicted and actual output. The optimization algorithm iteratively adjusts the model’s parameters using backpropagation to reduce the loss function. A well-optimized model can make accurate predictions on new, unseen data, while a poorly optimized model may overfit the training data and perform poorly on new data. For our experimental purpose, we have used a COBYLA optimizer [30].

3.1.4. Implementation details of VQC

In our study, the VQC is implemented using a hybrid quantum–classical approach, combining the power of quantum circuits with classical optimization techniques. Below, we provide the key details of the VQC implementation:

- Quantum Circuit Design: The VQC is built upon a quantum circuit architecture designed to process the feature vectors extracted from ultrasonic imaging data. The number of qubits in the circuit corresponds to the dimensionality of the feature space. Each qubit represents a feature in the input data. The circuit employs parameterized quantum gates RX and X gate to manipulate the qubits for our work. These gates are the building blocks of the variational circuit, and their parameters are adjusted during the training process. Entanglement is introduced between qubits using controlled gates, allowing the circuit to capture correlations between features.
- Hybrid Training Process: The training of the VQC involves a hybrid quantum–classical workflow. The input data is encoded into the quantum circuit, and the circuit is executed on a quantum simulator to compute the output probabilities. These probabilities are used to calculate the loss function, which reflects the classification error. The gradients of the loss function with respect to the circuit parameters are computed using a classical optimizer (COBYLA for our work). These gradients guide the update of the quantum gate parameters to minimize the loss.
- Parameter Initialization and Optimization: The parameters of the quantum gates are initialized randomly or based on a heuristic. The optimization process is carried out iteratively, where the classical optimizer adjusts the parameters to gradually improve the model’s performance. The hybrid nature of the training allows leveraging classical computational resources while exploring the quantum-enhanced feature space.

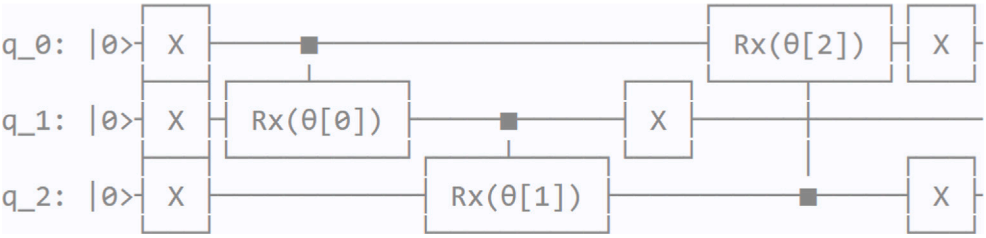

Fig. 4. Illustrative diagram showing an example of an ansatz circuit.

3.2. Data generation

3.2.1. Bulk FE simulation-based weld TFM dataset creation

The TFM is an advanced post-processing technique used in ultrasonic testing (UT) to improve imaging resolution and enhance defect detection capabilities. TFM leverages the data acquired through techniques like FMC to focus ultrasonic waves at specific points within the inspected material, resulting in superior imaging quality and accurate defect localization [21]. The raw N × N A-scan matrix is created using the FMC technique by transmitting each phased array element and receiving the reflected signals through each phased array element(where N is the number of elements in the phased array). The region of inspection is discretized into the number of grids in the 2-D x–z plane that serve as the pixel locations after the FMC signal has been obtained. The TFM determines the time of flight (TOF) between each region of inspection pixel and the positions of the transducer and receiver, as shown in Fig. 5. The delay law is applied to the received transducer elements to ensure that each element receives the signal at the same time. The resulting amplitude is added to determine the intensity $I_1(p)$. To create the intensity map predicted by Eq. (1), this procedure is performed for all transmission cycles and pixel positions in the region of inspection. However, the periodic mobility of the received signals causes additional imaging artifacts to appear in the TFM image. Therefore, the TFM envelope is calculated by taking an imaginary part of the Hilbert-transformed standard raw A-scans $M_{uv}(t)$ and the norm of the two TFM pictures, one from the standard raw A-scan $A_{uv}(t)$, and the other from the standard raw A-scan $A_{uv}(t)$ [22]. Eq. (2) provides the intensity of the TFM envelope $I_2(p)$.

$$I_1(p) = |\sum_{u=1}^{N}\sum_{v=1}^{N} A_{uv}(t_u^p + t_v^p)| \tag{1}$$

$$I_2(p) = |\sum_{u=1}^{N}\sum_{v=1}^{N} A_{uv}(t_u^p + t_v^p) + \sum_{u=1}^{N}\sum_{v=1}^{N} M_{uv}(t_u^p + t_v^p)| \tag{2}$$

here,

$$t_u^p = \frac{d_{D_u}^{(t)}}{C_u^{(t)}} = \frac{\sqrt{(x_t - x_p)^2 + z_p^2}}{C_u^{(t)}} \tag{3}$$

$$t_v^p = \frac{d_{D_v}^{(r)}}{C_v^{(t)}} = \frac{\sqrt{(x_p - x_r)^2 + z_p^2}}{C_v^{(r)}} \tag{4}$$

where $A_{uv}(t)$ and $M_{uv}(t)$ are the FMC signal from raw A-scans and Hilbert-transformed standard raw A-scans. The uth transmission element and the vth receiving element with a distance of $d^t_{D_u}$ and $d^r_{D_v}$ from the pixelpoint P(x,z) to the transmitter and receiver array respectively. $C_u^{(t)}$ and $C_v^{(r)}$ are the sound velocities for transmission and reception.

In order to generate high-quality weld datasets, a series of steps must be taken. The first step in this process is to create defective samples in a laboratory setting. These samples are then subjected to FMC-TFM using A-scan signals to produce TFM images. A commercial FE Abaqus/Explicit (v. 18.0) solver is used, as outlined in the ABAQUS User Manual v. 6.11, Dassault system, Providence, RI, USA. With these TFM images in hand, finite element (FE) simulation is used to model wave propagation in the weld specimen [22]. This simulation generates the FE-based weld TFM imaging dataset using suitable materials properties and dimensions. To ensure the dataset is broad and varied, artificial weld defect parameters are introduced into the models. Flaw shape, size, and orientation are all manipulated to create a large weld defect dataset. This is done using computer-aided design models. 515 porosity-based and 364 slag-based computer-aided design models are prepared for FE simulations to generate the synthetic weld TFM images.

FE-assisted weld TFM images for porosity and slag are shown in Fig. 6 and Fig. 7, respectively. Generating a single weld TFM image takes around 5 h, which is very time-consuming. We have used DCGAN, an AI-based CNN elaborated on in the next section, to overcome this issue.

3.2.2. Artificial intelligence-based data generation using DCGAN

DCGANs are based on the generative adversarial networks (GANs) framework, which consists of two main components: a generator network and a discriminator network. The generator network synthesizes new samples, such as images, while the discriminator network learns to distinguish between real and fake samples. DCGANs specifically employ CNN architectures for both the generator and discriminator networks [31]. The generator network in DCGAN consists of convolutional layers, followed by upsampling layers (e.g., transposed convolutions) to gradually increase the spatial dimensions of the input noise vector and generate realistic images. Batch normalization layers are often used to stabilize training. The discriminator network is a CNN that typically consists of convolutional layers followed by downsampling layers (e.g., max pooling) to reduce the spatial dimensions and classify images as real or fake. The training process of DCGANs involves a competitive game between the generator (G) and discriminator (D) networks. The generator aims to generate realistic samples that can deceive the discriminator while the discriminator tries to classify real and fake samples correctly. The training is performed iteratively, where the generator and discriminator networks are updated in an alternating fashion through mutual optimization of hyperparameters by employing a backpropagation algorithm. The complete GAN training process is illustrated in Fig. 8. The standard normal distribution is used to sample the latent space vector $z^1 \ldots z^n$. G(z), a generator function, maps the latent vector z to the image space of random distribution $P_g(z)$. In order to produce fake images that are reasonably near to real images, the generator (G) seeks to estimate the distribution of the real dataset $P_{data}(x)$. The job of the discriminator (D) is to use binary classifications of 0 or 1 to distinguish between real and fake images. The probability $D(x)$ ϵ [0, 1] is produced when we feed real samples to discriminator (D). At the same time, when we supply fake samples to discriminator (D) generated by generator (G), it produces the probability of $D(G(z))$ ϵ [0, 1].

Generator and discriminator neural networks are trained in an adversarial manner to generate data by mimicking some real distribution G(z)$\simeq P_g(z)$ which is close to real dataset distribution $D(x) \simeq P_{data}(x)$. The loss function used to train the DCGAN for all datasets is shown in Eq. (5).

$$\min_G \max_D V(D,G) = E_{x\sim P_{data}(x)}\,[log D(x)] + E_{z\sim P_g(z)}\,[log(1 - D(G(x)))] \tag{5}$$

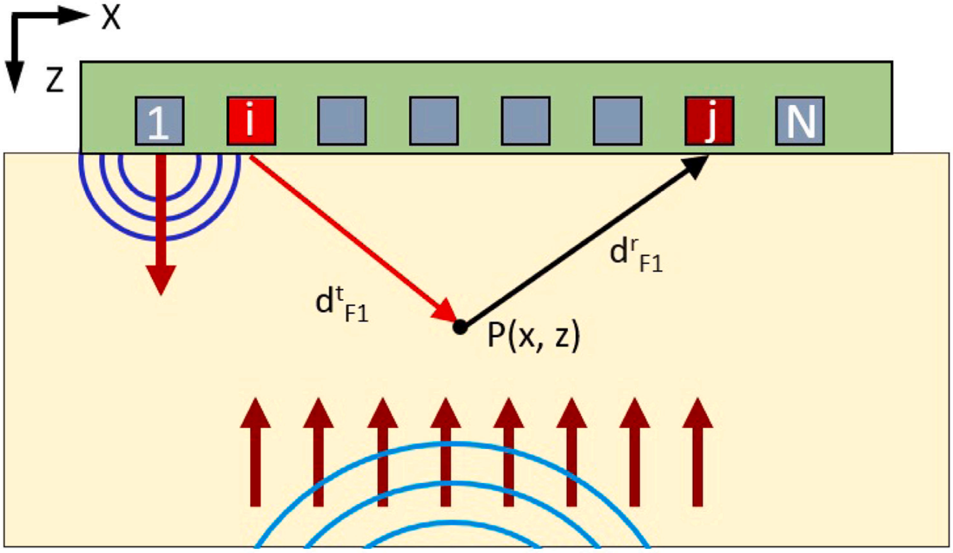

Fig. 5. The schematic diagrams illustrate the test sample scanning strategy Full Matrix Capture (FMC) and Total Focusing Method (TFM).

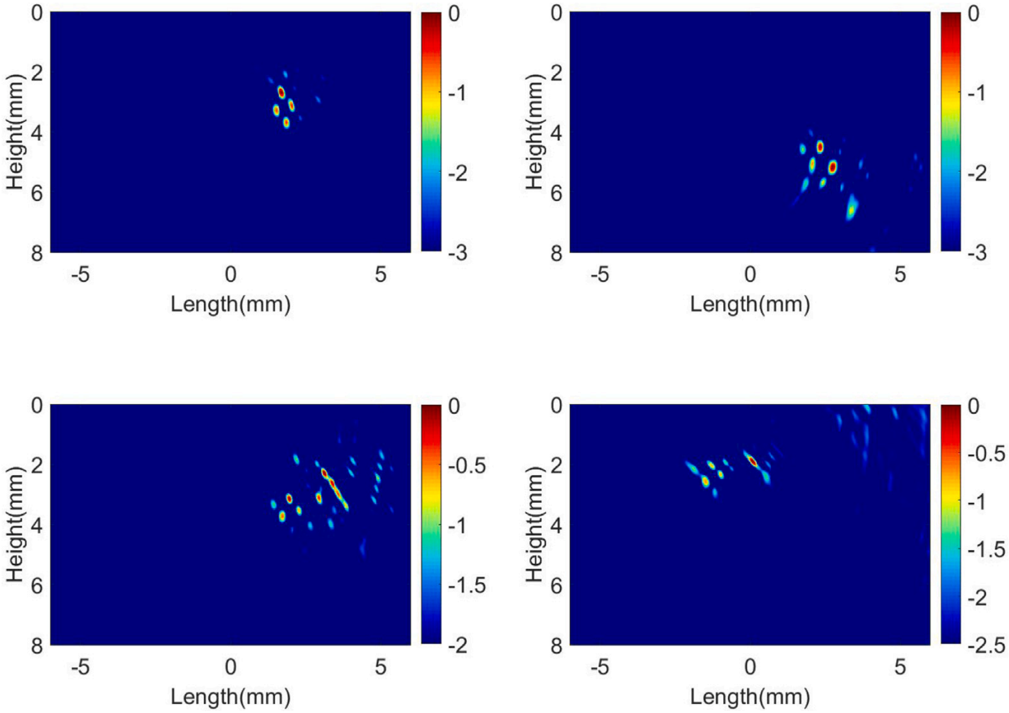

Fig. 6. FE simulation-based weld TFM images of porosity type defect in weld sample.

Stability is one of the most common challenges in training Generative Adversarial Networks (GANs). Due to the complex nature of the GAN architecture, training can often become unstable, leading to poor-quality generated samples. Deep Convolutional GANs (DCGANs) have been developed to address this issue, incorporating several techniques to stabilize the training process and improve the quality of the generated samples. One such technique is Batch Normalization. By normalizing the inputs to each layer, batch normalization helps reduce internal covariate shifts, leading to a faster and more stable training process. This technique is particularly useful in GANs, where the generator and discriminator networks are trained together in an adversarial setting. Additionally, DCGANs often use alternative convolutional layers for upsampling in the generator. Fractional-strided convolutions (transposed convolutions or deconvolutions) are used instead of traditional upsampling techniques to avoid checkerboard artifacts. These artifacts are caused by traditional upsampling techniques, which can lead to a distorted image with visible checkerboard patterns. Another technique used in DCGANs is avoiding fully connected layers in favor of convolutional layers with appropriate spatial dimensions to preserve spatial information and improve the generation process.

The training of DCGAN is done by supplying FE-based weld TFM images to its structure. All these datasets are grayscale images with pixel intensity values ranging between [0, 255], then normalized to −0.5 to +0.5 pixel values. Determination of trained network efficiency is done by analyzing discriminator (D) and generator (G) loss values. The discriminator loss decreases over the iterations to stabilize its

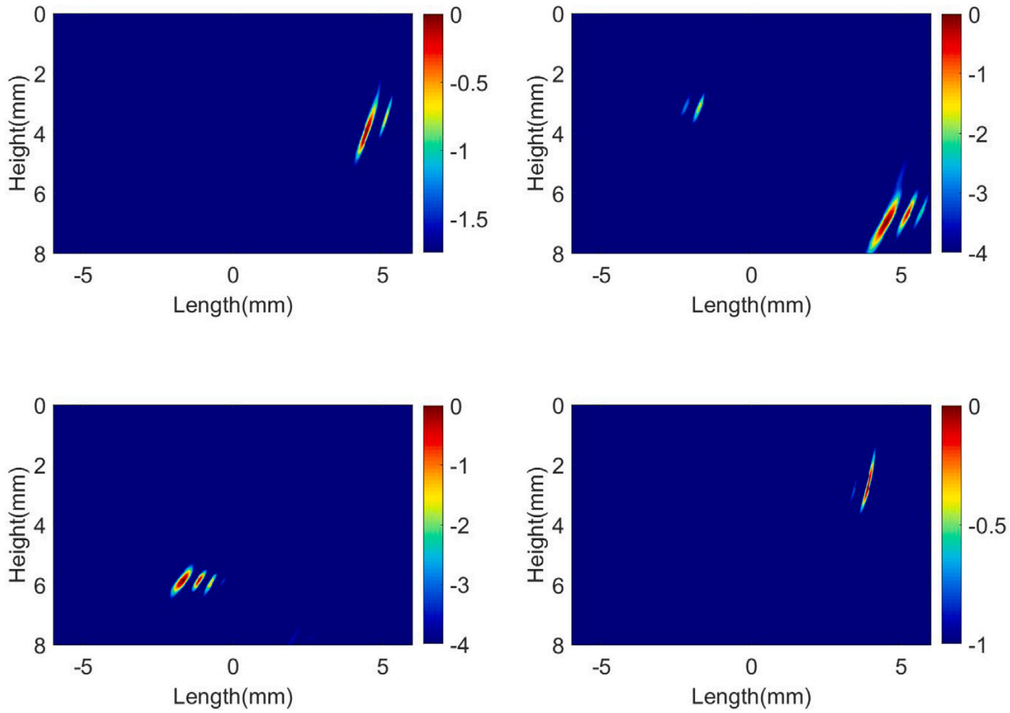

Fig. 7. FE simulation-based weld TFM images of slag type defect in weld sample.

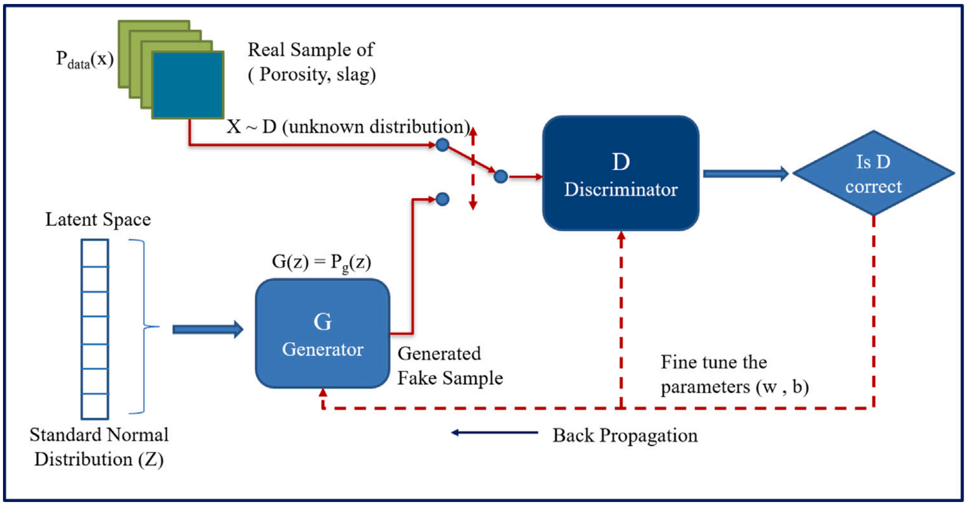

Fig. 8. An illustrative flowchart of GAN for generating synthetic weld TFM images.

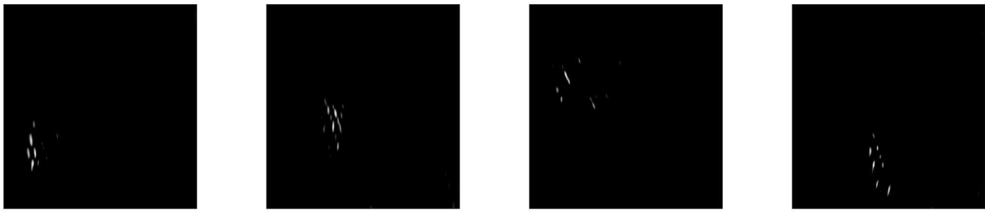

Fig. 9. DCGAN generated synthetic weld TFM images of porosity type defect in weld sample.

model, whereas the generator loss shoots up to adjust its hyperparameters [22]. Generator and Discriminator competition helps them improve their network by generating synthetic weld TFM images which are then compared with FE simulation-based TFM datasets. Porosity and slag images generated using DCGAN are shown in Figs. 9 and 10.

3.2.3. Noisy dataset generation

Noise is extracted from experimental TFM images and added to FE and DCGAN simulated images to generate weld TFM noisy images. Extracting noise from experimental images and incorporating it into simulated images involves several steps. First, the plate is scanned

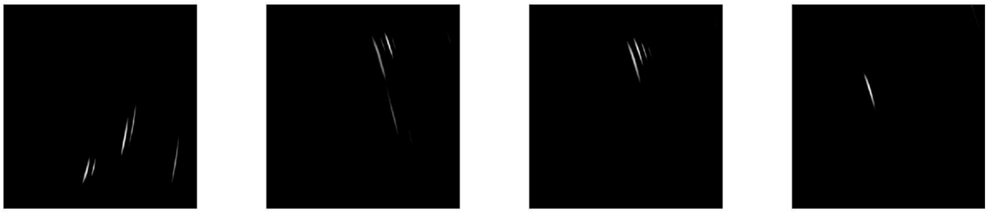

Fig. 10. DCGAN generated synthetic weld TFM images of slag type defect in weld sample.

Table 1
The component of confusion matrix: two weld defects - porosity and slag.

Actual	Predictions on weld defects	
	Porosity	Slag
Porosity	True positive (TP)	False negative (FN)
Slag	False positive (FP)	True negative (TN)

Table 2
Classification metrics for 200 and 400 MLP trained porosity and slag images tested on 500 images.

		Porosity	Slag	Precision	Recall	F1-score
200 batch	Porosity	180	70	0.66	0.72	0.69
	Slag	94	156	0.69	0.62	0.65
Overall						0.67
400 batch	Porosity	205	45	0.70	0.82	0.76
	Slag	87	163	0.78	0.65	0.71
Overall						0.73

along the weld area length to produce the weld TFM (Total Focusing Method) images, which are then stored in a folder. A commercial software reads these images one at a time using a predefined function. A sliding window approach is used to extract the noise from the experimental images. This involves defining the kernel size and stride values for the window's travel across the image. We use a kernel size of 5 × 5 and a stride value of '1'. This process creates a 2D array of the same size as the measured image to store the extracted noise. After the kernel finishes its travel across the measured image, the noise value for the entire image is obtained. Once the noise is extracted from the experimental images, simulated weld TFM images are read in commercial software. The noise array is then added directly to the simulated images. This procedure allows the simulated images to accurately mimic the experimental images' noise. Extracting noise from experimental images and adding it to simulated images is essential for accurately simulating real-world scenarios. This process is repeated to add the noise in all simulated images, including FEM and DCGAN images.

The experimentally measured weld TFM image of slag is created with 8 dB noise amplitude. The original FE-simulated images have 6 dB amplitude, hence to compare it with experimental TFM images, it is imaged into 8 dB amplitude after adding noise to it. It is challenging to match the simulated image with noise closely to the experimental image due to imaging artifacts, These imaging artifacts are inherent to the measured images because of various reasons such as non-uniformity of material properties, manufacturing processes, component geometry, weld area corner echos, defect scattering, instrumental noise, skilled operator, etc.

4. Results and analysis

4.1. Evaluation metrics for classification

The network trained on quantum and classical models is evaluated using the classification matrix. The classification matrix helps us to measure the number of weld defects misclassified using trained algorithms. The confusion matrix for two weld defect classes is shown in Table 1. It consists of four components: true positive (TP), true negative (TN), false positive (FP), and false negative (FN).

TP means our algorithm predicts an image as porosity, which is actually from the porosity class. FN means the algorithm predicts the image as slag, but it belongs to the porosity class. TN means the algorithm predicts the image as slag and belongs to the slag class. FN means the algorithm predicts the image as porosity but is from the slag class.

Evaluating metrics used for calculating the accuracy of the classification system are precision, recall, and F1-score. Precision measures the false positive rate, i.e., how many predictions made by the algorithm were correct. Whereas recall tells about false negatives, i.e., how accurately the algorithm measures all the positives. F1-score is the harmonic mean of precision and recall [32]. The precision, recall, f1-score, and accuracy are calculated as shown in Eq. (6), Eq. (7), Eq. (8), Eq. (9).

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{6}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{7}$$

$$\text{F1-Score} = \frac{2.\text{Precision.Recall}}{\text{Precision+Recall}} \tag{8}$$

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FN} + \text{TN} + \text{FP}} \tag{9}$$

4.2. Results without adding noise to dataset

4.2.1. Classical results using multi-layer perceptron

To compare with classical machine learning, the 200 and 400 batches of weld TFM images are used to train the multi-layer perception (MLP), a classical feed-forward artificial neural network. This network consists of three hidden layers of the ReLU function and a softmax layer. After training the model, Table 2 presents the classification matrix for both sets of training batches. The loss value for both batches of training images has been shown in Fig. 11. The loss value for the 200 and 400 batches of weld images settles down to 0.00159 and 0.00328, respectively, after 100 epochs. The general observations from both tables are as follows:

- While testing the simulated weld TFM images trained with 200 images out of 250 images of porosity and slag each, the trained network detected 180 images of porosity and 156 images of slag correctly.
- When the training images increased to 400, out of 250 images of porosity and slag each, the trained network detected 205 images of porosity and 163 images of slag correctly.

Using the classification metrics from Table 2 using precision and recall, overall classification accuracy for 200 and 400 batches is 67% and 73%, i.e., as the dataset size increases, classification accuracy increases.

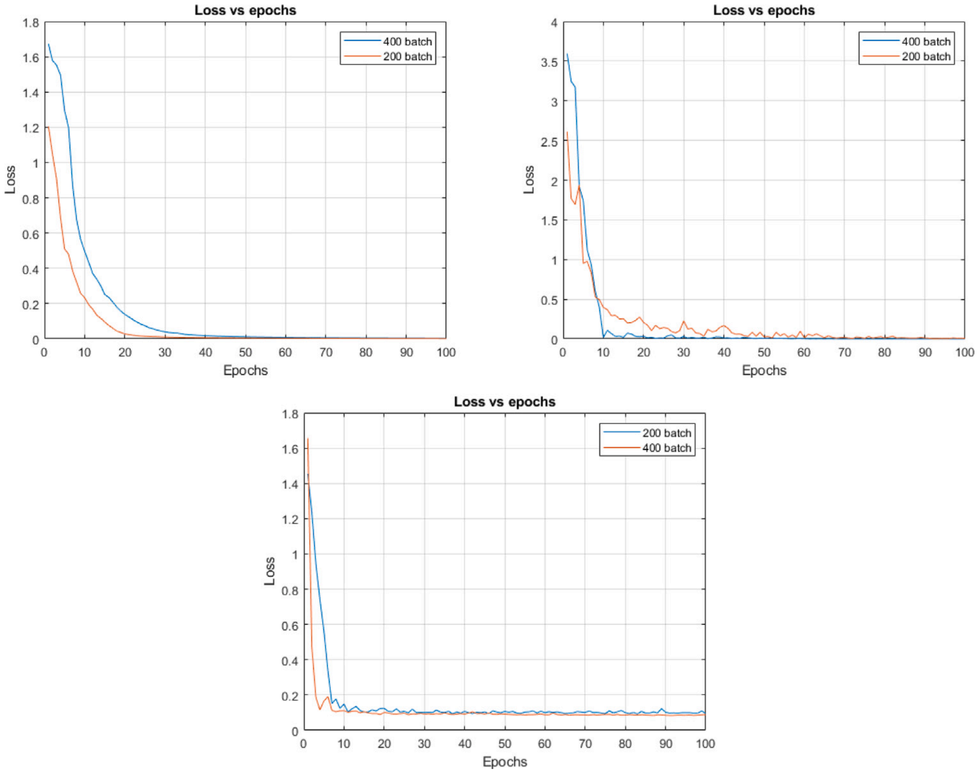

Fig. 11. Training loss for 200 and 400 batches of images of porosity and slag using MLP, Autoencoders and RCNN.

Table 3
Classification metrics for 200 and 400 autoencoder trained porosity and slag images tested on 500 images.

		Porosity	Slag	Precision	Recall	F1-score
200 batch	Porosity	204	46	0.78	0.82	0.80
	Slag	58	192	0.81	0.77	0.79
Overall						0.79
400 batch	Porosity	217	33	0.82	0.87	0.84
	Slag	48	202	0.86	0.81	0.83
Overall						0.83

4.2.2. Classical results using autoencoder

The autoencoder is also trained on the same dataset of 200 and 400 batches of weld TFM images. The encoder and decoder levels consist of two Batch normalization and leaky relu layers. Adam optimizer and mean squared error loss function have been used to train the model. After training the model, Table 3 presents the classification matrix for both sets of training batches. The loss value for both batches of training images is shown in Fig. 11. The loss value for 200 and 400 batches of weld images settles down to 0.00423 and 0.00006, respectively, after 100 epochs. The general observations from both tables are as follows:

- While testing the simulated weld TFM images trained with 200 images out of 250 images of porosity and slag each, the trained network detected 204 images of porosity and 192 images of slag correctly.
- When the training images increased to 400, out of 250 images of porosity and slag each, the trained network detected 217 images of porosity and 202 images of slag correctly.

Using the confusion matrix from Table 3 after calculating precision and recall, overall classification accuracy for 200 batches is 79% and 83%, i.e., as the size of the dataset increases, classification accuracy increases.

4.2.3. Classical results using region-based convolutional neural network

We have trained the RCNN model using our weld TFM dataset and performed a series of experiments to evaluate its performance. The model is trained on a high-performance GPU and utilizes a pre-trained CNN as the backbone to extract features from the input images. The RCNN model is trained using a combination of labeled data and region proposals generated by the selective search algorithm. Fig. 12 shows a bounding box and selective search applied on one training image to create regions to extract prominent features.

We can see from Fig. 12 that the selective search algorithm has formed regions of potential object detection. Training the network is done using Adam optimizer and binary cross entropy loss function. The batch size for each epoch has been kept constant at 64, and training has been done for 100 epochs. The loss values for both 200 and 400 batches of weld TFM images have been shown in Fig. 11, which nearly touches zero, which indicates the model has very effectively learned the pattern in images, meaning the model makes the predictions that closely matches the true values. Table 4 presents the classification metrics for both sets of training batches. The general observations from both tables are as follows:

- While testing the simulated weld TFM images trained with 200 images out of 250 images of porosity and slag each, the trained network detected 221 images of porosity and 219 images of slag correctly.
- When the training images increased to 400, out of 250 images of porosity and slag each, the trained network detected 230 images of porosity and 226 images of slag correctly.

Using the confusion matrix from Table 4, after calculating precision and recall, overall classification accuracy for 200 batches is 88% and 91%, i.e., as the size of the dataset increases, classification accuracy increases.

From the above figures and tables, some general observations can be made:

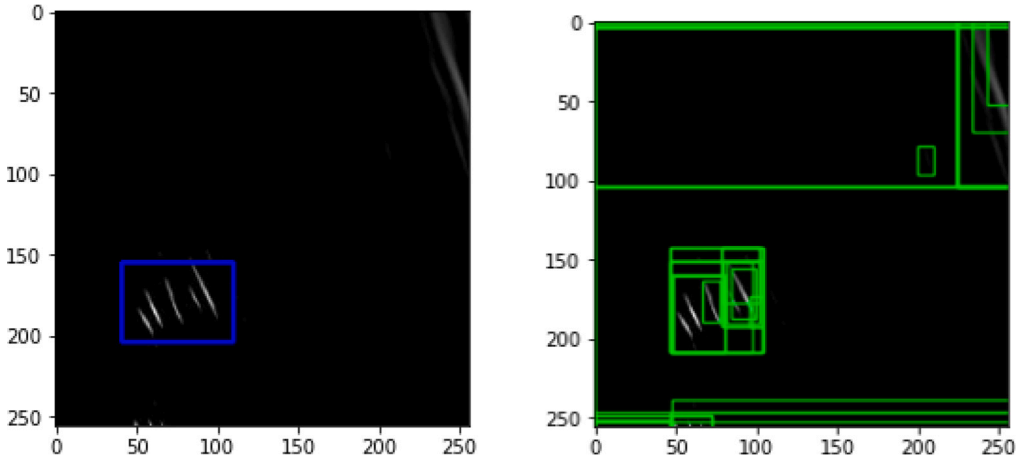

Fig. 12. Example image of welds defects showing the bounding box and selective search applied on one of the training images.

Table 4
Classification metrics for 200 and 400 RCNN trained porosity and slag images tested on 500 images.

		Porosity	Slag	Precision	Recall	F1-score
200 batch	Porosity	221	29	0.88	0.88	0.88
	Slag	31	219	0.88	0.88	0.88
Overall						0.88
400 batch	Porosity	230	20	0.90	0.92	0.91
	Slag	24	226	0.92	0.90	0.91
Overall						0.91

Table 5
Classification metrics for 200 and 400 VQC trained porosity and slag images with three features tested on 500 images with 50 shots.

		Porosity	Slag	Precision	Recall	F1-score
200 batch	Porosity	200	50	0.75	0.80	0.77
	Slag	63	187	0.79	0.73	0.76
Overall						0.77
400 batch	Porosity	220	30	0.80	0.88	0.84
	Slag	55	195	0.87	0.78	0.82
Overall						0.83

- When trained on VQC, the weld defect detection accuracy increases as the number of shots increases and the size of the training dataset increases.
- Comparing among classical networks, RCNN has the highest accuracy among the three classical models discussed in the paper.
- Quantum classifier VQC has more accuracy than all classical networks discussed in the paper, with an overall accuracy of 97% for 400 batches of weld TFM images trained at 2000 shots.

Hence, it is evident that the quantum classifier is advantageous over classical algorithms for weld TFM image detection. In the next section, we will introduce noise in the weld images and check the prediction accuracy of similar algorithms discussed above in the paper.

4.2.4. Quantum results using variational quantum classifier

The proposed VQC model is implemented in a quantum–classical hybrid manner. The two trained VQC model is validated on the remaining testing datasets. Training has been done using the IBMQ QASM simulator with COBYLA optimizer and two local feature maps on the International Business Machines quantum machine learning platform. The trained VQC models are tested on 500 images of porosity and slag (250 each) supplied in one batch to evaluate the effectiveness of weld defect classification prediction. The output of the VQC is to classify each image with a probability of being porosity or slag. The key to our approach is to minimize the loss function, thereby reducing the difference between the predicted and the actual output. The loss values for the 200 and 400 batches of training images for 50 and 2000 shots are shown in Fig. 13, respectively. From the plots, it is clear that the loss value for both subsets has dropped below 0.1. Loss values for 200 and 400 training set images after 100 iterations are 0.0233 and 0.0791, respectively. The decrease of these values to near zero clearly shows that the predicted output is close to the actual output. These predictions on weld defects create the ‘confusion matrix.’

After training the model, Table 5 presents the evaluation metrics at 50 shots for both training batches. The general observations from both tables are as follows:

- While testing the simulated weld TFM images trained with 200 images out of 250 images of porosity and slag each, the trained network detected 200 images of porosity and 183 images of slag correctly. When the training images increased to 400, out of 250 images of porosity and slag each, the trained network detected 220 images of porosity and 195 images of slag correctly.
- Precision is calculated using the definition explained in Section 4.1. For example, in Table 5, for 200 trained weld TFM images, out of 250 porosity images, VQC predicts 200 images as porosity, and out of 250 slag images, 67 images are predicted as porosity, but those are slag. So precision is 200/(200+67)= 0.75.
- Recall is also calculated using the definition from Section 4.1. For example, in Table 5, for 200 trained weld TFM images, out of 250 porosity images, VQC predicts 200 images as porosity and 50 images as slag. So recall is 200/(200+50) = 0.80.
- As we increased the training weld images from 200 to 400, the number of correctly classified images increased, increasing the overall classification accuracy.

Table 6 presents the evaluation metrics at 2000 shots for 200 and 400 images of training batches. The model trained with 200 and 400 images at 50 shots has a weld defect classification accuracy of 77% and 83%, respectively, while at 2000 shots, it has an accuracy of 95% and 97%, respectively. We could notice that the accuracy increases as the quantity of the training datasets and the number of shots increases.

4.3. Results by adding noise to dataset:

The noisy images have been superimposed on the weld TFM images to generate the noisy dataset. These images have been fed to VQC and other classical algorithms (discussed above in the paper) for classification purposes. The loss values for 200 weld images trained on all the algorithms, including quantum and classical with 500 testing images, have been shown in Fig. 14. The loss values for all the algorithms have

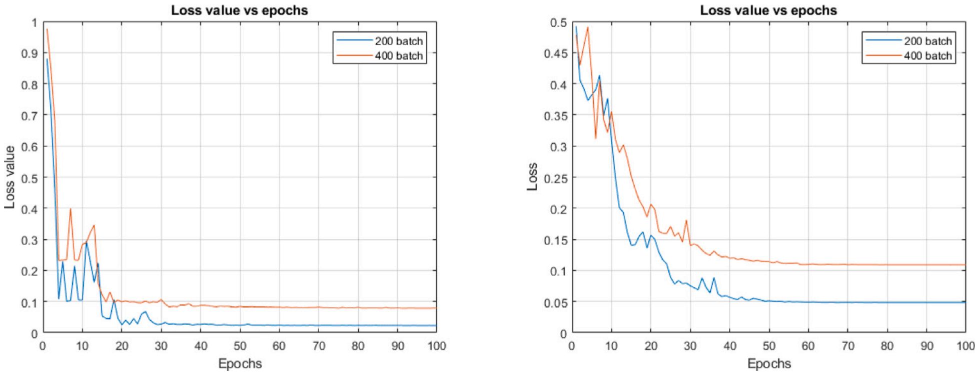

Fig. 13. Training loss for 200 and 400 batches of images of porosity and slag using VQC for 50 and 2000 shots.

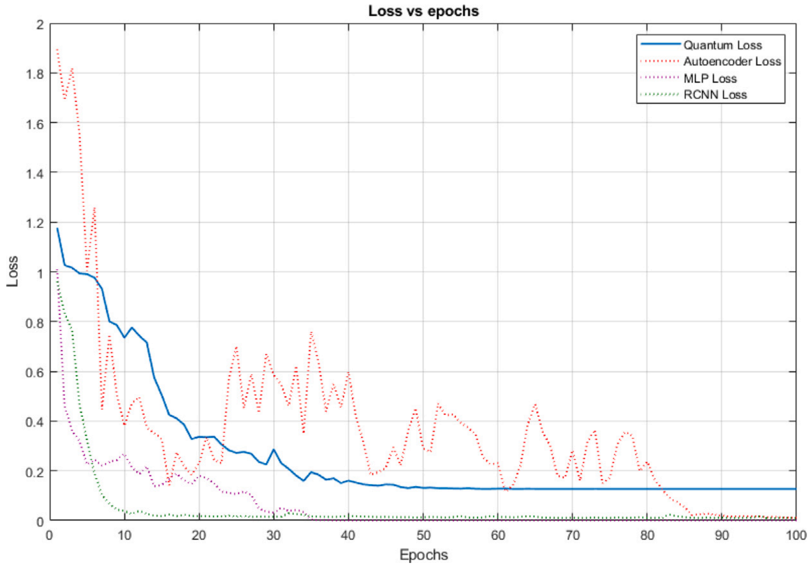

Fig. 14. Training loss for 200 batches of noisy images of porosity and slag using various classification algorithms.

Table 6
Classification metrics for 200 and 400 VQC trained porosity and slag images with three features tested on 500 images with 2000 shots.

		Porosity	slag	Precision	Recall	F1-score
200 batch	Porosity	248	2	0.91	0.99	0.95
	Slag	23	227	0.99	0.91	0.95
Overall						0.95
400 batch	Porosity	248	2	0.95	0.99	0.97
	Slag	13	237	0.99	0.95	0.97
Overall						0.97

been settled down almost to zero, indicating that models have learned accurately to differentiate the images.

The classification metrics for all the algorithms are shown in Table 7 The following conclusion can be drawn from the table.

- Even after superimposing the noise on weld TFM images, both quantum and classical algorithms accurately classify the images.
- VQC classification accuracy has decreased to some extent due to the addition of noise in the weld images, which is an obvious reason.
- Even though noise has been added to the images, quantum algorithm accuracy matches the classification accuracy of classical algorithms.

Table 7
Classification metrics for 200 and 400 VQC, MLP, Autoencoders and RCNN trained porosity and slag images tested on 500 images.

		Porosity	Slag	Precision	Recall	F1-score
VQC	Porosity	217	33	0.84	0.87	0.85
	Slag	41	209	0.86	0.84	0.85
Overall						0.85
MLP	Porosity	172	78	0.64	0.68	0.66
	Slag	97	153	0.66	0.61	0.63
Overall						0.64
Auto	Porosity	191	59	0.74	0.76	0.75
	Slag	67	183	0.76	0.73	0.74
Overall						0.74
RCNN	Porosity	219	31	0.84	0.88	0.86
	Slag	40	210	0.87	0.84	0.85
Overall						0.85

- From Table 7, classification accuracy for porosity and slag using VQC is 85%.
- The classification accuracy for MLP, autoencoders, and RCNN is 65%, 74%, and 85%, respectively.

The tables above show that the quantum algorithm performs comparably to classical techniques when adding noise to the images. The accuracy of classification in noisy images has decreased as we compare results with and without noise. The VQC method has the highest accuracy decline, yet compared to the classification of noisy images, VQC performs on par with traditional techniques. All the quantum results have The next section discusses the conclusion and the future work of this paper.

4.4. Speed advantage of using VQC:

The training process in VQC involves optimizing the parameters of quantum circuits to minimize a loss function. Due to the quantum parallelism, VQC converges faster than classical models. For instance, the quantum gradient descent algorithm used in VQC explores the parameter space more efficiently, reducing the number of iterations needed to reach optimal parameters. The hybrid quantum–classical approach used in VQC allows the model to take advantage of quantum speedup for certain parts of the computation while relying on classical computing for others. During inference, VQC classifies input data points more efficiently by leveraging quantum parallelism to evaluate multiple potential outcomes simultaneously. This is particularly beneficial for high-dimensional datasets, where classical algorithms might struggle with the curse of dimensionality. The ability of quantum circuits to encode and process high-dimensional data efficiently leads to faster inference times.

The training time for the VQC model is 1048 s for 400 images for training 100 epochs and 720 s for 200 images of training whereas for classical models it varies between 2 to 3 h which clearly shows the speed advantage using VQC over the classical methods mentioned in the paper. The computational efficiency and speed of training and inference in VQC highlight the potential advantages of quantum computing in machine learning. As quantum hardware continues to develop, we anticipate that these advantages will become increasingly apparent, making VQC a valuable tool for tackling complex classification tasks in various domains. The findings from our initial experiments underscore the importance of further exploration into the scalability and applicability of VQC in real-world scenarios.

5. Future work

In this paper, we introduced VQC, a quantum machine learning algorithm trained on the quantum computing simulator platform to classify the simulation-assisted weld defects. The VQC model is trained on the combined datasets of FE and artificial intelligence-generated datasets. The training time for the VQC model is 1048 s for 400 images for training 100 epochs. The proposed model predicted maximum overall accuracy on the non-noisy and noisy weld defect images as 97% and 85%, respectively. Classical algorithms like MLP, autoencoders, and RCNN are also trained and tested on the same dataset to check the performance of the VQC model. The classical models' maximum prediction accuracy on the noisy and non-noisy datasets is 91% and 85%, both shown by the RCNN model, which is less than the VQC prediction accuracy. The numerical experiments showed that VQCs outperform the classical results in accuracy and time. Hence, the VQC model is a potential tool for real-time monitoring of weld defects. In future work, the authors want to experimentally introduce more than one weld defect of different kinds into the simulated weld images to train the VQC model and implement it on the field to provide weld monitoring.

VQC has emerged as a promising approach to leverage the power of quantum computing to solve complex computational problems. They provide a useful, responsive, and extensible quantum machine learning model to classify real-world weld defect detection problems. The results show that high accuracy could be achieved even by supplying a smaller number of features, although there could be different ways of constructing ansatzes for VQC leading to further improvement within the same algorithm. Authors are considering the use of quantum convolutional networks directly on the images instead of feature selection as future work with the hope of pushing the boundary further. The current experiments are primarily conducted on simulated data, which provides a controlled environment for initial testing. However, future work will involve extensive testing of VQC on actual weld inspection data. This will include datasets obtained from industrial partners and real-world inspection scenarios, where we can assess the practical applicability and robustness of our approach. While our current study focuses on porosity and slag defects, we plan to expand our experiments to include a broader range of defect types, such as cracks, lack of fusion, and undercuts. These additional experiments will help us better understand the generalizability of VQC and its effectiveness in detecting and classifying a wider variety of defects in future. To realize their full potential and enable quantum computing to address some of the most serious issues society faces today, continued research and development in VQCs will be essential.

CRediT authorship contribution statement

Anurag Dubey: Writing – review & editing, Writing – original draft, Visualization, Validation, Methodology, Investigation, Funding acquisition, Formal analysis, Data curation, Conceptualization. **Thulsiram Gantala:** Writing – review & editing, Writing – original draft, Visualization, Validation, Supervision, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Anupama Ray:** Supervision, Software, Methodology. **Anil Prabhakar:** Writing – review & editing, Visualization, Validation, Supervision, Software, Resources, Project administration. **Prabhu Rajagopal:** Writing – review & editing, Visualization, Validation, Supervision, Resources, Project administration, Methodology, Investigation, Funding acquisition, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This research was supported in part by a grant from the Mphasis F1 Foundation to the Centre for Quantum Information, Communication, and Computing (CQuICC), India. The authors also gratefully acknowledge the support from Dr Prabhu Rajagopal under the National Swarnajaynati Fellowship.

Appendix A

A.1. Classification using multi-layer perceptron

MLP is an artificial neural network consisting of multiple layers of interconnected nodes known as neurons. Each neuron performs a weighted sum of its inputs, passes the result through an activation function, and generates an output. MLPs are known for their ability to approximate complex nonlinear functions, making them suitable for various machine-learning tasks, including image classification.

In the context of image classification, MLP takes the raw pixel values of an image as input and processes them through one or more hidden layers before generating a final output, as shown in Fig. 15. The architecture of a neural network typically consists of three main types of layers: input layer, hidden layer(s), and output layer. Each of these layers plays a specific role in the network's ability to learn and make predictions [33].

The input layer is the network's first layer and represents the image's raw pixel values. Each neuron in the input layer corresponds to a pixel value, and together, they form a representation of the image that the network can process. The input layer is where the network receives

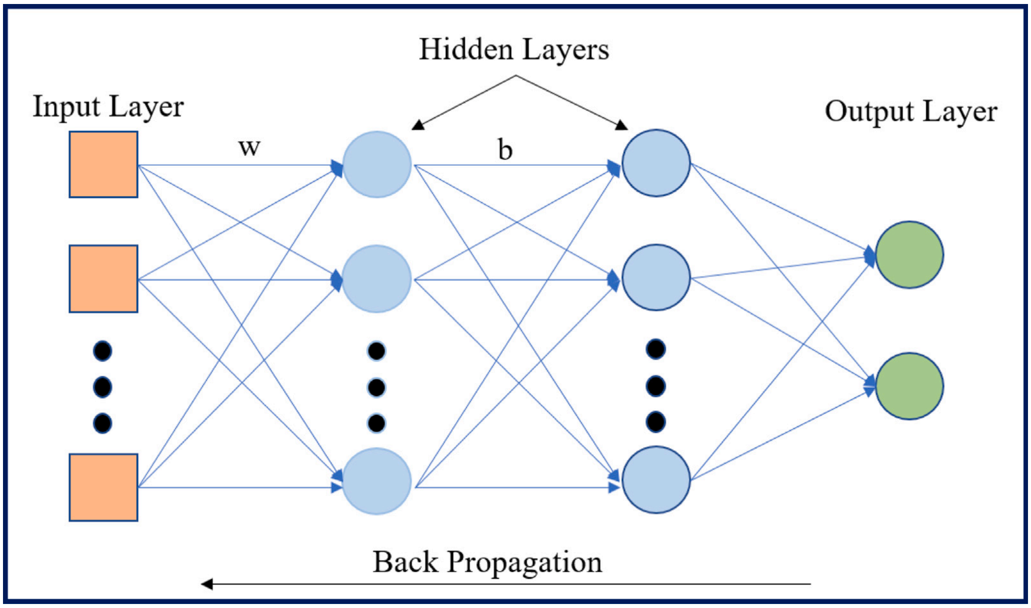

Fig. 15. Schematic diagram illustrating the architecture for MLP network.

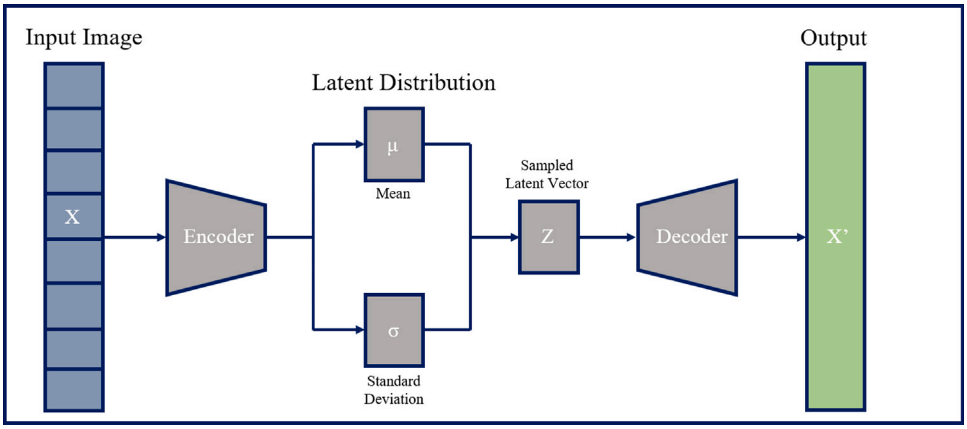

Fig. 16. Schematic diagram illustrating the architecture for an ordinary Autoencoder.

its input from the outside world and prepares it for processing by subsequent layers. The network's hidden layer(s) perform intermediate computations by applying weighted sums and activation functions to the inputs from the previous layer. Each neuron in the hidden layer(s) captures different features or representations of the input image. These features can be thought of as abstract representations of the image, such as edges, corners, and textures. By combining these features in various ways, the network is able to recognize complex patterns and make accurate predictions about the input. The output layer of the network produces the final classification prediction. Each neuron in the output layer represents a specific class label, and the neuron with the highest activation indicates the predicted class.

The training process of an MLP for image classification involves several steps [34]. Firstly, the raw pixel values of the input images are preprocessed and normalized to ensure consistent input across the network. The dataset is divided into training and validation sets. The network's weights and biases are initialized randomly, and the forward propagation process begins. During forward propagation, the inputs flow through the network, and the outputs are generated. The generated outputs are compared with the actual labels, and the error is calculated. Backpropagation is then performed, where the error is propagated backward through the network, and the weights are adjusted using optimization algorithms like gradient descent or its variants. This process is repeated iteratively until convergence or a predefined stopping criterion is met.

A.2. Classification using autoencoders

An autoencoder is a neural network architecture consisting of an encoder and a decoder [35]. The encoder encodes the input data into a lower-dimensional latent space representation, while the decoder reconstructs the input data from the latent space representation. In the case of image classification, the autoencoder takes an image as input, compresses it into a latent representation, and then reconstructs the image from the latent space.

The training process of an autoencoder involves two main steps: encoding and decoding. The input image is passed through the encoder network during encoding, which maps it to a lower-dimensional latent space representation. This latent representation captures the essential features of the input image. Subsequently, the latent representation is passed through the decoder network during decoding, which aims to reconstruct the original input image. The reconstruction error is computed as the difference between the original input image and the reconstructed image [36]. The autoencoder is trained to minimize this reconstruction error through backpropagation and optimization algorithms like stochastic gradient descent. The network architecture of an autoencoder is shown in Fig. 16.

A.3. Classification with region-based convolutional neural network model

Object detection is a fundamental task in computer vision involving the localization and classification of objects within an image. Traditional object detection methods relied on handcrafted features and sliding window techniques, which were computationally expensive and lacked accuracy. RCNN addressed these limitations by introducing a unified framework that leverages the power of CNNs for feature extraction and classification [37].

The RCNN architecture consists of three main components: region proposal, feature extraction, and object classification/localization [38]. Initially, a region proposal algorithm (such as selective search) generates a set of potential object regions within an input image. These regions are subsequently warped and fed into a CNN to extract features. The CNN, usually pre-trained on large-scale image classification tasks, acts as a powerful feature extractor. Finally, the extracted features are inputted into separate, fully connected layers for classification and bounding box regression.

The training process of RCNN involves several steps. First, a set of labeled images is used to generate region proposals and associate them with ground truth object annotations. These proposals and their associated ground truth labels are used to fine-tune the pre-trained CNN. This process allows the network to learn discriminative features for object recognition [39]. During training, two separate loss functions are used: a softmax loss for classification and a smooth L1 loss for bounding box regression. The network is trained end-to-end using backpropagation and stochastic gradient descent.

Data availability

Data will be made available on request.

References

[1] Drinkwater BW, Wilcox PD. Ultrasonic arrays for non-destructive evaluation: A review. NDT e Int 2006;39(7):525–41.

[2] Satyanarayan L, Sridhar C, Krishnamurthy C, Balasubramaniam K. Simulation of ultrasonic phased array technique for imaging and sizing of defects using longitudinal waves. Int J Press Vessels Pip 2007;84(12):716–29.

[3] Jobst M, Connolly GD. Demonstration of the application of the total focusing method to the inspection of steel welds. In: 10th European conference on non-destructive testing. 2010, p. 1–11.

[4] Cruza JF, Camacho J, Mateos R, Fritsch C. A new beamforming method and hardware architecture for real time two way dynamic depth focusing. Ultrasonics 2019;99:105965.

[5] Virkkunen I, Koskinen T, Jessen-Juhler O, Rinta-aho J. Augmented ultrasonic data for machine learning. J Nondestruct Eval 2021. http://dx.doi.org/10.1007/s10921-020-00739-5.

[6] Jiangsha A, Tian L, Bai L, Zhang J. Data augmentation by a cyclegan-based extra-supervised model for nondestructive testing. Meas Sci Technol 2022;33(4):045017.

[7] Gantala T, Sudharsan P, Balasubramaniam K. Automated defect recognition (ADR) for monitoring industrial components using neural networks with phased array ultrasonic images.. Meas Sci Technol 2023.

[8] Munir N, Kim H-J, Park J, Song S-J, Kang S-S. Convolutional neural network for ultrasonic weldment flaw classification in noisy conditions. Ultrasonics 2019;94:74–81. http://dx.doi.org/10.1016/j.ultras.2018.12.001.

[9] Bai L, Le Bourdais F, Miorelli R, Calmon P, Velichko A, Drinkwater BW. Ultrasonic defect characterization using the scattering matrix: A performance comparison study of Bayesian inversion and machine learning schemas. IEEE Trans Ultrason Ferroelectr Freq Control 2021;68(10):3143–55. http://dx.doi.org/10.1109/TUFFC.2021.3084798.

[10] Bernieri A, Ferrigno L, Laracca M, Molinara M. Crack shape reconstruction in eddy current testing using machine learning systems for regression. IEEE Trans Instrum Meas 2008;57(9):1958–68. http://dx.doi.org/10.1109/TIM.2008.919011.

[11] Sambath S, Nagaraj P, Selvakumar N. Automated defect recognition for welds using simulation assisted TFM imaging with artificial intelligence. Automatic Defect Classif Ultrasonic NDT Artif Intell 2021;30:20–8.

[12] Gantala T, Balasubramaniam K. Automated defect recognition for welds using simulation assisted tfm imaging with artificial intelligence. J Nondestruct Eval 2021;40(1):28. http://dx.doi.org/10.1007/s10921-021-00761-1.

[13] Siljama O, Koskinen T, Jessen-Juhler, Virkkunen I. Automat Flaw Detect Multichannel Phased Array Ultrasonic Data Mach Learn 2021;67. http://dx.doi.org/10.1007/s10921-021-00796-4.

[14] Cantero-Chinchilla S, Wilcox PD, Croxford AJ. A deep learning based methodology for artefact identification and suppression with application to ultrasonic images. NDT & E Int 2022;126:102575.

[15] Popescu M-C, Balas VE, Perescu-Popescu L, Mastorakis N. Multilayer perceptron and neural networks. WSEAS Trans Circ Syst 2009;8(7):579–88.

[16] Bank D, Koenigstein N, Giryes R. Autoencoders. In: Machine learning for data science handbook: data mining and knowledge discovery handbook. Springer; 2023, p. 353–74.

[17] Pabitha C, Vanathi B. Densemask RCNN: A hybrid model for skin burn image classification and severity grading. Neural Process Lett 2021;53:319–37.

[18] Forcer TM, Hey AJ, Ross D, Smith P. Superposition, entanglement and quantum computation. Quantum Inf Process 2002;2(2):97–116.

[19] Zaidenberg DA, Sebastianelli A, Spiller D, Le Saux B, Ullo SL. Advantages and bottlenecks of quantum machine learning for remote sensing. In: 2021 IEEE international geoscience and remote sensing symposium IGARSS. IEEE; 2021, p. 5680–3.

[20] Alam M, Kundu S, Topaloglu RO, Ghosh S. Quantum-classical hybrid machine learning for image classification (ICCAD special session paper). In: 2021 IEEE/ACM international conference on computer aided design. IEEE; 2021, p. 1–7.

[21] Holmes C, Drinkwater BW, Wilcox PD. Post-processing of the full matrix of ultrasonic transmit–receive array data for non-destructive evaluation. NDT e Int 2005;38(8):701–11.

[22] Gantala T, Balasubramaniam K. Automated defect recognition for welds using simulation assisted TFM imaging with artificial intelligence. J Nondestruct Eval 2021;40:1–24.

[23] Kherif F, Latypova A. Principal component analysis. In: Machine learning. Elsevier; 2020, p. 209–25.

[24] Brownlee J. How to choose a feature selection method for machine learning. Machine Learn Mastery 2019;10.

[25] Chan J, Zhou C, Livny M, Yoo S, Carminati F, Guan W, Wang A, Spentzouris P, Sun S, Li AC, et al. SISSA: Application of quantum machine learning to high energy physics analysis at LHC using IBM quantum computer simulators and IBM quantum computer hardware. PoS 2021;930.

[26] Havlíček V, Córcoles AD, Temme K, Harrow AW, Kandala A, Chow JM, et al. Supervised learning with quantum-enhanced feature spaces. Nature 2019;567(7747):209–12.

[27] Schuld M, Killoran N. Quantum machine learning in feature Hilbert spaces. Phys Rev Lett 2019;122(4):040504.

[28] Yetiş H, Karaköse M. Variational quantum circuits for convolution and window-based image processing applications. Quantum Sci Technol 2023;8(4):045004.

[29] Pramanik S, Chandra MG, Sridhar C, Kulkarni A, Sahoo P, Vishwa CD, Sharma H, Navelkar V, Poojary S, Shah P, et al. A quantum-classical hybrid method for image classification and segmentation. In: 2022 IEEE/ACM 7th symposium on edge computing. IEEE; 2022, p. 450–5.

[30] Pellow-Jarman A, Sinayskiy I, Pillay A, Petruccione F. A comparison of various classical optimizers for a variational quantum linear solver. Quantum Inf Process 2021;20(6):202.

[31] Fang W, Zhang F, Sheng VS, Ding Y. A method for improving CNN-based image recognition using dcgan.. Comput Mater Continua 2018;57(1).

[32] Goutte C, Gaussier E. A probabilistic interpretation of precision, recall and F-score, with implication for evaluation. In: European conference on information retrieval. Springer; 2005, p. 345–59.

[33] Dino HI, Abdulrazzaq MB. Facial expression classification based on SVM, KNN and MLP classifiers. In: 2019 international conference on advanced science and engineering. IEEE; 2019, p. 70–5.

[34] Zanchettin C, Ludermir TB, Almeida LM. Hybrid training method for MLP: optimization of architecture and training. IEEE Trans Syst Man Cybern B 2011;41(4):1097–109.

[35] Sun Y, Xue B, Zhang M, Yen GG. A particle swarm optimization-based flexible convolutional autoencoder for image classification. IEEE Trans Neural Netw Learn Syst 2018;30(8):2295–309.

[36] Bunrit S, Kerdprasop N, Kerdprasop K. Improving the representation of cnn based features by autoencoder for a task of construction material image classification. J Adv Inf Technol 2020;11(4).

[37] Cheng B, Wei Y, Shi H, Feris R, Xiong J, Huang T. Revisiting rcnn: On awakening the classification power of faster rcnn. In: Proceedings of the European conference on computer vision. 2018, p. 453–68.

[38] Ren Y, Zhu C, Xiao S. Object detection based on fast/faster RCNN employing fully convolutional architectures. Math Probl Eng 2018;2018:1–7.

[39] Chen X, Gupta A. An implementation of faster rcnn with study for region sampling. 2017, arXiv preprint arXiv:1702.02138.