engineering proceedings

MDPI

Proceeding Paper

Hybrid Quantum–Classical AI for Industrial Defect Classification in Welding Images †

Akshaya Srinivasan [1,2,*], Xiaoyin Cheng [1], Jianming Yi [3], Alexander Geng [1,*], Desislava Ivanova [4], Andreas Weinmann [5] and Ali Moghiseh [1,*]

[1] Fraunhofer Institute for Industrial Mathematics ITWM, 67663 Kaiserslautern, Germany; xiaoyin.cheng@itwm.fraunhofer.de
[2] Department of Computer Science, RPTU Kaiserslautern-Landau, 67663 Kaiserslautern, Germany
[3] MuVision UG, Gasstraße 21, 67655 Kaiserslautern, Germany; jianming.yi@muvision.com
[4] Faculty of Applied Mathematics and Informatics, Technical University of Sofia, 8 Kliment Ohridski Blvd, 1000 Sofia, Bulgaria; d_ivanova@tu-sofia.bg
[5] Algorithms for Computer Vision, Imaging and Data Analysis Lab, Technical University of Applied Sciences, Würzburg-Schweinfurt, Ignaz-Schön-Str. 11, 97421 Schweinfurt, Germany; andreas.weinmann@thws.de
* Correspondence: akshaya.srinivasan@itwm.fraunhofer.de (A.S.); alexander.geng@itwm.fraunhofer.de (A.G.); ali.moghiseh@itwm.fraunhofer.de (A.M.)
† Presented at the 15th International Scientific Conference TechSys 2026—Engineering, Technologies and Systems, Plovdiv, Bulgaria, 14–16 May 2026.

Abstract

Hybrid quantum–classical machine learning offers a promising direction for advancing automated quality control in industrial settings. In this study, we investigate two hybrid quantum–classical approaches for classifying defects in aluminum TIG welding images and benchmarking their performance against a conventional deep learning model. A convolutional neural network is used to extract compact and informative feature vectors from weld images, effectively reducing the higher-dimensional pixel space to a lower-dimensional feature space. Our first quantum approach encodes these features into quantum states using a parameterized quantum feature map composed of rotation and entangling gates. We compute a quantum kernel matrix from the inner products of these states, defining a linear system in a higher-dimensional Hilbert space corresponding to the support vector machine (SVM) optimization problem and solving it using a Variational Quantum Linear Solver (VQLS). We also examine the effect of the quantum kernel condition number on classification performance. In our second method, we apply angle encoding to the extracted features in a variational quantum circuit and use a classical optimizer for model training. Both quantum models are tested on binary and multiclass classification tasks, and the performance is compared with the classical CNN model. Our results show that while the CNN model demonstrates robust performance, hybrid quantum–classical models perform competitively. This highlights the potential of hybrid quantum–classical approaches for near-term real-world applications in industrial defect detection and quality assurance.

Keywords: hybrid quantum–classical algorithm; quantum machine learning; welding defect classification; variational quantum linear solver; variational quantum circuit; support vector machine; NISQ devices

check for updates

Academic Editors: Nikola G. Shakev, Valyo N. Nikolov, Nikolay R. Kakanakov and Sevil A. Ahmed-Shieva

Published: 1 August 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

1. Introduction

Automated quality control is a cornerstone of modern industrial manufacturing, enabling early defect detection, process optimization, and assurance of product quality. In

sectors such as the automotive industry, aerospace, and heavy machinery, welding is a critical joining process in which errors can compromise structural safety. Convolutional neural networks (CNNs) have achieved notable success in image-based classification tasks, including weld quality assessment [1,2]. However, these models demand significant computational resources, particularly as the data dimensionality and model complexity increase.

In parallel with the evolution of classical machine learning, quantum computing has emerged as a novel computational paradigm, exploiting principles of superposition and entanglement to process information in fundamentally different ways. While fault-tolerant quantum computing remains a long-term goal, the current noisy intermediate-scale quantum (NISQ) era [3] has enabled the development of hybrid quantum–classical models that combine quantum circuits with classical optimization and pre-/post-processing. These hybrid approaches have shown promise in domains such as machine learning, finance, and computer vision tasks (see, e.g., [4–11]). Quantum machine learning (QML), especially in its hybrid quantum–classical form, offers new possibilities for extracting information from data with potentially fewer parameters or enhanced generalization capabilities. In this work, we investigate two hybrid quantum-classical approaches for the classification of weld defects in aluminum TIG (tungsten inert gas) welding images. Figure 1 gives an overview of our work.

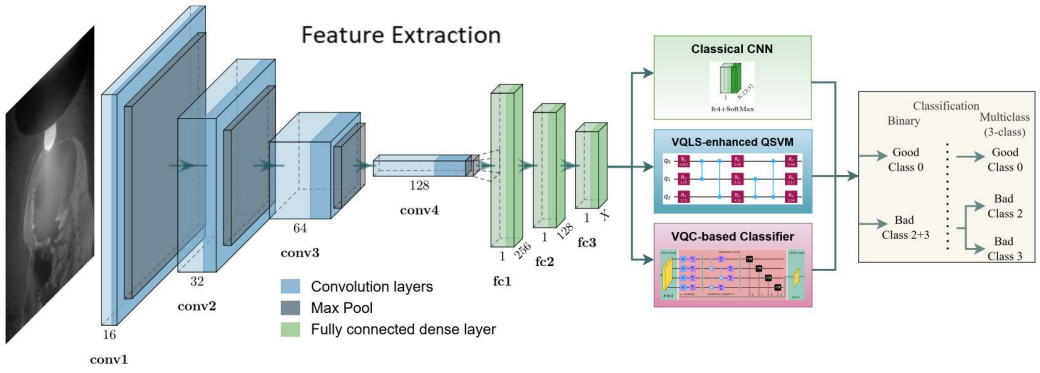


Figure 1. Full architecture of the study, showing CNN-based feature extraction followed by three classification approaches: classical CNN, VQLS-enhanced QSVM, and VQC-based classifier.

We begin by extracting features from welding images using a CNN architecture adapted from Model-1 described in the study [12]. Unlike generic pretrained models such as ResNet18 [13,14], we selected Model-1 because it was specifically designed and tested for weld image classification. Out of the 13 architectures evaluated in that study, Model-1 showed superior performance, making it well-suited for our task. We trained this model end-to-end on our dataset to generate feature vectors for subsequent classification. This part serves as a feature extractor, transforming a higher-dimensional pixel space into a lower-dimensional feature space. These extracted features form the input for two distinct quantum models, too.

Our first approach employs a quantum kernel method, introduced by Yi et al. in [15]. Feature vectors are embedded into quantum states using a parameterized feature map composed of rotation and entangling gates. A quantum kernel matrix is constructed via inner products of these states, effectively mapping the data into a higher-dimensional Hilbert space. This kernel is integrated into a support vector machine (SVM) formulation, where the optimization problem is solved using the Variational Quantum Linear Solver (VQLS), a hybrid algorithm designed to approximate the solution to linear systems on quantum hardware [16,17]. Additionally, we investigate the condition number of the quantum kernel matrix and its influence on classification performance and numerical

stability. The model is tested on a range of classification tasks, including binary and multiclass scenarios, to measure how well it performs.

In the second approach, we use a variational quantum circuit (VQC) where classical features are angle-encoded into quantum gates. A parameterized quantum circuit is trained using a classical optimizer to minimize a loss function defined by the classification task. This model is evaluated on both binary and multiclass classification problems to assess its expressivity and performance under varying task complexities.

By comparing these hybrid methods with a classical CNN baseline, this study aims to assess the feasibility of quantum models in real-world industrial applications. Through this comparative analysis, we seek to better understand the potential and limitations of current quantum machine learning approaches in the context of weld defect image classification.

2. Algorithms

2.1. Feature Extraction

Due to the current limitations of NISQ-era quantum hardware, particularly in terms of qubit count, gate fidelity, and circuit depth, processing high-resolution image data directly on quantum processors is not yet feasible. Encoding an entire image of even moderate resolution requires a large number of qubits. Downsizing the input too aggressively risks discarding vital spatial and structural information.

To overcome this challenge, we adopt a hybrid classical–quantum approach in which the classical component performs dimensionality reduction to make the data suitable for quantum processing. A convolutional neural network (CNN) is used to extract meaningful features from the raw images, compressing them into a lower-dimensional and information-rich vector. This preserves essential image characteristics while keeping the input dimensions within the feasible range for current quantum devices.

As illustrated in Figure 2, feature vectors are extracted from a fully connected (dense) layer of the CNN. These vectors serve as inputs for the subsequent quantum processing steps. We systematically evaluated multiple feature sizes, specifically, vectors of lengths 7, 15, 31, 63, and 127 to determine the optimal dimensionality for each quantum pipeline. The goal was to balance the quantum circuit depth and fidelity against the classification performance of the model. This intermediate feature representation serves a dual purpose: it reduces the input dimensionality for quantum encoding and captures the most relevant information from the input data. By integrating classical convolutional layers with quantum models, we design an interface that transforms complex higher-dimensional image data into simpler lower-dimensional information that quantum circuits can process effectively.

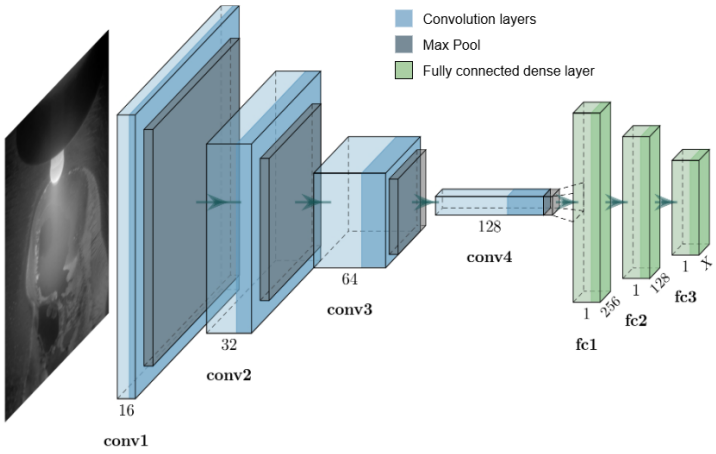


Figure 2. CNN architecture for feature extraction. The last fully connected layer outputs a feature vector $X \in \{7, 15, 31, 63, 127\}$.

2.2. Classical Convolutional Neural Network

As a baseline for performance benchmarking, we implement a conventional convolutional neural network (CNN) model inspired by the architecture described as Model-1 in [12]. This model is used both for feature extraction in the hybrid quantum–classical pipeline and as a classifier for end-to-end image-based defect detection, depending on whether the final fully connected (FC) layer is included. If the final FC layer is present, the model performs classification; else, the output from the preceding layer is used as a feature vector input to the hybrid quantum–classical pipeline. The CNN comprises a series of convolutional layers with max-pooling and activation functions, designed to progressively abstract spatial features from the input images. These convolutional layers learn hierarchical representations of weld characteristics such as texture, edge continuity, and defect patterns. The extracted spatial features are then flattened and passed through one or more dense layers. For the classical classification pipeline, we append a final fully connected layer followed by a softmax activation function as shown in Figure 3. The number of outputs in this layer determines the classification tasks: two for binary classification (good vs. defective welds) and three or more for multiclass classification (good weld, contamination, and lack of fusion). The softmax function transforms the raw outputs into normalized class probabilities, allowing the model to assign confidence scores to each class. The network is trained using categorical cross-entropy loss and optimized via backpropagation using the Adam optimizer for different feature sizes.

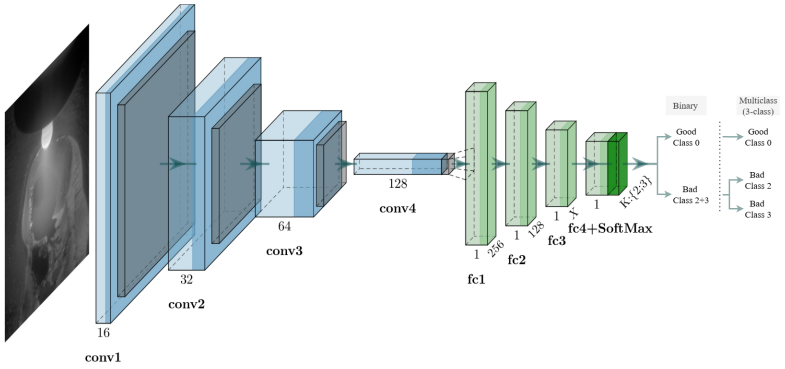


Figure 3. Classical CNN architecture for classification.

This classical CNN serves as a strong benchmark due to its proven performance in previous studies and industrial applications. Moreover, it forms the backbone of our hybrid models by supplying robust feature vectors to the quantum circuits. Thus, its role is central not only in the classical workflow but also as a bridge between raw image data and quantum processing in the hybrid architectures.

2.3. VQLS-Enhanced QSVM

The VQLS-enhanced QSVM approach extends the classical support vector machine (SVM) into the quantum domain by leveraging a quantum kernel and solving the associated optimization problem using a Variational Quantum Linear Solver (VQLS). The pipeline of this algorithm is illustrated in Figure 4. This hybrid model aims to classify images based on quantum-encoded features while maintaining compatibility with Noisy Intermediate-Scale Quantum (NISQ) hardware. The pseudocode for this method is given in Algorithm 1. The main steps are explained in detail as follows:

Algorithm 1 VQLS-enhanced QSVM

Input: Feature samples $\mathbf{X}_{\text{train}} = \{\mathbf{x}_1, \cdots, \mathbf{x}_N\}$, labels $\mathbf{y}_{\text{train}} = \{y_1, \cdots, y_N\}$
Output: Optimal parameters $\boldsymbol{\alpha}_{\text{opt}}$
Normalize $\mathbf{X}_{\text{train}} \rightarrow \hat{\mathbf{X}}_{\text{train}}$
Construct kernel matrix K using quantum kernel estimation
Pauli decomposition of K into $\sum_{\ell=0}^{N} c_\ell A_\ell$ with coefficients c_ℓ and Pauli strings A_ℓ
Initialize $i \leftarrow 0$, cost $C \leftarrow 1$, threshold $\epsilon \leftarrow 0.01$, max iterations $T_{\max} \leftarrow 300$
Randomly initialize parameters $\boldsymbol{\alpha}^{(0)}$
while $C > \epsilon$ **and** $i < T_{\max}$ **do**
  $sum_1 \leftarrow 0$
  for A_m in $\{A_1, \cdots, A_N\}$ **do**
    for A_n in $\{A_1, \cdots, A_N\}$ **do**
      Construct Circuit-1 to compute $\langle 0|V(\alpha)^\dagger A_m^\dagger A_n V(\alpha)|0\rangle$ using Hadamard test
      Execute with 10,000 shots, measure ancilla qubit with overlap results $\rightarrow$ get expectation value
      $sum_1 \leftarrow sum_1 + c_m^* c_n \cdot$ measured value
    end for
  end for
  $sum_2 \leftarrow 0$
  for A_m in $\{A_1, \cdots, A_N\}$ **do**
    for A_n in $\{A_1, \cdots, A_N\}$ **do**
      Construct Circuit-2 for overlap of $|b\rangle$ with ansatz after A_n; Circuit-3 for overlap of ansatz after $A_m^\dagger$ with $|b\rangle$
      Circuit-2 to compute $\langle 0|U^\dagger A_n V(\alpha)|0\rangle$
      Execute, measure, store result
      Circuit-3 for $\langle 0|V(\alpha)^\dagger A_m^\dagger U|0\rangle$
      Execute, measure, store result
      $sum_2 \leftarrow sum_2 + c_m^* c_n$ product of both results
    end for
  end for
  Compute $C \leftarrow 1 - - - \frac{sum_1}{sum_2}$
  $i \leftarrow i + 1$
  Update $\boldsymbol{\alpha}^{(i)}$ using COBYLA optimizer
end while
return $\boldsymbol{\alpha}_{\text{opt}}$
Measure final ansatz state $\boldsymbol{\alpha}_{\text{opt}}$ to extract amplitudes θ'
Apply linear regression: $\theta = A \cdot \theta' + b$
Define decision function: $f(\mathbf{x}) = \sum_i \alpha_i K(\mathbf{x}_i, \mathbf{x}) + \text{bias}$
return $\boldsymbol{\alpha}_{\text{opt}}$, $f(\mathbf{x})$

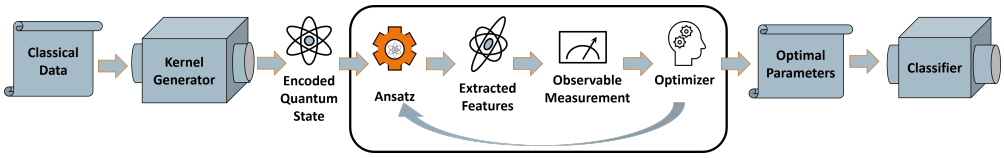


Figure 4. Pipeline for VQLS-enhanced QSVM algorithm.

2.3.1. Feature Normalization

To ensure stable performance during quantum processing, the CNN-extracted feature vectors are first normalized. Normalization is necessary because these vectors are used to prepare quantum states, which require unit norm (i.e., $\|\mathbf{x}\| = 1$). Let $\mathbf{x}_i \in \mathbb{R}^d$ be the feature vector for sample i, and the normalized vector is given by: $\tilde{\mathbf{x}}_i = \frac{\mathbf{x}_i}{\|\mathbf{x}_i\|}$. These normalized vectors are then encoded into quantum states $|\tilde{\mathbf{x}}_i\rangle$.

2.3.2. Quantum Kernel Construction

The quantum kernel matrix $K \in \mathbb{R}^{n \times n}$ is constructed using the inner products between quantum-encoded data points:

$$K_{ij} = \left|\langle \tilde{\mathbf{x}}_i | \tilde{\mathbf{x}}_j \rangle\right|^2 \tag{1}$$

This kernel effectively maps classical features into a higher-dimensional Hilbert space where linear decision boundaries become more discriminative.

2.3.3. Linear System Formulation

For a soft-margin linear SVM, the optimal separating hyperplane can be obtained by solving a linear system [18]:

$$M\boldsymbol{\alpha} = \mathbf{b} \tag{2}$$

where $M = K + \lambda I$, $\boldsymbol{\alpha}$ are the dual coefficients, λ is a regularization parameter, and $\mathbf{b}$ contains the labels in the dual formulation.

This equation is translated into the quantum domain as

$$\hat{M}|\alpha\rangle = |b\rangle, \tag{3}$$

where $\hat{M}$ is a Hermitian matrix obtained via Pauli decomposition of the classical matrix M, $\hat{M} = \sum_j c_j P_j$, and $|\alpha\rangle$ is the quantum state approximating the solution.

2.3.4. Hardware-Efficient Ansatz Construction

To approximate the solution $|\alpha\rangle$, we construct a parameterized quantum circuit (ansatz) using rotational $R_y(\theta)$ and entangling controlled-$R_z(\phi)$ gates, as shown in Figure 5. The parameters are initialized randomly and optimized during training.

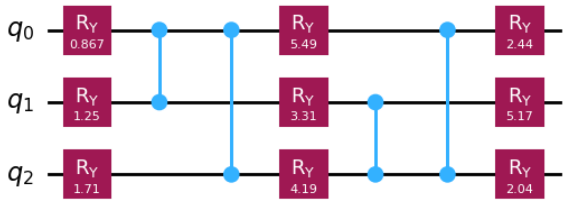


Figure 5. Quantum circuit for hardware-efficient ansatz.

2.3.5. Cost Function via Hadamard Test

A global cost function quantifies how close the current ansatz state $|\alpha(\boldsymbol{\theta})\rangle$ is to the true solution $|b\rangle$ of the linear system $\hat{M}|\alpha(\boldsymbol{\theta})\rangle = |b\rangle$, where $\hat{M}$ is the Hermitian matrix representing the linear system operator. The cost function $C(\boldsymbol{\theta})$ is given by

$$C(\boldsymbol{\theta}) = 1 - \frac{|\langle b|\hat{M}|\alpha(\boldsymbol{\theta})\rangle|^2}{\langle \alpha(\boldsymbol{\theta})|\hat{M}^2|\alpha(\boldsymbol{\theta})\rangle} \tag{4}$$

The numerator $(|\langle b|\hat{M}|\alpha(\boldsymbol{\theta})\rangle|^2)$ captures the overlap between the trial solution $\hat{M}|\alpha(\boldsymbol{\theta})\rangle$ and the true target vector $|b\rangle$, evaluated using Hadamard tests, while the denominator $(\langle \alpha(\boldsymbol{\theta})|\hat{M}^2|\alpha(\boldsymbol{\theta})\rangle)$ normalizes the overlap. Optimization of the variational parameter $\boldsymbol{\theta}$ to minimize $C(\boldsymbol{\theta})$ is performed using the classical COBYLA optimizer [19].

2.3.6. Solution State Preparation

Upon convergence, the final ansatz circuit represents the state, from which we measure the state vector to obtain the amplitude distribution:

$$|\alpha^*\rangle = \sum_i c_i |i\rangle \tag{5}$$

These coefficients c_i are used to construct the solution vector corresponding to the decision function. To extract meaningful classification parameters, we rescale the amplitudes using a linear regression model trained on the classical dual SVM form, yielding the final model parameters:

$$\boldsymbol{\theta} = A\boldsymbol{\theta}' + \mathbf{b}, \quad \text{bias} = b_0 \tag{6}$$

2.3.7. Decision Function

The final classifier uses the quantum kernel $K(\mathbf{x}_i, \mathbf{x})$ and the trained parameters to evaluate new inputs $\mathbf{x}$ using

$$f(\mathbf{x}) = \sum_i \alpha_i K(\mathbf{x}_i, \mathbf{x}) + b \tag{7}$$

where $\mathbf{x}_i$ represents the support vector data points, α_i represents the corresponding optimized coefficients and b is the bias.

For multiclass classification, we adopt a one-vs-rest (OvR) approach: a separate binary classifier is trained for each class against all others. During inference, the classifier with the highest decision value determines the predicted class. This strategy allows the VQLS-enhanced QSVM framework to be effectively extended beyond binary tasks.

2.4. Variational Quantum Circuit (VQC)-Based Classifier

The second quantum pipeline in our study is inspired by the hybrid transfer learning approach proposed by Geng et al. in [7]. Unlike the VQLS-enhanced QSVM, which solves a quantum linear system for classification, this model uses a variational quantum circuit trained via gradient descent. The classical component first reduces the input feature dimension, while the quantum circuit provides a trainable embedding to enhance classification performance. The pseudocode for this method is given in Algorithm 2, the pipeline is illustrated in Figure 6, and the key steps are outlined below:

Algorithm 2 VQC-based Classifier

Input: Feature vectors $\mathbf{X}_{\text{train}} = \{\mathbf{x}_1, \cdots, \mathbf{x}_N\}$ and labels $\mathbf{y}_{\text{train}}$
Output: Trained VQC parameters θ^* and classifier weights
for each input $\mathbf{x}_i$ **do**
  Reduce to 4D: $\mathbf{x}_i' = W\mathbf{x}_i + \mathbf{b}$
  Encode $\mathbf{x}_i'$ using R_y gates: $|\psi_{\text{init}}\rangle$
  Apply VQC: $|\psi(\theta)\rangle = U(\theta)|\psi_{\text{init}}\rangle$
  Measure $\langle Z_i \rangle$ on each qubit to obtain $\mathbf{z}$
  Compute prediction: $\hat{y}_i = \text{Softmax}(W_z\mathbf{z} + \mathbf{b}_z)$
end for
Compute loss: $\mathcal{L} = \text{CrossEntropy}(\hat{y}, y)$
Compute gradients (parameter-shift rule or backpropagation)
Update parameters θ, W_z, $\mathbf{b}_z$ using Adam
return Optimal θ^* and trained classifier

https://doi.org/10.3390/engproc2026150096

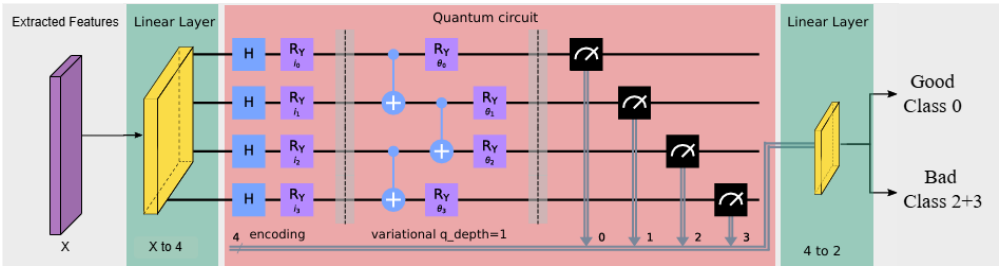


Figure 6. Pipeline for VQC-based classifier algorithm for binary classification.

2.4.1. Dimensionality Reduction

Given a classical feature vector $\mathbf{x} \in \mathbb{R}^d$ extracted from a CNN, we first reduce its dimension to match the number of available qubits. A linear transformation is applied:

$$\mathbf{x}' = W\mathbf{x} + \mathbf{b}, \quad \mathbf{x}' \in \mathbb{R}^4 \tag{8}$$

This projection retains critical semantic information while enabling feasible quantum encoding on near-term devices.

2.4.2. Quantum Embedding

Each element x_i' of the reduced feature vector is encoded into the state of a qubit using a parameterized single-qubit rotation gate:

$$|\psi_{\text{init}}\rangle = \bigotimes_{i=1}^{4} R_y(x_i')H|0\rangle \tag{9}$$

2.4.3. Variational Quantum Circuit (VQC)

The quantum processing is carried out by a hardware-efficient ansatz, constructed from trainable $R_y(\theta_i)$ gates and entangling CX gates. The unitary transformation $U(\theta)$ prepares the trial quantum state:

$$|\psi(\theta)\rangle = U(\theta)|\psi_{\text{init}}\rangle \tag{10}$$

We use a single variational layer, which strikes a good balance between maintaining accuracy and minimizing the number of circuit executions, helping to manage noise and errors in NISQ devices.

2.4.4. Measurement

Each qubit is measured in the Pauli-Z basis to yield a 4-dimensional output vector:

$$z_i = \langle\psi(\theta)|Z_i|\psi(\theta)\rangle, \quad i = 1, \ldots, 4 \tag{11}$$

$$\mathbf{z} = [z_1, z_2, z_3, z_4] \in \mathbb{R}^4 \tag{12}$$

2.4.5. Post-Quantum Processing

The measured quantum outputs are passed through a final classical linear layer followed by a softmax function for binary classification:

$$\hat{y} = \text{Softmax}(W_z\mathbf{z} + \mathbf{b}_z) \tag{13}$$

2.4.6. Training

The model is trained using cross-entropy loss and the Adam optimizer. Gradients are computed using the parameter-shift rule. All experiments are performed on the `PennyLane default.qubit` simulator, running locally on a CUDA-enabled GPU.

For multiclass classification, the VQC-based classifier model adjusts the number of outputs in the final classical linear layer to match the number of classes, with two for binary and three or more for multiclass tasks. The softmax activation then produces class probabilities accordingly, enabling seamless extension from binary to multiclass classification.

3. Dataset

The Kaggle dataset used in this study originates from an industrial setting, where high-resolution image sequences of Tungsten Inert Gas (TIG) welding processes were captured using a high dynamic range (HDR) camera system [20]. The primary objective of the data acquisition was to support the non-destructive testing (NDT) of aluminum welds by enabling real-time defect detection through visual inspection without damaging the material or structure.

The original dataset contains approximately 30,000 grayscale images, recorded at a sampling rate of 55 frames per second. Each image has a native resolution of 800×974 pixels, providing fine spatial details essential to recognize subtle weld imperfections. A team of welding experts manually annotated the dataset, categorizing the images into six classes corresponding to various types of welding defects as shown in Figure 7.

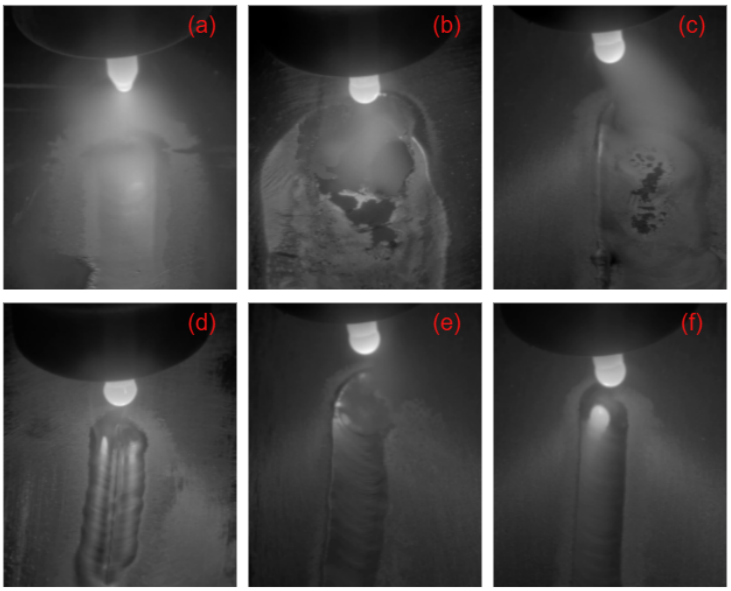


Figure 7. Six-class categorization of welding defects: (**a**) good weld, (**b**) burn through, (**c**) contamination, (**d**) lack of fusion, (**e**) misalignment, and (**f**) lack of penetration. These images are taken from Kaggle [20].

For this study, we curated a representative subset of this dataset, focusing on a three-class classification task. The selected classes include: Class 0—good weld, Class 2—contamination, and Class 3—lack of fusion. These three categories capture a meaningful distinction between good welds and two common, visually identifiable defect types, facilitating robust model training and evaluation, as seen in Figure 8.

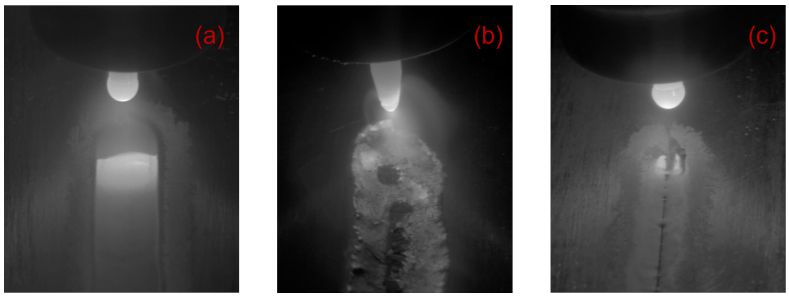


Figure 8. Three-class categorization of welding defects: (**a**) good weld, (**b**) contamination, and (**c**) lack of fusion.

The images were spatially downsampled to a resolution of 400 × 487 pixels to reduce the computational load while preserving defect-relevant features. We used a total of 1100 images in our study, allocating 700 for training and validation and 400 (100 of good welds, 150 each for defect classes 2 and 3) for testing. The dataset was balanced across the selected classes to ensure fair evaluation during both the training and testing phases. This carefully designed subset serves as a reliable benchmark for assessing the performance of both classical and hybrid quantum–classical machine learning models in industrial visual inspection scenarios.

4. Hyperparameter Tuning

To get the best performance from our hybrid quantum–classical classifier, we tuned several key hyperparameters using the Ray Tune Python package (v3.11) [21], which offers scalable and efficient optimization. Since quantum circuits are sensitive to noise, and training is expensive, careful tuning was necessary to ensure both stability and accuracy.

4.1. Tuned Hyperparameters

The following parameters were explored:

- Batch size—Tested values: 4, 8, 12, 16. Smaller batch sizes provided more frequent updates and some regularization, while larger ones helped with more stable gradient estimates.
- Feature Size—We tried feature vectors of sizes [15] 7, 15, 31, 63, and 127. These come from the CNN feature extractor before quantum encoding. Larger feature sizes can preserve more information but increase the complexity of the quantum circuit.
- Number of Qubits—Set based on the reduced feature size, with values 4, 7, 12, and 15. Each qubit represents one encoded feature.
- Quantum Depth—We fixed the variational circuit depth to 1 as discussed by Geng in [10]. Deeper circuits could improve model capacity, but they are time-consuming and more prone to noise, especially on current hardware.

4.2. Search Method—Population-Based Training (PBT)

We used PBT from the Ray Tune package to optimize hyperparameters efficiently. This method trains multiple models in parallel, each with different hyperparameters. After every perturbation interval, models are compared. The better ones are kept (“exploited”), while others are replaced or slightly modified (“explored”).

- Perturbation interval—every 5 epochs
- Resample probability—0.3 (i.e., 30% of models explore new values)
- Objective—Minimize training loss

PBT works well for our setup because it balances trying new settings and sticking with those that work. It is also adaptive as it tunes hyperparameters during training, which is useful in a noisy hybrid quantum setup.

4.3. Training Setup

All hyperparameter optimization experiments used the `PennyLane default.qubit` simulator, running on a local GPU machine. We used the Adam optimizer with cross-entropy loss and trained each configuration for 50 epochs. Random seeds and model checkpoints were used to improve reproducibility.

4.4. Best Configuration

As shown in Figure 9, the hyperparameter search reveals that the performance improves as the feature size increases from 7 to 63, after which gains slightly degrade, likely due to overfitting or encoding noise. Notably, the 12-qubit configuration achieves the highest accuracy and lowest loss at feature size 63; the results also show that 4 qubits produce comparable loss and accuracy values, with only a marginal performance difference. Considering the best trade-off between model complexity, qubit limitations of near-term NISQ hardware, and generalization capability, the optimal configuration was identified as a feature size of 63, 4 qubits, and a batch size of 16. This batch size was selected based on prior tuning, as it provided a stable balance between gradient estimation and convergence speed. This configuration was used for all subsequent benchmarking experiments.

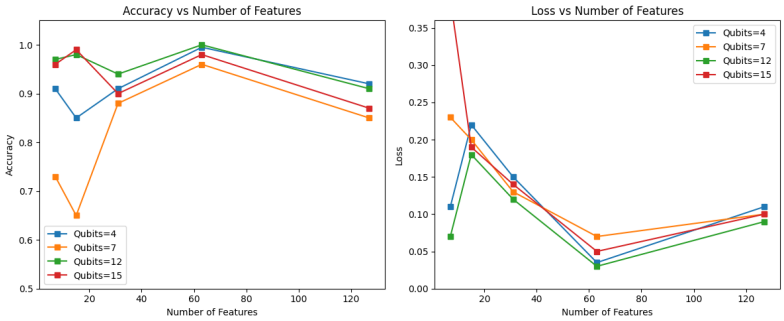


Figure 9. Hyperparameter tuning for the VQC-based classifier over feature sizes (7, 15, 31, 63, 127) and qubit configurations (4, 7, 12, 15), with a fixed batch size of 16.

5. Results

We evaluated the performance of three models, classical CNN, VQLS-enhanced QSVM, and VQC-based classifier, on binary (good vs. defective welds) and multiclass (three-class: good weld, contamination, and lack of fusion) classification tasks at a feature size of 63. The classical CNN, implemented following the architecture in [12], was trained using the Adam optimizer with a learning rate of 1×10^{-4} and batch size of 10. Its performance was measured using accuracy and loss metrics on a fixed test set of 400 images across different feature sizes $(7, 15, 31, 63, \text{and } 127)$. The VQLS-enhanced QSVM used feature vectors extracted from this CNN model, which were amplitude-encoded into quantum states to compute a quantum kernel matrix. The associated linear system was then solved variationally using a hardware-efficient ansatz with three qubits, optimized by the COBYLA algorithm to minimize the cost function described in the previous section. These experiments were performed on the `Qiskit Aer simulator` using 10,000 shots per circuit. For each feature size, training continued until the cost function converged or a maximum number of iterations was reached, and the model performance was again evaluated on the same test set. The VQC-based classifier followed a hybrid design where the CNN-extracted features were first linearly reduced in dimension and then embedded

into a four-qubit quantum circuit via H and $R_y(x_i)$ rotations. A shallow variational ansatz processed these inputs, and Pauli-Z expectation values were passed through a final classical linear-softmax layer for prediction. Training was conducted for 50 epochs with a batch size of 16 on the `PennyLane default.qubit` simulator using the Adam optimizer, cross-entropy loss, and the parameter-shift rule for gradient computation. The number of qubits was fixed at four, as determined through hyperparameter tuning discussed in the previous section.

5.1. Binary Classification

5.1.1. Classical CNN

The classical CNN model was evaluated on the binary classification task (good vs. defective welds) for different feature sizes. As shown in Table 1, the model achieved perfect accuracy at all feature sizes with a steady decrease in loss as the feature size increased.

Table 1. Binary classification results using the classical CNN model on the test set (400 images).

Feature Size	**7**	**15**	**31**	**63**	**127**
Accuracy	1.000	1.000	1.000	1.000	1.000
Loss	0.0204	0.0066	0.0145	0.0020	0.0003

5.1.2. VQLS-Enhanced QSVM

Table 2 shows the binary classification results for the VQLS-enhanced QSVM. The accuracy improved with larger feature sizes, peaking at 63 features with 96.8% accuracy and a loss of 0.2011. The performance dropped slightly at 127 features, likely due to circuit noise and overfitting. Training was performed until convergence, with the number of iterations varying depending on input properties such as the linear system's condition number (κ).

Table 2. Binary classification results using the VQLS-enhanced QSVM model on the test set (400 images).

Feature Size	**7**	**15**	**31**	**63**	**127**
Accuracy	0.753	0.856	0.871	0.968	0.948
Loss	0.454	0.395	0.341	0.201	0.238

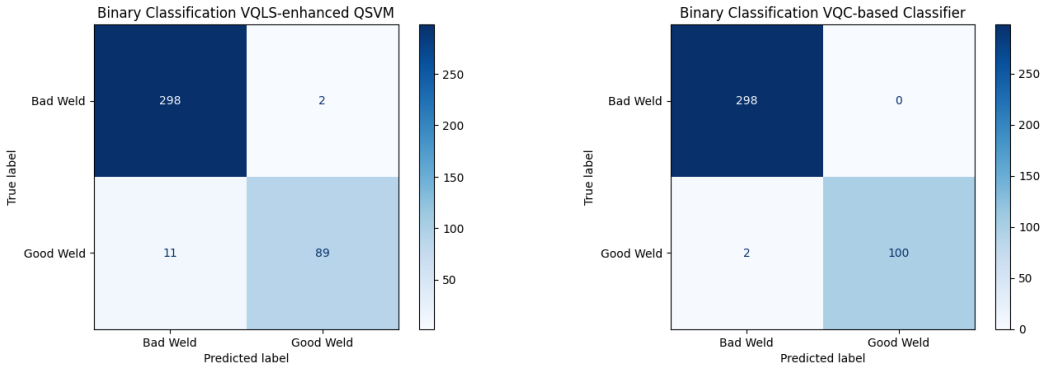
5.1.3. VQC-Based Classifier

The VQC-based classifier results in Table 3 demonstrated strong performance, with the accuracy increasing alongside the feature size and loss decreasing correspondingly. The best results were observed at a feature size of 63. Beyond this, accuracy slightly declined, likely due to specific feature characteristics impacting the model's effectiveness.

Table 3. Binary classification results using the VQC-based classifier model on the test set (400 images).

Feature Size	**7**	**15**	**31**	**63**	**127**
Accuracy	0.915	0.876	0.925	0.997	0.951
Loss	0.126	0.235	0.164	0.042	0.113

The confusion matrices for the binary classification task at feature size 63 are shown in Figure 10. The VQLS-enhanced QSVM exhibits a small number of misclassifications, whereas the VQC-based classifier achieves near-perfect separation, with almost all samples correctly assigned to their respective classes, confirming the accuracy values reported in the tables above.



(**a**) (**b**)

Figure 10. Confusion matrices for binary classification at feature size 63: (**a**) VQLS-enhanced QSVM; (**b**) VQC-based classifier.

5.2. Multiclass Classification

5.2.1. Classical CNN

For the three-class classification task (good weld, contamination, lack of fusion), the classical CNN achieves high accuracy for all feature sizes (see Table 4). The lowest loss is observed at a feature size of 63, suggesting that this size may provide an optimal trade-off between model complexity and generalization performance.

Table 4. Multiclass classification results using the classical CNN model on the test set (400 images).

Feature Size	7	15	31	63	127
Accuracy	1.000	0.9925	0.9975	1.000	0.9975
Loss	0.0326	0.0288	0.0056	0.0011	0.0309

5.2.2. VQLS-Enhanced QSVM Multiclass

Table 5 presents the results from the three-class classification for the VQLS-enhanced QSVM. Similar to the binary case, the three-class classification accuracy improved with larger feature sizes. The model achieved its highest accuracy of 92.4% at 63 features with the lowest loss. However, the overall accuracy dropped compared to the binary case, due to an increase in circuit complexity and kernel evaluation demands. As with the binary task, increasing the feature size beyond 63 did not yield further gains, supporting the conclusion that this size provides the best balance between information richness and circuit feasibility.

Table 5. Multiclass classification results using the VQLS-enhanced QSVM model on the test set (400 images).

Feature Size	7	15	31	63	127
Accuracy	0.625	0.700	0.772	0.924	0.884
Loss	0.684	0.482	0.437	0.359	0.384

5.2.3. VQC-Based Classifier Multiclass

The VQC-based classifier exhibited strong multiclass performance with an accuracy of 98.9% at 63 features (Table 6). Lower feature sizes led to reduced accuracy caused by insufficient information, while larger feature sizes introduced marginal performance drops, possibly due to data characteristics and overfitting effects.

Table 6. Multiclass classification results using the VQC-based classifier model on the test set (400 images).

Feature Size	7	15	31	63	127
Accuracy	0.890	0.892	0.942	0.989	0.946
Loss	0.203	0.209	0.126	0.107	0.115

The confusion matrices for the three-class task at feature size 63 are presented in Figure 11. The VQLS-enhanced QSVM shows a notable increase in misclassifications compared to the binary case, reflecting the added difficulty of distinguishing three classes with a kernel-based approach. In contrast, the VQC-based classifier retains its strong performance, demonstrating its ability to separate all three weld categories even in the complex multiclass scenario.

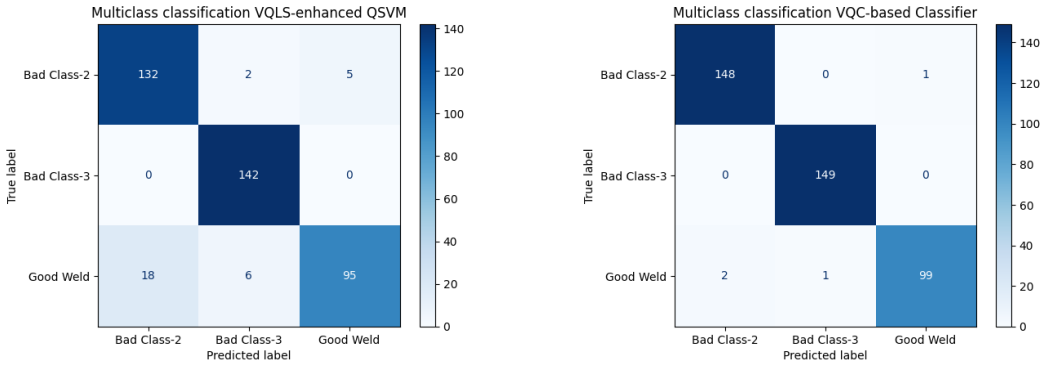


Figure 11. Confusion matrices for multiclass (3-class) classification at feature size 63: (**a**) VQLS-enhanced QSVM; (**b**) VQC-based classifier.

When comparing the three models of classical CNN, VQLS-enhanced QSVM, and the VQC-based classifier on identical binary and multiclass classification tasks, the classical CNN consistently achieved perfect or near-perfect accuracy across all feature sizes in both scenarios. This performance reflects its reliability and strong generalization capabilities for image classification problems. The VQC-based classifier achieved near-perfect accuracy, on par with the classical CNN, particularly at a feature size of 63 in both binary and multiclass tasks. Its shallow variational quantum circuit, combined with classical preprocessing and post-processing layers, enabled efficient learning and inference. This architecture proved to be well suited for near-term quantum (NISQ) devices, offering a good trade-off between quantum resource usage and performance. The VQLS-enhanced QSVM also delivered good accuracy in the binary classification task, with its best results observed at a feature size of 63. However, the training process was significantly more time-consuming due to the complexity of the variational quantum linear solver and the repeated circuit evaluations required for kernel matrix construction and cost minimization. These evaluations are performed iteratively until convergence, and the number of iterations can vary based on the input data properties such as the condition number (κ) of the linear system, which directly impacts the computational cost. Its performance declined in the multiclass scenario, primarily due to the increased circuit complexity and the higher dimensionality of quantum kernel evaluations required for distinguishing three classes.

Some of the observed variations in model performance across feature sizes can be due to the intrinsic characteristics of the dataset. Reflections from the weld head occasionally led to misclassifications, especially in the multiclass task, where some good welds visually resembled contamination (Class 2). However, contamination defects were typically well-localized and visually distinct, allowing all models to detect them reliably. In the case of

fusion defects (Class 3), the presence of a dark line in the weld center served as a critical visual cue (as shown in Figure 12), which helped both the VQC and CNN models maintain high classification performance.

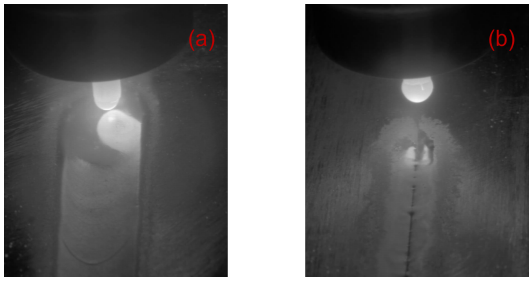


Figure 12. Visual cues influencing classification results. (**a**) Good weld with reflection head resembling contamination, and (**b**) lack of fusion defect with dark line in the center aiding detection.

6. Conclusions

In this study, we investigated the classification of welding images using hybrid quantum–classical algorithms on a high-resolution industrial dataset acquired from Tungsten Inert Gas (TIG) welding processes. We benchmarked three distinct pipelines: a classical convolutional neural network (CNN), a VQLS-enhanced quantum support vector machine (QSVM) and a hybrid variational quantum circuit (VQC)-based classifier. All three approaches used CNN-based feature extraction to reduce the high-dimensional image data into a manageable feature vector suitable for quantum encoding. This decision addresses the current hardware limitations of NISQ-era quantum devices. While the classical CNN achieved consistently high performance as expected, the hybrid quantum models also demonstrated competitive results. The VQC-based classifier outperformed VQLS-enhanced QSVM in binary and multiclass classification, offering a strong balance between accuracy and quantum circuit depth. The VQLS-enhanced QSVM showed promising results as well, but it faced practical limitations due to its circuit complexity and training overhead. Notably, the characteristics of the dataset also contributed to the high classification performance. Defects such as contamination and lack of fusion were visually distinct, which likely made the classification task more tractable across all models. Overall, our results indicate that hybrid quantum-classical architectures can already offer competitive performance on real-world industrial tasks such as weld-defect classification. As quantum hardware advances, such approaches could become increasingly practical and scalable for high-sensitivity applications such as non-destructive testing and automated industrial inspection.

Author Contributions: All authors—conceptualization, investigation, data curation, writing -original draft preparation, review and editing, visualization. All authors have read and agreed to the published version of the manuscript.

Funding: This work was partially funded by the Federal Ministry for Economic Affairs and Climate Action through the EniQmA project (funding number 01MQ22007A).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: All data and simulation materials used in this study are available from the corresponding author upon reasonable request, subject to any applicable restrictions.

Acknowledgments: This work was supported by the Quantum Initiative Rhineland-Palatinate (QUIP), by the Federal Ministry for Economic Affairs and Climate Action through the EniQmA project and by the European Union, Interreg Upper Rhine Valley program within the UpQuantVal

project. We also acknowledge the Fraunhofer-wide Capacity Building Campaign Quantum Now, which enabled us to perform experiments on both U.S.- and EU-based IBM Quantum systems.

Conflicts of Interest: Author Jianming Yi was employed by the company MuVision UG. The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

Abbreviations

The following abbreviations are used in this manuscript:

CNN	Convolutional Neural Network
NISQ	Noisy Intermediate-Scale Quantum
QSVM	Quantum Support Vector Machine
QML	Quantum Machine Learning
SVM	Support Vector Machine
TIG	Tungsten Inert Gas
VQC	Variational Quantum Circuit
VQLS	Variational Quantum Linear Solver
NDT	Non-Destructive Testing
HDR	High Dynamic Range
PBT	Population-Based Training
OvR	One-vs-Rest

References

1. Zhang, H.; Chen, Z.; Zhang, C.; Xi, J.; Le, X. Weld defect detection based on deep learning method. In Proceedings of the 2019 IEEE 15th International Conference on Automation Science and Engineering (CASE), Vancouver, BC, Canada, 22–26 August 2019; pp. 1574–1579.
2. Yaping, L.; Weixin, G. Research on X-ray welding image defect detection based on convolution neural network. *J. Phys. Conf. Ser.* **2019**, *1237*, 032005. [CrossRef]
3. Preskill, J. Quantum computing in the NISQ era and beyond. *Quantum* **2018**, *2*, 79. [CrossRef]
4. Alam, M.; Kundu, S.; Topaloglu, R.O.; Ghosh, S. Quantum-classical hybrid machine learning for image classification. In Proceedings of the 2021 IEEE/ACM International Conference on Computer Aided Design (ICCAD), Munich, Germany, 1–4 November 2021; pp. 1–7.
5. Biamonte, J.; Wittek, P.; Pancotti, N.; Rebentrost, P.; Wiebe, N.; Lloyd, S. Quantum machine learning. *Nature* **2017**, *549*, 195–202. [CrossRef] [PubMed]
6. Schuld, M.; Petruccione, F. *Machine Learning with Quantum Computers*; Springer: Cham, Switzerland, 2021; Volume 676. [CrossRef]
7. Geng, A.; Moghiseh, A.; Redenbach, C.; Schladitz, K. Hybrid quantum transfer learning for crack image classification on NISQ hardware. *AIP Conf. Proc.* **2025**, *3182*, 130001. [CrossRef]
8. Geng, A.; Moghiseh, A.; Redenbach, C.; Schladitz, K. A hybrid quantum image edge detector for the NISQ era. *Quantum Mach. Intell.* **2022**, *4*, 15. [CrossRef]
9. Geng, A.; Moghiseh, A.; Redenbach, C.; Schladitz, K. Improved FRQI on superconducting processors and its restrictions in the NISQ era. *Quantum Inf. Process.* **2023**, *22*, 104. [CrossRef]
10. Geng, A. Application of Hybrid Quantum Machine Learning for Image Processing in the NISQ Era. Ph.D. Thesis, Rheinland-Pfälzische Technische Universität Kaiserslautern-Landau, Kaiserslautern, Germany, 2024. . [CrossRef] [PubMed]
11. Srinivasan, A.; Geng, A.; Macaluso, A.; Kiefer-Emmanouilidis, M.; Moghiseh, A. A comparative study of Q-Seg, quantum-inspired techniques, and U-Net for crack image segmentation. *tm—Tech. Mess.* **2025**, *92*, 321–331. [CrossRef]
12. Bacioiu, D.; Melton, G.; Papaelias, M.; Shaw, R. Automated defect classification of Aluminium 5083 TIG welding using HDR camera and neural networks. *J. Manuf. Process.* **2019**, *45*, 603–613. [CrossRef]
13. Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; et al. PyTorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems*; Curran Associates, Inc.: Red Hook, NY, USA, 2019; Volume 32, pp. 8024–8035.
14. Russakovsky, O.; Deng, J.; Su, H.; Krause, J.; Satheesh, S.; Ma, S.; Huang, Z.; Karpathy, A.; Khosla, A.; Bernstein, M.; et al. ImageNet large scale visual recognition challenge. *Int. J. Comput. Vis.* **2015**, *115*, 211–252. [CrossRef]

15. Yi, J.; Suresh, K.; Moghiseh, A.; Wehn, N. Variational quantum linear solver enhanced quantum support vector machine. *arXiv* **2023**, arXiv:2309.07770. [CrossRef]
16. Bravo-Prieto, C.; LaRose, R.; Cerezo, M.; Subasi, Y.; Cincio, L.; Coles, P.J. Variational quantum linear solver. *Quantum* **2023**, *7*, 1188. [CrossRef]
17. Rebentrost, P.; Mohseni, M.; Lloyd, S. Quantum support vector machine for big data classification. *Phys. Rev. Lett.* **2014**, *113*, 130503. [CrossRef] [PubMed]
18. Ladický, L.; Torr, P.H.S. Locally linear support vector machines. In Proceedings of the 28th International Conference on Machine Learning (ICML), Bellevue, WA, USA, 28 June–2 July 2011; pp. 985–992.
19. Powell, M.J.D. A direct search optimization method that models the objective and constraint functions by linear interpolation. In *Advances in Optimization and Numerical Analysis*; Gomez, S., Hennart, J.P., Eds.; Springer: Dordrecht, The Netherlands, 1994; pp. 51–67. [CrossRef]
20. Bacioiu, D. TIG Aluminium 5083 Defect Image Dataset. Available online: https://www.kaggle.com/datasets/danielbacioiu/tig-aluminium-5083 (accessed on 12 June 2025).
21. Liaw, R.; Liang, E.; Nishihara, R.; Moritz, P.; Gonzalez, J.E.; Stoica, I. Tune: A research platform for distributed model selection and training. *arXiv* **2018**, arXiv:1807.05118. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.