Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

AFCC: Adaptive Fusion of Local Contour Descriptors and Hierarchical Convolutional Features for Weld Formation Classification on Curved Workpieces

Cheng-Hui Chen

chchen@ncut.edu.tw

National Chin-Yi University of Technology https://orcid.org/0000-0002-9492-820X

Tsung-Hua Tsai

National Chin-Yi University of Technology

Research Article

Keywords: Adaptive Feature Fusion, Weld Formation Classification, Curved Workpieces, Industrial Visual Inspection, Multi-Level Visual Features

Posted Date: November 6th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-11465596/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

AFCC: Adaptive Fusion of Local Contour Descriptors and Hierarchical Convolutional Features for Weld Formation Classification on Curved Workpieces

Cheng-Hui Chen[1*], Tsung-Hua Tsai[1]

[1] Department of Computer Science and Information Engineering, National Chin-Yi University of Technology, Taichung 411030, Taiwan.
*Corresponding author: chchen@ncut.edu.tw

Abstract.
Reliable visual inspection of welds on curved workpieces remains challenging because non-planar geometry, viewpoint variation, and non-uniform reflection cause the same weld structure to appear inconsistently across images. This paper proposes AFCC, an adaptive fusion framework that combines local contour descriptors with hierarchical convolutional features to classify weld formation as normal, excessive/distorted, or insufficient. Hu moment descriptors computed over sliding windows of an edge-enhanced image characterize localized bead geometry, while five groups of intermediate EfficientNet-B0 embeddings represent edge, texture, structural, texture-and-saturation, and high-level semantic information. Rather than concatenating the streams or assigning equal weights, AFCC weights each stream according to its chance-corrected balanced accuracy on the validation folds of five-fold cross-validation. The framework was evaluated on 1,586 weld images acquired at-line in an industrial metal-rack production facility, including a held-out test set of 477 images. With a Siamese neural network classifier, AFCC achieved 95.39% accuracy, 94.32% macro recall, and a 92.31% macro F1 score. Its macro F1 score exceeded those of a fine-tuned EfficientNet-B0 baseline and a representative deep-feature fusion method by 13.95 and 7.59 percentage points, respectively. AFCC missed only 4 of the 120 defective test welds, compared with 14 and 8 for these baselines, and its three-class output can route each weld to grinding, supplementary welding, or pass-through. Ablation results further indicated that the contour and convolutional representations are complementary and that reliability-guided weighting improves defect detection beyond equal weighting.

Keywords: Adaptive Feature Fusion, Weld Formation Classification, Curved Workpieces, Industrial Visual Inspection, Multi-Level Visual Features

1 Introduction

Welded metallic components are widely used in semiconductor process equipment, machine tools, structural frames and racks, and automated production systems, appearing in vacuum chambers, pipelines, frames, enclosures, and structural supports. Many of these components have cylindrical, arc-shaped, or otherwise non-

planar surfaces. In high-vacuum and precision equipment, welded joints must meet strict requirements for material compatibility, assembly accuracy, and surface quality [1], and weld-related failures can compromise the structural performance of these assemblies [2]. Weld quality is thus closely tied to equipment reliability. As component geometries become more complex and production volumes increase, manual inspection alone can no longer keep pace, and automated visual inspection has become an important means of maintaining consistent weld quality.

Welding parameters such as heat input, wire-feed speed, and travel speed affect the amount and distribution of deposited material, and thus the weld bead's geometry [3]. Normal formation (C_N) exhibits a continuous, complete weld bead; excessive or unstable deposition produces enlarged or irregular bead geometry (C_{ED}); and insufficient deposition results in incomplete material coverage and reduced bead volume (C_{IN}). The dataset examined in this paper covers these three conditions, which are defined by the external boundary, material distribution, and overall geometry of the weld. The scope of this study is limited to these externally observable conditions; internal defects, such as porosity or cracking, fall outside this scope.

Weld-quality inspection has been approached through image- and signal-based monitoring, multi-scale segmentation, and transfer learning: Convolutional Neural Network-Long Short-Term Memory (CNN-LSTM) networks have been used to analyze sequential weld-pool images [4], multi-scale networks have supported weld-defect segmentation [5], signal-based analysis has enabled rail-weld defect detection [6], and transfer learning has improved weld-defect classification when training data are limited [7]. These studies confirm the feasibility of image-based inspection, yet the effect of surface curvature on feature representation and classification has received comparatively little attention. On curved or non-planar surfaces, geometric projection, viewpoint variation, and spatially varying reflection can make the same weld structure appear quite different from one image to the next. Non-uniform illumination, spatter, and incomplete weld boundaries add further inconsistency [8], to the point that even normal welds may look substantially different across images. A representation used for curved-surface inspection must therefore remain informative under this variability rather than depending on a single global cue.

Local contour descriptors and hierarchical convolutional features describe the same weld region from two complementary perspectives: the former characterizes the weld boundary, material distribution, and localized geometric variation, while the latter captures edges, texture, spatial structure, and higher-level appearance. Edge detection provides a basis for extracting structural boundaries [9], while Hu moments offer compact shape descriptors with invariance to translation, scaling, and rotation [10], and related invariant shape descriptors have been applied to characterize weld defects [11]; when computed from localized weld regions, they can characterize local variations in weld geometry. Contour information is nonetheless sensitive to reflection, spatter, background noise, and boundary fragmentation, and benefits from convolutional features that retain texture and appearance information. Features from multiple convolutional layers preserve different levels of abstraction, from fine edges to higher-level semantic patterns [12], avoiding the loss of fine-grained boundary information that can occur when only the final layer is used. Limited-data learning has motivated

regularization of GAN discriminators [13] and feature refinement for few-shot SAR classification [14]. Although these tasks differ from weld inspection, they illustrate the need to control overfitting and redundant features when labeled data are scarce.

Although local shape descriptors and convolutional representations are complementary, they do not necessarily contribute equally to weld-formation discrimination. Dual-stream and multi-feature fusion strategies demonstrate the value of combining complementary representations [15], but direct concatenation and fixed or equal weighting do not account for differences in the validated discriminative reliability of individual feature streams. This limitation is especially important for curved metallic workpieces: reflections and fragmented edges can weaken contour descriptors, whereas convolutional features may underrepresent localized bead deformation. A fusion method should therefore preserve both perspectives while preventing a weak stream from dominating the final representation.

To address this gap, this paper proposes AFCC, an Adaptive Fusion of Local Contour Descriptors and Hierarchical Convolutional Features framework for weld formation classification on curved workpieces. AFCC jointly represents localized weld geometry and multi-level visual appearance. The contribution of each stream is determined from its classification performance on the validation folds, so the fusion weights reflect the overall discriminative reliability of the corresponding representation rather than a manually predetermined or equal allocation. These stream-level weights are fixed after validation and applied consistently during inference.

The contributions of this paper are summarized as follows:

- **Complementary feature representation:** Local contour geometry and hierarchical convolutional features are jointly represented to characterize boundary, texture, spatial, and higher-level appearance variations in curved-surface welds.
- **Performance-guided adaptive fusion:** The contribution of each feature stream is determined from its classification performance on the validation folds, avoiding manually predetermined or equal fusion weights and improving the recall of the two minority defect classes over equal weighting.
- **Three-class recognition on production weld images:** AFCC is evaluated on images acquired at-line in an industrial facility and distinguishes C_{N}, C_{ED}, and C_{IN} weld formation, which provides more decision-relevant information than a binary normal/defective decision.

The remainder of this paper is organized as follows. Section 2 reviews related work on edge-aware preprocessing, contour descriptors, convolutional representations, feature fusion, and language-model assistance in industrial inspection. Section 3 presents the AFCC framework together with a decision-gated explanation module. Section 4 reports the experimental setup, classification results, error analysis, and a resource analysis of the explanation module. Section 5 concludes the paper.

2 Related Work

2.1 Preprocessing and Edge-Aware Enhancement

Edge enhancement can support structural feature extraction. For example, edge-aware processing has been studied for keypoint matching in Synthetic Aperture Radar (SAR) images [16]. Morphological concepts have also supported photovoltaic fault diagnosis through dilation- and erosion-based clustering [17]. For weld-boundary extraction, grayscale conversion simplifies the image representation, while Gaussian smoothing reduces noise before edge detection [18]. Canny edge detection includes non-maximum suppression (NMS) along the gradient direction [9]. Efficient algorithms for the more general NMS operation have also been studied [19].

Industrial defects, however, often vary in appearance and include blurry textures, irregular surfaces, or low-contrast patterns without clear edge boundaries. In such cases, edge-based features alone may not provide sufficient discriminatory power, motivating the shape-level and appearance-level representations examined next.

2.2 Shape-Invariant Contour Features and Geometric Moments

Geometric moments are a well-established means of describing object shape [10]. Among them, Hu moments, known for their invariance to translation, scaling, and rotation, have been widely applied in content-based image retrieval [20], and invariant shape descriptors have likewise been used to characterize and retrieve weld defects in radiographic images [11]. Because such descriptors are computed analytically rather than learned, they remain applicable when labeled training data are scarce. However, moment-based methods such as Hu moments mainly capture global shape descriptors and may lack sensitivity to fine-grained textures or multi-scale spatial variations, which limits their performance in recognizing complex industrial defects with subtle and localized irregularities.

2.3 Hierarchical Representations from Convolutional Networks

Convolutional embeddings refer to high-level feature representations obtained from CNNs, which encode semantic, spatial, and contextual information within an image. These embeddings are typically extracted from intermediate or output layers of CNN architectures such as ResNet, Inception-ResNet, and EfficientNet, and are widely used in image classification, recognition, and segmentation tasks [21–23]. Their hierarchical structure and spatial sensitivity make convolutional embeddings effective for visual recognition, particularly when sufficient training data are available.

Data augmentation and network design are commonly used to improve learned representations. An improved multidimensional normalization ResNet has been applied to rotating-machinery fault diagnosis [24]. In stainless-steel pipe weld anomaly detection, a dual-flow feature enhancement network processes illumination-enhanced and original images in parallel and fuses their features [15]. This provides an example of combining complementary image representations, although its two streams differ from the contour and convolutional branches considered here.

At the same time, deeper convolutional layers encode increasingly class-specific, object-level patterns, whereas shallower layers respond mainly to edges and textures [12]. In tasks driven by low-level defect patterns, relying solely on deep-layer embeddings, especially in deeper models such as EfficientNet, may introduce semantic information that is irrelevant to the defect and reduce classification precision when training data are limited.

2.4 Feature Fusion Strategies and Adaptive Decision Models

Feature fusion and metric learning have both been used to improve classifier adaptability when training data are limited. Siamese Neural Networks (SNN), originally proposed for signature verification, learn a similarity metric between paired inputs and thereby support classification under low-data conditions [25]. A Siamese architecture built on spiking neurons has further achieved noise-robust SAR image classification through metric learning [26]. Few-shot learning has also been applied in such settings; for example, a feature refinement network has been proposed for few-shot SAR image classification when labeled samples are scarce [14].

Among fusion strategies, a dual-stage deep feature fusion approach that integrates multiple CNN features at both the feature and decision levels has reported a 3.58% improvement over state-of-the-art methods in waste classification [27], suggesting that multi-level fusion of deep features may also benefit visual quality-control tasks such as weld inspection. These studies indicate that feature fusion and metric-learning classifiers such as SNN can each improve classification stability when data are limited; however, these fusion schemes typically assign fixed, learned, or equal stream contributions and do not derive them explicitly from the chance-corrected validation reliability of each stream. This study addresses this gap by weighting the geometric stream and each semantic-group stream according to its validated reliability and comparing the result with equal weighting.

2.5 Language-Model Assistance in Industrial Inspection

Recent work has examined resource-efficient adaptation of vision-language models for multimodal defect inspection in manufacturing [28], together with broader efficiency-oriented designs for multimodal large language models [29]. Hallucination and label-inconsistent explanations nevertheless remain concerns: independently extracted visual knowledge can support post-generation correction [30], whereas grounding objectives alone do not consistently prevent hallucination in open-ended generation [31]. In this study, language-model assistance is treated as an optional downstream interface rather than part of the AFCC classifier. It receives the AFCC prediction and selected visual representations only for samples flagged as defective. Its resource demand is analyzed in Section 4.5.3, whereas the quality of the generated explanations is left for future evaluation.

3 Proposed Method: Adaptive Fusion of Local Contour Descriptors and Hierarchical Convolutional Features (AFCC)

AFCC targets weld formation classification under contour ambiguity and structural noise. The overall architecture is illustrated in Fig. 1.

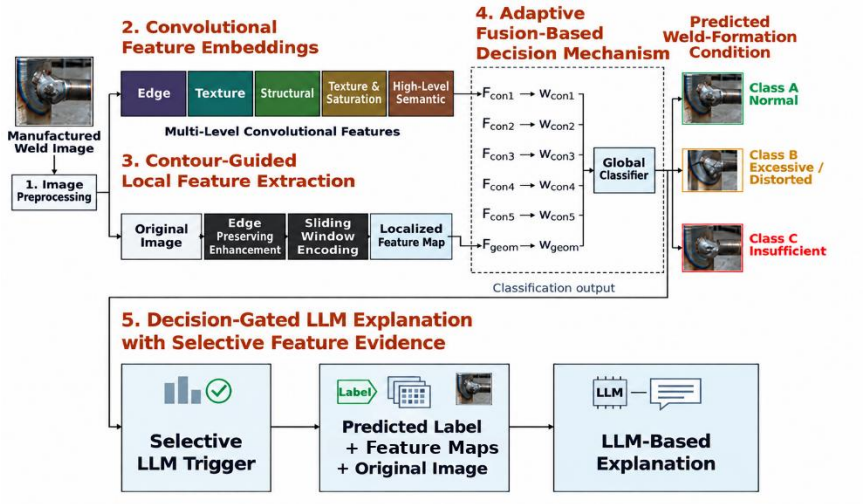


Fig. 1 AFCC architecture (A = C_N, B = C_{ED}, C = C_{IN})

Given an input weld image x, the core objective is to predict its weld-formation class y ∈ {C_N, C_{ED}, C_{IN}} (Section 1). The AFCC classifier itself is not specific to this taxonomy because the class definitions enter through the training labels. As illustrated in Fig. 1, the core architecture comprises preprocessing, two parallel feature-extraction branches, performance-guided fusion, and SNN classification. An optional fifth stage uses the predicted class to gate downstream language-model assistance for samples classified as C_{ED} or C_{IN}; this stage does not alter the AFCC prediction.

3.1 Image Preprocessing

Because the images are acquired on the production floor, they may contain interference such as oil residues, iron filings, and surface contaminants, so preprocessing is applied before feature extraction. For the contour-guided branch (Section 3.2), the color image is converted to grayscale to focus on shape-related features and then smoothed with a Gaussian filter to suppress noise before edge detection. The convolutional branch (Section 3.3) receives the RGB image, so that color and texture cues remain available to the backbone.

3.2 Contour-Guided Local Feature Extraction

Welding defects frequently present as localized contour distortions. To capture such irregularities, the contour-guided branch uses the three-stage pipeline summarized in Fig. 2.

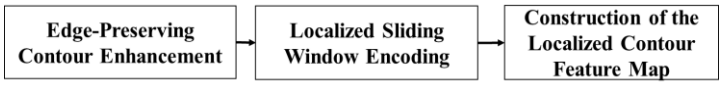


Fig. 2 Contour-guided local feature extraction

The process begins with Edge-Preserving Contour Enhancement, which applies Canny edge detection followed by morphological filtering to emphasize weld boundaries while reducing noise. Next, Localized Sliding Window Encoding partitions the edge image into fixed-size patches, each encoded using Hu moment descriptors to capture shape-invariant features. Finally, Localized Contour Feature Map Construction aligns descriptors with their spatial coordinates, forming a structured representation suitable for downstream classification and regional defect analysis.

3.2.1 Edge-Preserving Contour Enhancement

Weld contours are detected with the Canny edge detector [9]. The process begins with gradient estimation, where the standard 3×3 Sobel operator is applied in the horizontal and vertical directions to obtain the gradients G_x and G_y of the grayscale image at each pixel location (x, y).

The gradient magnitude G at each pixel is then computed as Equation (1).

$$G = \sqrt{G_x^2 + G_y^2} \tag{1}$$

This magnitude map is refined via NMS to preserve local maxima along the gradient direction, followed by dual-thresholding to classify edge responses into strong, weak, and non-edge pixels. Edge tracking by hysteresis is then applied to ensure contour continuity.

To address fragmented or noisy weld boundaries, morphological dilation is applied to fill local discontinuities and enhance edge connectivity, followed by erosion to complete morphological closing. The dilation operation is defined as Equation (2).

$$I_{dilated} = I_{edge} \oplus S \tag{2}$$

where I_{edge} represents the binarized image after edge detection, $\oplus$ denotes morphological dilation, and S denotes the structuring element, which determines the dilation range. The erosion step contracts the dilated regions while preserving connections established by closing. The erosion operation is defined as Equation (3).

$$I_{eroded} = I_{dilated} \ominus S \tag{3}$$

where I_{eroded} represents the processed image after erosion, and $\ominus$ denotes the mathematical operation of erosion.

Dilation followed by erosion constitutes morphological closing, which fills small gaps and improves local boundary connectivity before window-based encoding.

3.2.2 Localized Sliding Window Encoding

To improve the sensitivity of feature extraction to region-specific welding defects, the edge-enhanced image is partitioned into *M* fixed-size overlapping subregions using a sliding window approach. Each subregion is denoted as $I_{\mathrm{k}}(x, y)$, where $k \in 1,2, \ldots, M$represents the index of the window, and(x, y) denotes the pixel coordinate within the window of dimensions $w \times w$. For each window *k*, the image centroid $(\bar{x}, \bar{\mathrm{y}})$ is first computed from the raw spatial moments. The central moment $\mu_{pq}(k)$, which captures the distribution of intensity around the centroid, is defined as Equation (4).

$$\mu_{pq}(k) = \sum_{x}\sum_{y}(x - \bar{x})^{p}(y - \bar{\mathrm{y}})^{q}\, I_{\mathrm{k}}(x, y) \tag{4}$$

where $I_{\mathrm{k}}(x, y)$represents the pixel intensity at coordinate (*x, y*) in the k-th window, $\bar{x}$, $\bar{\mathrm{y}}$ denote the centroid coordinates of window I_{k}, and *p*, *q* are non-negative integers indicating the moment order.

To ensure invariance to image scale, the normalized central moment $\eta_{pq}(k)$ is computed as Equation (5).

$$\eta_{pq}(k) = \frac{\mu_{pq}(k)}{\mu^{00}(k)^{\left(1+\frac{p+q}{2}\right)}} \tag{5}$$

where $\mu^{00}(k)$ denotes the zeroth-order central moment, equivalent to the total intensity of the window. These normalized central moments are then combined into seven scalar values based on Hu's formulation to form a compact 7-dimensional shape descriptor $h(k) \in \mathbb{R}^{7}$ that captures geometric invariants of the local region.

3.2.3 Construction of the Localized Contour Feature Map

Each local descriptor $h(k)$ is assigned to its spatial location within the image via the centroid coordinates$(x_{\mathrm{k}}, y_{\mathrm{k}})$ of the corresponding window. This mapping results in a structured set of tuples represented as Equation (6).

$$\mathcal{H} = \left(x_{\mathrm{k}}, y_{\mathrm{k}}, h(k)\right) | k = 1,2, \ldots, M \tag{6}$$

where $x_{\mathrm{k}}, y_{\mathrm{k}}$ denote the spatial center coordinates of the k-th window, $h(k) \in \mathbb{R}^{7}$ is the Hu moment-based geometric descriptor for window k, and M denotes the total number of sliding windows.

The set forms a localized contour feature map in which each Hu descriptor remains associated with the center of its source window. This representation retains spatial correspondence for visualization and regional analysis. For classification, however, the M local descriptors are aggregated into the geometric stream described in Section 3.4; consequently, the classifier uses the distribution of localized shape responses but does not explicitly model their absolute ordering along the weld seam.

3.3 Feature Extraction via Convolutional Embeddings

To represent weld appearance at multiple levels, an EfficientNet-B0 backbone [23] is adopted, with the final pooling and fully connected layers omitted so that intermediate feature maps remain available. The maps are organized into five semantic groups according to network depth and their dominant visual responses: edge, texture, structural, texture-and-saturation, and high-level semantic features. These names provide an operational description of the observed activation patterns rather than a claim that each stage encodes only the stated property. The multi-stage extraction process is formalized in Equation (7).

$$F_i = f_{i(I_{rgb})}, \qquad where\ i = 1, 2, \dots, L \tag{7}$$

where $f_{i(\cdot)}$ represents the output feature map from the i-th block of the CNN backbone encoder, L is the number of encoder blocks, and I_{rgb} denotes the RGB input image (Section 3.1).

The grouping of feature maps into five semantic groups is defined as Equation (8).

$$G_j = \{ F_i \mid i \in S_j \}, \ where\ j = 1, 2, \dots, 5 \tag{8}$$

where S_j denotes the set of encoder stages associated with the j-th semantic group. In this study, the stem forms the edge group, stage 1 the texture group, stages 2–3 the structural group, stages 4–5 the texture-and-saturation group, and stages 6–7 the high-level semantic group, so that shallow stages supply edge and texture information and deeper stages supply semantic abstractions.

The CNN backbone encoder's architectural configuration, including its mapping onto the five semantic groups that feed the adaptive fusion mechanism introduced in Section 3.4, is illustrated in Fig. 3.

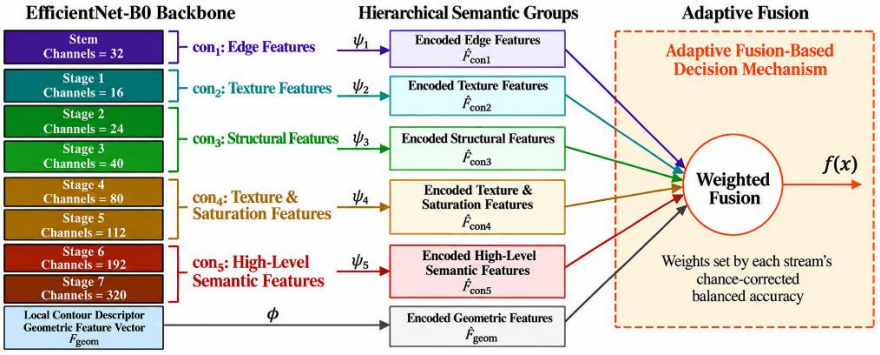


Fig. 3 EfficientNet-B0 stages and the five semantic groups

Each semantic group is treated as a separate feature stream in the fusion stage (Section 3.4); together with the geometric stream produced by the contour-guided branch, AFCC fuses six streams in total.

3.4 Adaptive Fusion-Based Decision Mechanism

The aim of this stage is to integrate the handcrafted geometric descriptors from Section 3.2 and the grouped convolutional embeddings from Section 3.3 into a common representation for weld-formation classification. Let $F_{\text{geom}} \in \mathbb{R}^7$ denote the handcrafted Hu moment feature vector, obtained by average pooling the M local window descriptors in Equation (6); this aggregation summarizes the distribution of local contour responses and intentionally removes their absolute coordinates. Let con_i, $i = 1, 2, \ldots, 5$, denote the d_i-dimensional convolutional feature vectors obtained by global average pooling the feature maps of the five semantic groups in Equation (8): edge, texture, structural, texture-and-saturation, and high-level semantic features. Each semantic-group stream con_i is projected to a latent vector of common dimension by an encoder $\psi_i(\cdot)$, and the geometric stream by a separate encoder $\varphi(\cdot)$, as defined in Equation (9).

$$\hat{F}_{geom} = \varphi(F_{geom}), \qquad \hat{F}_{con_i} = \psi_i(con_i), \qquad i = 1, 2, \ldots, 5 \tag{9}$$

To determine the fusion weights, a separate SNN classifier is trained for each feature stream, that is, one for the geometric stream and one for each of the five semantic groups, using the encoded features of that stream alone. Each stream-specific classifier is then evaluated on the validation folds, so that the weights reflect how discriminative each stream is under the same classifier type that performs the final prediction. Denote $\rho_{geom} \in [0, 2/3]$ as the chance-corrected score of the geometric stream, and $\rho_{con_i} \in [0, 2/3]$ as that of the i-th semantic group. For a stream s, the score is $\rho_s = \max(BA_s - 1/K, 0)$, where BA_s is the balanced accuracy of the stream-specific classifier on the validation folds, i.e., its recall averaged over the $K = 3$ classes [32], and $1/K$ is

the balanced accuracy expected from a classifier whose predictions are independent of the true class. The normalized fusion weights are computed as Equation (10).

$$w_{geom} = \frac{\rho_{geom}}{\left(\rho_{geom} + \sum_{i=1}^{5} \rho_{con_i}\right)}, \quad w_{con_i} = \frac{\rho_{con_i}}{\left(\rho_{geom} + \sum_{i=1}^{5} \rho_{con_i}\right)} \tag{10}$$

where w_{geom}, w_{con_i} are the relative contributions of the geometric stream and the i-th semantic group to the fused representation.

This chance-corrected formulation acts as a soft weighting with two properties. First, overall accuracy is not used as the reliability measure because, in this dataset, about three quarters of the samples are C_N, so a stream that assigns every sample to C_N would still reach an accuracy of about 75%; balanced accuracy weights the three classes equally and is consistent with the macro-averaged metrics used for evaluation. Second, a classifier whose predictions are independent of the true class attains an expected balanced accuracy of $1/K$, because its per-class recalls equal the probabilities of predicting each class and these sum to one. Subtracting $1/K$ therefore removes the score that is attainable without discriminative information, following the same principle as Cohen's κ [33], so that a stream at chance level receives zero weight and the weights of informative streams are separated more clearly than with raw accuracy. Provided that at least one stream exceeds chance level, the weights are non-negative and sum to unity, so the fusion in Equation (11) is a valid convex combination of the encoded streams. Because $1/K$ is fixed by the number of classes, the weighting introduces no additional hyperparameters and is recomputed from the data whenever the classifier or dataset changes. Adaptation therefore occurs when the model is calibrated on validation data, and the resulting weights remain fixed during inference.

The fused representation is then obtained as the weighted sum of all encoded features, as defined in Equation (11).

$$F_{fused} = w_{geom} \cdot \hat{F}_{geom} + \sum_{i=1}^{5} w_{con_i} \cdot \hat{F}_{con_i} \tag{11}$$

The fused vector F_{fused} is then classified by a global SNN [25] (Fig. 1). Each encoder is trained together with its stream-specific SNN and then frozen; the global SNN is subsequently trained on the resulting fused vectors. Because the encoders are trained separately, the weighted sum does not assume aligned latent coordinates; it acts as a fixed, reliability-weighted mixing of the stream embeddings, from which the global SNN learns the class structure of the combined representation. The ablation study in Section 4.5.1 supports this design empirically.

For each input sample x, the global SNN computes non-negative pairwise similarity scores against reference samples from each class C_k. The class-level score $s_k(x)$ is the mean of the pairwise similarity scores for the reference samples belonging to C_k. These class-level scores are normalized to sum to one, $p_k(x) = s_k(x) / [s_1(x) + \ldots + s_K(x)]$,

and are interpreted as normalized similarity scores rather than calibrated posterior probabilities. The predicted label is the class with the highest normalized score, as defined in Equation (12).

$$\hat{y}(x) = argmax_{\mathrm{k}} p_{\mathrm{k}}(x) \tag{12}$$

Beyond producing the classification decision, the predicted class can support routing to class-specific remediation in an industrial inspection workflow. C_{ED} predictions can be directed to localized grinding, C_{IN} predictions to supplementary filler welding, and C_{N} predictions to normal pass-through. The present study evaluates the classification stage only; robotic positioning, repair throughput, and post-repair effectiveness are not experimentally assessed. These actions should be interpreted as intended downstream uses of the AFCC output rather than as validated closed-loop repair performance.

3.5 Decision-Gated Explanation Module

An optional downstream module is considered to illustrate how AFCC outputs could support natural-language assistance without invoking a large language model (LLM) for every inspected image. This module operates after the classifier in Sections 3.1–3.4 and does not affect the predicted weld-formation class.

The explanation module is governed by a decision-gated trigger function that opens only when the highest-scoring predicted class is C_{ED} or C_{IN}, as defined in Equation (13).

$$T(x) = 1 \text{ if } \hat{y}(x) \neq C_{\mathrm{N}}, \text{ and } T(x) = 0 \text{ otherwise} \tag{13}$$

Samples classified as C_{N} ($T(x) = 0$) incur no language-model invocation, so the number of requests equals the number of welds flagged for remediation.

For each triggered sample, the language model receives the original image x, the predicted label, and visual representations V(x), as defined in Equation (14). V(x) contains the feature maps of streams whose fusion weights in Equation (10) exceed the uniform value 1/6. For the geometric stream, the localized contour feature map is submitted; for a semantic-group stream, the corresponding group feature map is submitted. The maps are rendered and downscaled before transmission. Because the fusion weights are estimated at stream level and fixed after validation, the same stream types are selected for every triggered sample. They indicate which representations were more reliable across the validation data, but they are not sample-specific attribution maps and should not be interpreted as a faithful causal explanation of an individual prediction.

$$e(x) = \mathrm{LLM}\big(x, \hat{y}(x), V(x)\big) \text{ if } T(x) = 1, \text{ and } e(x) = \emptyset \text{ if } T(x) = 0 \tag{14}$$

Because e(x) is conditioned on the AFCC prediction, the generated text is expected to remain label-consistent, but it may also rationalize an incorrect prediction. The sup-

plied representations provide additional task context, not proof of explanation faithfulness. Although grounding and external visual knowledge have been investigated as ways to reduce hallucination [30, 31], the present study does not evaluate hallucination, factual correctness, or expert usefulness of the generated explanations.

Efficiency formulation. Let D denote a deployment set of N inspected images. An all-image baseline submits every image in full to a general-purpose language model for joint classification and explanation, incurring N invocations. The comparison concerns invocation policies and does not establish equivalent classification or explanation quality. Two quantities characterize the resource cost of this process: the average payload size and the average end-to-end latency per inspected image.

For each image x, the all-image call transmits the original image with its image-file payload $c_{\mathrm{img}}(x)$ and returns both a classification and an explanation. The average payload, $B_{\mathrm{avg}}^{\mathrm{all}}$, and the average end-to-end latency, $L_{\mathrm{avg}}^{\mathrm{all}}$, of this baseline are defined in Equation (15).

$$\begin{aligned} \mathrm{B}_{\mathrm{avg}}^{\mathrm{all}} &= \frac{1}{N}\sum_{x\in\mathcal{D}} \mathrm{c}_{\mathrm{img}}(x) \\ \mathrm{L}_{\mathrm{avg}}^{\mathrm{all}} &= \frac{1}{N}\sum_{x\in\mathcal{D}} \mathrm{L}_{\mathrm{LLM}}(x) \end{aligned} \tag{15}$$

In contrast, AFCC invokes the language model only for the samples for which the gate opens ($T(x) = 1$), and each triggered call carries the original image together with the predicted label and the feature-map evidence $V(x)$. The corresponding averages per inspected image for AFCC under this policy are defined in Equation (16).

$$\begin{aligned} B_{avg}^{AFCC} &= \frac{1}{N}\sum_{x\in\mathcal{D}} T(x)\cdot\left[c_{img}(x) + c_{label}(x) + c_V(x)\right] \\ \mathrm{L}_{\mathrm{avg}}^{\mathrm{AFCC}} &= \frac{1}{N}\sum_{x\in\mathcal{D}} \left[\mathrm{L}_{\mathrm{cls}}(x) + T(x)\cdot \mathrm{L}_{\mathrm{LLM}}(x)\right] \end{aligned} \tag{16}$$

where $c_{\mathrm{label}}(x)$ and $c_{\mathrm{V}}(x)$ denote the payload sizes of the predicted label and the downscaled feature maps, respectively, in the same unit as $c_{\mathrm{img}}(x)$. $L_{\mathrm{cls}}(x)$ is the AFCC recognition latency incurred for every inspected sample, and $L_{\mathrm{LLM}}(x)$ is the measured request elapsed time from request submission to receipt of the complete response, including upload and network communication.

4 Results and Discussion

This section evaluates AFCC on an industrial weld dataset and discusses its performance and limitations.

4.1 Dataset Description

The dataset employed in this study was acquired from a large-scale industrial manufacturer specializing in metal rack production. All images were captured using a Photonic Science HAWKeye sCMOS industrial camera. Annotation was performed by experienced welding technicians. The inspected welds were located at the corner joints of the racks, which are structurally critical to overall strength. Images were acquired in an at-line fashion as part of the facility's regular inspection workflow, rather than under separately staged laboratory conditions, so that the lighting, surface contamination, and pose variability captured in the dataset are representative of authentic production-floor conditions. The structure and inspection locations of the welded joints are depicted in Fig. 4.

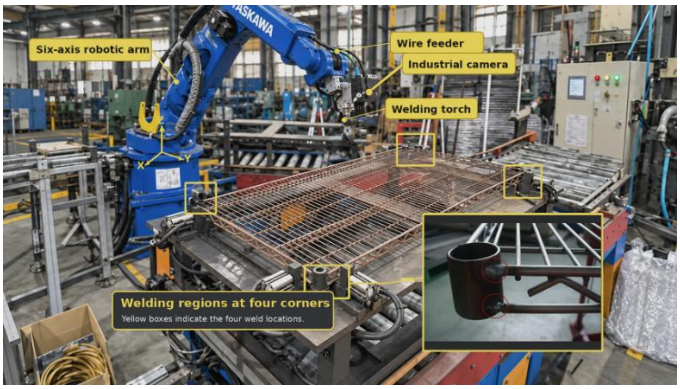


Fig. 4 Welded joints and inspection regions

The inspected rack joints connect structural members to a curved corner post, so the surface normal and apparent bead geometry vary across the inspection region. Figure 5 provides a conceptual illustration of this curvature-induced change in surface orientation. It is not intended to reproduce the exact joint geometry or weld trajectory of every specimen in the dataset. The schematic is included only to show why viewing angle and reflection vary more strongly on a curved support than on a planar plate.

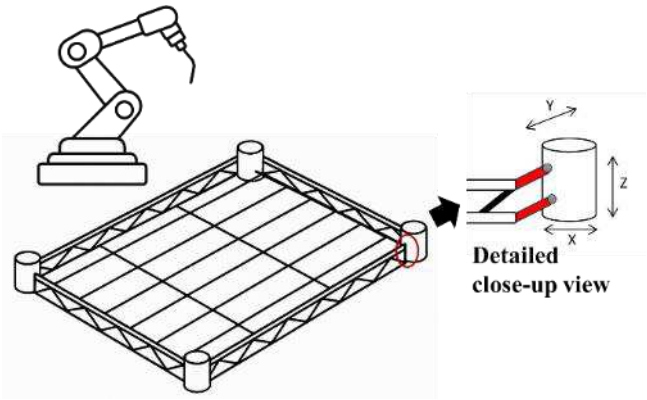


Fig. 5 Weld orientation on a curved workpiece (schematic)

In Fig. 5, the marked paths and fusion zones illustrate how a weld can extend across regions with changing surface orientation. The actual dataset consists of the rack-corner welds shown in Fig. 4 rather than generic pipe or pressure-vessel seams.

To construct the dataset, each weld sample was captured using single-point imaging at a fixed distance of 10 cm. The resolution of the raw images was 1920 × 1080 pixels. C_N welds typically exhibit a continuous, saturated, and near-elliptical morphology, whereas C_{ED} welds show an enlarged or irregular bead with protrusions beyond the expected joint boundary, and C_{IN} welds show a reduced bead with incomplete material coverage. Based on these morphological differences, all samples were labeled into three categories: C_N, C_{ED}, or C_{IN}. Because each weld-formation category constitutes a distinct inspection outcome, classification performance is assessed using a one-vs-rest formulation applied to each class individually, as detailed in Section 4.3. Representative examples of each class are shown in Fig. 6.



(A) (B) (C)

Fig. 6 Representative samples: (A) C_N, (B) C_{ED}, (C) C_{IN}

The dataset size reflects production lines in which frequent product changes limit data collection, and AFCC is designed for limited training data. Both the image acquisition and the labeling of C_N, C_{ED}, and C_{IN} weld formation followed the manufacturer's existing quality-control procedure rather than a protocol designed specifically for this

study, so the resulting class distribution and visual variability reflect the facility's actual production conditions rather than a curated or artificially balanced sample.

4.2 Experimental Settings

The dataset comprises 1,586 labeled weld surface images: 1,188 C_N, 218 C_{ED}, and 180 C_{IN} samples, reflecting the class imbalance encountered in the production data. A stratified 30% of the images (477) was reserved for testing, and the remaining 1,109 images formed the development set (Table 1). Each image corresponds to a distinct weld joint, so no weld appears in more than one partition. Stratified five-fold cross-validation was performed within the development set: in each fold, approximately 887 images were used for training and approximately 222 for validation, so every development image contributed to validation once. Within each fold, the backbone, encoders, and stream-specific SNNs were trained on the training portion only, and the chance-corrected scores were computed on the validation portion. After the fusion weights were fixed, the final model was retrained on all 1,109 development images, using the numbers of epochs determined by early stopping during cross-validation. The held-out test set was excluded from training, fusion-weight estimation, and model selection and was used once for final evaluation. Results are therefore reported for this predefined hold-out test set; repeated-partition variability is not estimated in the present study.

Table 1 Dataset composition and partitioning

Set	Number of Images	Normal Weld (C_N)	Excessive/Distorted Weld (C_{ED})	Insufficient Weld (C_{IN})	Ratio (%)
Development	1,109	831	152	126	70%
Test	477	357	66	54	30%
Total	1,586	1,188	218	180	100%

The convolutional branch receives RGB images at a resolution of 224 × 224 pixels and uses EfficientNet-B0 [23] as its backbone, initialized with ImageNet-pretrained weights and then retrained on the development set with all layers updated; the retrained backbone is then frozen while the grouped features are extracted and the SNNs are trained. The contour-guided branch operates on the grayscale image (Section 3.1). The six stream-specific SNNs and the global SNN [25] use the same configuration: two weight-sharing fully connected branches (128 and 64 units), L2-normalized embeddings, and a contrastive loss with a margin of 1.0. Training uses Adam (learning rate 10^{-3}), batches of 32 pairs, equal proportions of positive and negative pairs, and a maximum of 100 epochs with early stopping on validation loss (patience 10). The class reference samples used by the global SNN are drawn from the development set only, so no test image is used as a reference. The chance-corrected scores ρ_s used in Equation (10) are computed on the validation folds only, and all results reported in Sections 4.5 and 4.6 are obtained on the held-out test set. All experiments were run on a workstation with an Intel Core i5-14400F CPU, 32 GB of RAM, and an NVIDIA GeForce RTX 5060 Ti GPU with 16 GB of memory.

4.3 Evaluation Metrics

For each class C_k, $k \in \{N, ED, IN\}$, four outcomes are defined in a one-vs-rest manner: TP_k counts samples of C_k predicted as C_k; FN_k counts samples of C_k predicted as another class; FP_k counts samples of other classes predicted as C_k; and TN_k counts the remaining samples. Accuracy is the proportion of all test samples that are correctly classified, as defined in Equation (17). $Recall_k$ and $F1_k$ are computed per class [34] as defined in Equations (18) and (19), where $Precision_k = TP_k/(TP_k + FP_k)$. The reported Recall and F1 score are macro-averages over the three classes, so that each weld-formation category contributes equally regardless of its sample frequency.

$$Accuracy = \frac{\sum_k TP_k}{\sum_k (TP_k + FN_k)} \tag{17}$$

$$Recall_k = \frac{TP_k}{(TP_k + FN_k)} \tag{18}$$

$$F1_k = \frac{2}{\left(\frac{1}{Precision_k} + \frac{1}{Recall_k}\right)} \tag{19}$$

4.4 Visualization of Extracted Features

4.4.1 Contour-Guided Local Feature Extraction

Fig. 7 visualizes the three stages of the contour-guided pipeline (Section 3.2) for a representative sample of each class.

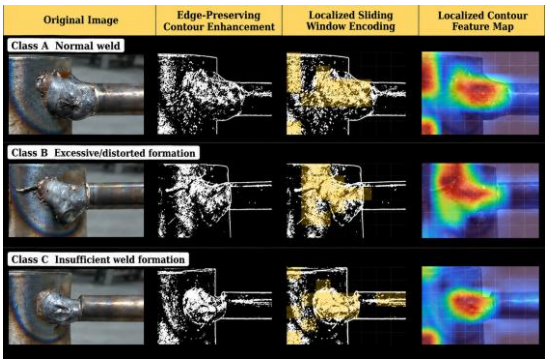


Fig. 7 Contour-guided feature extraction (A = C_N, B = C_{ED}, C = C_{IN})

After contour enhancement, the C_N sample shows a continuous boundary, the C_{ED} sample a larger, fragmented edge map with protrusions beyond the joint, and the C_{IN} sample a smaller, compact contour. Sliding-window encoding expresses these differences through the distribution of local shape responses. The resulting feature map (last column) visualizes their spatial distribution. The contour-guided branch subsequently

pools the local descriptors into the geometric stream defined in Section 3.4, so the visualization should not be interpreted as evidence that absolute spatial coordinates are retained in the final fused vector.

4.4.2 Feature Extraction via Convolutional Embeddings

Fig. 8 shows the five groups of intermediate embeddings produced by the CNN backbone (Section 3.3) for representative samples of each class.

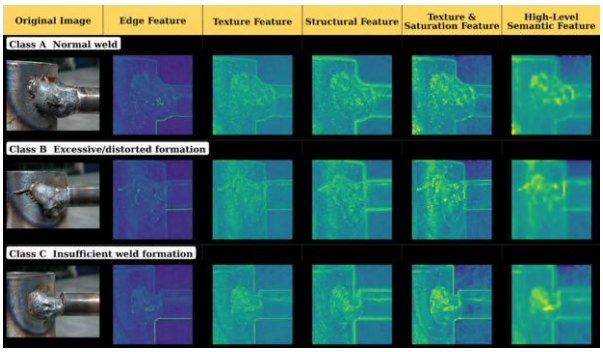


Fig. 8 Feature maps of the five semantic groups (A = C_N, B = C_{ED}, C = C_{IN})

The edge, texture, and structural groups reproduce the contour differences observed in Fig. 7: C_{ED} welds show enlarged, fragmented boundaries with protrusions, whereas C_{IN} welds show thinner, foreshortened boundaries. The texture-and-saturation group responds over a larger area for C_{ED}, consistent with excess deposition, and over a more confined area for C_{IN}, while the high-level semantic group yields stable patterns for C_N and broader or weaker responses for C_{ED} and C_{IN}, respectively. These group-wise differences, combined with the contour descriptors in F_{fused} (Section 3.4), form the basis for classification.

4.5 Classification Performance and Comparative Analysis

4.5.1 Ablation Study of the Adaptive Fusion Mechanism

Table 2 reports, for each stream, the balanced accuracy of its stream-specific SNN averaged over the five validation folds, the corresponding chance-corrected score, and the resulting fusion weight from Equation (10). Five streams receive weights between 0.166 and 0.210, whereas the high-level semantic group, whose balanced accuracy is only 12.2 percentage points above the chance level of 1/3, receives a weight of 0.049. The geometric, texture, structural, and texture-and-saturation streams exceed 1/6 and supply the feature maps used by the explanation module in Section 3.5.

Table 2 Fusion weights of the six streams

Stream	Validation BA	Chance-corrected score	Fusion weight
Geometric	82.00%	0.4867	0.1943
Edge	75.00%	0.4167	0.1663

Texture	82.00%	0.4867	0.1943
Structural	86.00%	0.5267	0.2102
Texture-and-saturation	80.00%	0.4667	0.1863
High-level semantic	45.50%	0.1217	0.0486
Sum	–	**2.5050**	**1.0000**

To quantify the contribution of each component of AFCC, three reduced configurations were evaluated. The contour-guided branch only uses the geometric stream; the convolutional branch only uses the five EfficientNet-B0 semantic groups of Section 3.3, weighted by Equation (10) over these five streams; and the equal-weight configuration keeps all six streams but assigns $w = 1/6$ to each instead of the chance-corrected weights of Equation (10). All configurations use the same SNN classifier and test set, so that differences reflect the feature representation and fusion strategy alone (Table 3).

Table 3 Ablation study on the test set

Configuration	**Accuracy**	**Recall**	**F1 Score**
Contour-guided branch only (geometric stream)	85.32%	80.74%	76.61%
Convolutional branch only (five semantic groups)	87.42%	81.22%	79.00%
AFCC with equal weights ($w = 1/6$)	92.87%	88.93%	87.96%
AFCC (proposed)	**95.39%**	**94.32%**	**92.31%**

Neither feature branch is sufficient on its own. The contour-guided branch yields the lowest scores (85.32% accuracy, 76.61% F1 score), indicating that the geometric stream alone cannot reliably separate the three formation classes, whereas the convolutional branch performs better (87.42% accuracy, 79.00% F1 score) but remains well below the complete framework (95.39% accuracy, 92.31% F1 score).

Combining all six streams with equal weights (w = 1/6) raises the F1 score to 87.96%, confirming that the two representations are complementary. The two weighting schemes differ most in the weight of the high-level semantic group (0.167 versus 0.049; Table 2). Replacing the equal weights with the chance-corrected weights of Equation (10) further raises the macro recall from 88.93% to 94.32% and the F1 score to 92.31%. The gain is concentrated in the defect classes, where recall rises from 86.36% to 92.42% for C_{ED} and from 85.19% to 94.44% for C_{IN}.

4.5.2 Comparison with Baseline Methods

AFCC was further compared with two baselines using the same development and test sets. The first is EfficientNet-B0 [23] fine-tuned end-to-end from ImageNet-pretrained weights with its own softmax classification head, which represents standard transfer learning for weld-image classification [7]; it differs from the convolutional-branch-only configuration in Table 3, which feeds the five semantic groups to the SNN. The second is Intelligent Fusion of Deep Features (IFDF) [27], a dual-fusion method originally proposed for multi-class waste classification that combines multiple deep models through feature-level and score-level fusion, re-implemented with the default

parameters of the original work, initialized with ImageNet-pretrained weights, and re-trained on the same development set. Fig. 9 shows the confusion matrices of the three methods, and Table 4 summarizes the overall metrics.

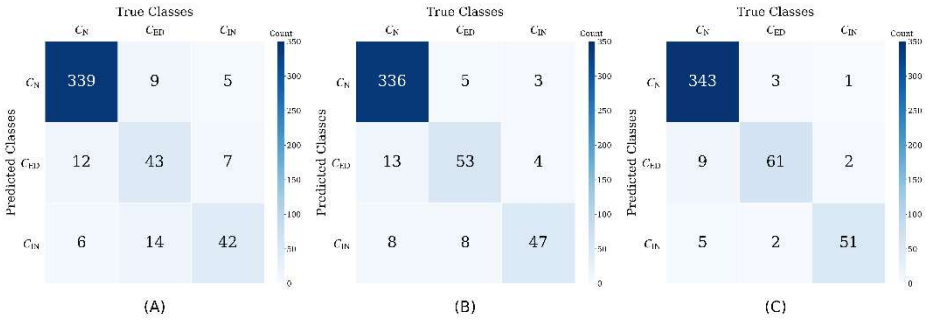


Fig. 9 Confusion matrices on the test set: (A) EfficientNet-B0, (B) IFDF, (C) AFCC

As shown in Fig. 9, AFCC misclassifies 22 of the 477 test images, compared with 41 for IFDF and 53 for the fine-tuned EfficientNet-B0. The three methods recognize C_N welds comparably (recall of 96.08% for AFCC, 94.96% for EfficientNet-B0, and 94.12% for IFDF), so the differences arise mainly in the defect classes: AFCC reaches a recall of 92.42% for C_{ED} and 94.44% for C_{IN}, compared with 65.15% and 77.78% for EfficientNet-B0 and 80.30% and 87.04% for IFDF. AFCC also confuses the two defect types less often (4 exchanges between C_{ED} and C_{IN}, versus 21 for EfficientNet-B0 and 12 for IFDF) and passes fewer defective welds as C_N (4, versus 14 and 8, respectively). Most of its remaining errors are C_N welds predicted as defective (14 samples), an error that leads to unnecessary remediation rather than allowing a defective weld to pass.

Table 4 Baseline comparison on the test set

Method	Accuracy	Recall	F1 Score
EfficientNet-B0 (baseline) [23]	88.89%	79.30%	78.36%
IFDF [27]	91.40%	87.15%	84.72%
AFCC (proposed)	**95.39%**	**94.32%**	**92.31%**

As listed in Table 4, AFCC achieves the highest value on all three metrics. Relative to the fine-tuned EfficientNet-B0, it raises accuracy, recall, and F1 score by 6.50, 15.02, and 13.95 percentage points, and relative to IFDF by 3.99, 7.17, and 7.59 percentage points, respectively. The fine-tuned network has the lowest scores of the three methods because of its low defect-class recall, whereas its recall for the majority C_N class remains comparable; overall accuracy therefore understates differences on the minority defect classes, which the macro-averaged recall and F1 score capture. The margin over both baselines suggests that adding the geometric stream and weighting each stream by its validated reliability improves the separation of the defect classes beyond what end-to-end fine-tuning or deep-feature fusion alone achieves.

4.5.3 Resource Analysis of the Decision-Gated Explanation Module

The resource demand of the explanation module in Section 3.5 was analyzed for the 477 test images. It compares an all-image policy with a decision-gated policy that submits only the 130 images classified as C_{ED} or C_{IN}. Each all-image request contained the original weld image, and each gated request contained the original image, a composite of the feature maps of the selected streams (Section 3.5, downscaled), and the predicted label as text; the payload of each request was measured from the submitted files (0.78 and 0.92 MB per request on average, respectively), and the response length was limited to 200 words. The network was measured on the client used for the experiment (Wi-Fi over a fibre connection; 513.3 Mbps downlink, 579.3 Mbps uplink, 14.03 ms latency), and the upload time of each request was computed as $u = 8p/b$ from the measured payload p (MB) and uplink rate b (Mbps). AFCC recognition time was measured on the workstation in Section 4.2 (0.85 s per image). Request elapsed time was measured with Claude Opus 5.5 from request submission to receipt of the complete response for all 477 independent all-image requests and 130 independent decision-gated requests. This client-side measurement includes upload, network communication, service-side processing, and response delivery; these components were not separately measured. The reported mean and standard deviation were calculated across the requests under each policy. Estimated upload time is reported separately for reference and is not added to the measured request elapsed time. Sequential batch time was estimated from the mean request elapsed time and, for the gated policy, includes AFCC recognition time for all 477 images; the complete batch was not timed as a single execution. The images were submitted to the cloud LLM service with the permission of the manufacturer.

Table 5 LLM workload and batch time

Item	All-image policy	Decision-gated policy
LLM requests	477	130
Response-length limit (words)	≤200	≤200
Measured payload per request (MB)	0.78	0.92
Total uploaded payload (MB)	371.5	120.1
AFCC recognition (min)	–	6.76
Estimated upload time per request (s)	0.011	0.013
Measured request elapsed time, Claude Opus 5.5 (s), mean ± SD	20.16 ± 1.30	22.95 ± 3.54
Estimated sequential batch time (min)	160.27	56.48

Payloads were measured from the submitted image files; they are not billed image-token counts or complete wire-level request sizes. Upload time was estimated from the measured uplink rate for reference only. Request elapsed time was measured across all 477 all-image and 130 decision-gated requests and already includes upload and network communication; estimated upload time was not added again. The sequential batch time is extrapolated from the mean request elapsed time and includes AFCC recognition for all 477 images under the gated policy; it is not a timed execution of the full batch.

As listed in Table 5, decision gating reduces the number of LLM requests from 477 to 130 (72.7%) while still covering 116 of the 120 defective welds in the test set (96.7%); the remaining 14 requests concern C_N welds predicted as defective. Although each gated request carries a slightly larger payload (0.92 versus 0.78 MB), the lower request count reduces the total uploaded payload from 371.5 to 120.1 MB (67.7%). At the measured uplink rate, upload time is negligible (0.011 and 0.013 s per request), so the batch time is governed primarily by the measured request elapsed time and the number of requests. Although gated requests carried additional evidence and had a longer mean elapsed time (22.95 versus 20.16 s), the estimated sequential batch time decreases from 160.27 to 56.48 min (64.8%), including the 6.76 min of AFCC recognition. In terms of Equations (15) and (16), the average payload per inspected image decreases from 0.78 to 0.25 MB and the average processing time from 20.2 to 7.1 s.

For continuous inspection with cloud LLM services, which are typically subject to request quotas or usage-based billing, decision gating lowers the request frequency and uploaded image-file volume. Any monetary saving depends on the provider's actual image-token and output-token billing and is not quantified here. In addition, each gated request provides the LLM with task-specific context, namely the predicted class and the feature maps of the most reliable streams, rather than the image alone. The payloads, network conditions, and request elapsed times were measured, whereas the correctness and usefulness of the generated explanations were not evaluated; quantifying these aspects with a specified model, prompt, and network environment is left for future work.

4.6 Error Analysis of Classification Results

The largest group of AFCC errors in Fig. 9(C) consists of C_N welds predicted as C_{ED} (9 of 22 errors). Fig. 10 shows a representative case together with the outputs of the contour-guided branch and the first two semantic groups.

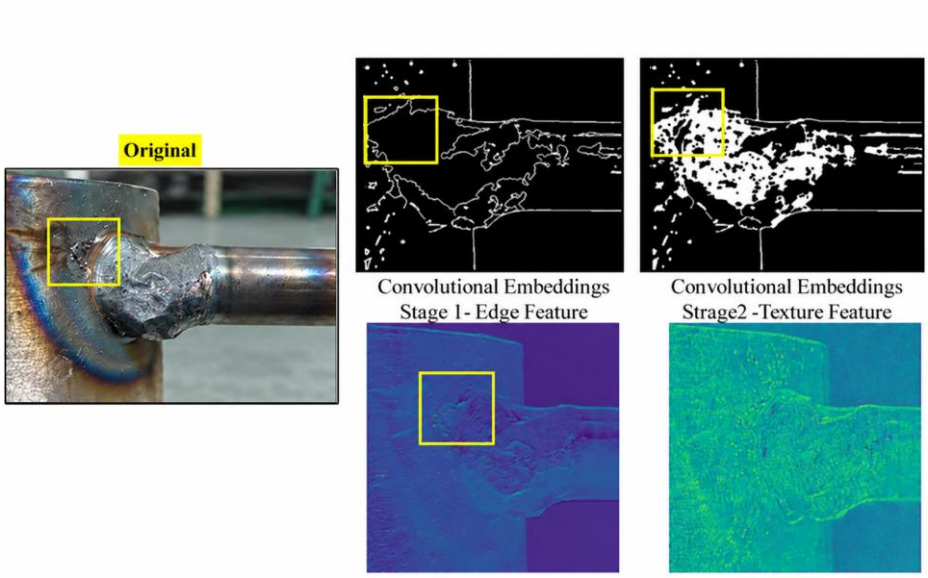


Fig. 10 C_N weld misclassified as C_{ED}; yellow boxes mark spatter

The weld in Fig. 10 is correctly formed, but spatter residue lies near the upper-left edge of the bead (yellow boxes). Its irregular texture and high local contrast produce edge-like responses, so contour enhancement merges the spatter with the weld boundary and the sliding-window encoding records an enlarged, irregular contour, which is the geometric signature that defines C_{ED}. The edge-group map responds strongly to the same region, and although the texture-group map partly suppresses it, the fused representation still indicates excess, irregular deposition, which leads to the C_{ED} prediction.

This example suggests that spatter near the weld boundary can contribute to false C_{ED} predictions because both feature branches respond to the affected region. Establishing its prevalence among all errors requires a systematic review of the misclassified samples. Spatter-aware masking or region-based filtering will be examined in future work.

5 Conclusions

This paper presented AFCC, a framework that fuses local contour descriptors with hierarchical convolutional features to classify weld formation on curved workpieces as normal, excessive/distorted, or insufficient. Each stream is weighted by its chance-corrected balanced accuracy on the validation folds. On 477 held-out test images acquired at-line, AFCC with an SNN classifier achieved 95.39% accuracy, 94.32% macro recall, and a 92.31% macro F1 score. Its macro F1 score exceeded those of a fine-tuned EfficientNet-B0 baseline and the IFDF fusion method by 13.95 and 7.59 percentage points, respectively. The ablation study further showed that neither branch alone nor equal

weighting reproduced the performance of the complete framework, supporting the use of complementary representations with validation-guided stream weighting.

These results were obtained from a development set of 1,109 images and a held-out test set of 477 images. Recall exceeded 92% for both defect classes, indicating that the three-class output can provide decision-relevant information for production workflows in which excessive material may be routed toward grinding and insufficient formation toward supplementary welding. The present experiments validate weld-formation classification only; they do not quantify robotic repair effectiveness or closed-loop remediation performance.

The illustrated error case suggests that spatter near the weld boundary can cause a normal weld to be classified as C_{ED}. The study is also limited to one production line, one camera configuration, externally observable formation conditions, and a single predefined hold-out test set; all images come from curved rack-corner joints, and no comparison with planar welds was performed. Future work will evaluate production-batch-grouped partitions, repeated partitions, additional weld geometries, and spatter-aware preprocessing. For the decision-gated explanation module, the 477-image batch estimate based on measured request times indicated that restricting LLM requests to the 130 flagged samples would reduce request volume by 72.7% and the estimated processing time by about 65%; its explanation quality will be evaluated with a specified model, prompt, and network environment.

References

1. Cai H, Wang R, Feng X, Cao H, Wang T, Yu H, Su M (2026) Review of welding technologies for electronic precision equipment: status, limitations, and prospects. Materials 19(18):3976. https://doi.org/10.3390/ma19183976
2. Thomas DJ (2018) Analyzing the failure of welded steel components in construction systems. J Fail Anal Prev 18(2):304–314. https://doi.org/10.1007/s11668-018-0392-x
3. Manokruang S, Vignat F, Museau M, Limousin M (2021) Process parameters effect on weld beads geometry deposited by wire and arc additive manufacturing (WAAM). In: Advances on Mechanics, Design Engineering and Manufacturing III. Lecture Notes in Mechanical Engineering. Springer, Cham, pp 9–14. https://doi.org/10.1007/978-3-030-70566-4_3
4. Yu R, Kershaw J, Wang P, Zhang Y (2022) How to accurately monitor the weld penetration from dynamic weld pool serial images using CNN-LSTM deep learning model? IEEE Robot Autom Lett 7(3):6519–6525. https://doi.org/10.1109/LRA.2022.3173659
5. Yang L, Zhao J, Wang Z, Liu Y, Chi D (2025) M-KANUNet: enhanced defect segmentation in X-ray images of copper pipe welds via multi-scale representation and Kolmogorov–Arnold Networks. Vis Comput 41:7201–7214. https://doi.org/10.1007/s00371-025-03800-4
6. Hu J, Cao Y, Huang J, Yu T, Yao J, Wang P, He Q (2024) Adaptive signal reconstruction based on VMD for rail welding joint defect detection. Russ J Nondestruct Test 60(11):1249–1262. https://doi.org/10.1134/S1061830924602502
7. Kumaresan S, Aultrin KSJ, Kumar SS, Dev Anand M (2023) Deep learning-based weld defect classification using VGG16 transfer learning adaptive fine-tuning. Int J Interact Des Manuf 17(6):2999–3010. https://doi.org/10.1007/s12008-023-01327-3

8. Yang M, Li S, Sun X, Wang H, Gao A, Du F, Chen C (2026) StripePoint-YOLO: task-adaptive detection of multi-type weld seam keypoints in noisy industrial welding scenes. Sensors 26(15):4980. https://doi.org/10.3390/s26154980
9. Canny J (1986) A computational approach to edge detection. IEEE Trans Pattern Anal Mach Intell PAMI-8(6):679–698. https://doi.org/10.1109/TPAMI.1986.4767851
10. Hu M-K (1962) Visual pattern recognition by moment invariants. IRE Trans Inf Theory 8(2):179–187. https://doi.org/10.1109/TIT.1962.1057692
11. Nacereddine N, Ziou D, Hamami L (2013) Fusion-based shape descriptor for weld defect radiographic image retrieval. Int J Adv Manuf Technol 68(9):2815–2832. https://doi.org/10.1007/s00170-013-4857-5
12. Zeiler MD, Fergus R (2014) Visualizing and understanding convolutional networks. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds) Computer Vision – ECCV 2014. Lecture Notes in Computer Science, vol 8689. Springer, Cham, pp 818–833. https://doi.org/10.1007/978-3-319-10590-1_53
13. Hou L (2023) Regularizing label-augmented generative adversarial networks under limited data. IEEE Access 11:28966–28976. https://doi.org/10.1109/ACCESS.2023.3259066
14. Wang Z, Li Y, Zhang R, Wang J, Cui H (2025) More diversity, less redundancy: feature refinement network for few-shot SAR image classification. Comput Electr Eng 123:110043. https://doi.org/10.1016/j.compeleceng.2024.110043
15. Cao R, Zhang J, Shen Y, Zhou H, Zhou P, Shen G, Xia Z, Zang Y, Liu Q, Hu W (2025) Dual-flow feature enhancement network for robust anomaly detection in stainless steel pipe welding. Vis Comput 41(8):5873–5889. https://doi.org/10.1007/s00371-024-03757-w
16. Hong Z et al (2024) EEKHI–SAR–SIFT: edge enhancement and keypoint homogeneous improved SAR–SIFT framework based on unbiased difference ratio edge detector strategy. IEEE J Sel Top Appl Earth Obs Remote Sens 17:13837–13852. https://doi.org/10.1109/JSTARS.2024.3438796
17. Liu S, Dong L, Liao X, Hao Y, Cao X, Wang X (2019) A dilation and erosion-based clustering approach for fault diagnosis of photovoltaic arrays. IEEE Sens J 19(11):4123–4137. https://doi.org/10.1109/JSEN.2019.2896236
18. Gonzalez RC, Woods RE (2008) Digital image processing, 3rd edn. Prentice Hall, Upper Saddle River, NJ
19. Neubeck A, Van Gool L (2006) Efficient non-maximum suppression. In: 18th International Conference on Pattern Recognition (ICPR), vol 3. IEEE, pp 850–855. https://doi.org/10.1109/ICPR.2006.479
20. Xie G, Guo B, Huang Z, Zheng Y, Yan Y (2020) Combination of dominant color descriptor and Hu moments in consistent zone for content based image retrieval. IEEE Access 8:146284–146299. https://doi.org/10.1109/ACCESS.2020.3015285
21. He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, pp 770–778. https://doi.org/10.1109/CVPR.2016.90
22. Szegedy C, Ioffe S, Vanhoucke V, Alemi A (2017) Inception-v4, Inception-ResNet and the impact of residual connections on learning. In: Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence (AAAI-17), pp 4278–4284. https://doi.org/10.1609/aaai.v31i1.11231
23. Tan M, Le QV (2019) EfficientNet: rethinking model scaling for convolutional neural networks. In: Proceedings of the 36th International Conference on Machine Learning (ICML), PMLR 97, pp 6105–6114

24. He C, Cao Y, Yang Y, Liu Y, Liu X, Cao Z (2023) Fault diagnosis of rotating machinery based on the improved multidimensional normalization ResNet. IEEE Trans Instrum Meas 72:3524311. https://doi.org/10.1109/TIM.2023.3293554
25. Bromley J, Bentz JW, Bottou L, Guyon I, LeCun Y, Moore C, Säckinger E, Shah R (1993) Signature verification using a "Siamese" time delay neural network. Int J Pattern Recognit Artif Intell 7(4):669–688. https://doi.org/10.1142/S0218001493000339
26. Yang J, Kim S (2024) Noise robust SAR image classification using Siamese spiking neural networks. IEEE Geosci Remote Sens Lett 21:4017205. https://doi.org/10.1109/LGRS.2024.3465532
27. Ahmad K, Khan K, Al-Fuqaha A (2020) Intelligent fusion of deep features for improved waste classification. IEEE Access 8:96495–96504. https://doi.org/10.1109/ACCESS.2020.2995681
28. Nguyen HV, Park H, Yoo N, Yang J (2026) Resource-efficient adaptation of large vision–language models for multimodal defect inspection in LNG tank manufacturing. Int J Adv Manuf Technol 144(7–8):5135–5152. https://doi.org/10.1007/s00170-026-17818-y
29. Jin Y, Li J, Gu T, Liu Y, Zhao B, Lai J, Gan Z, Wang Y, Wang C, Tan X, Ma L (2025) Efficient multimodal large language models: a survey. Vis Intell 3:27. https://doi.org/10.1007/s44267-025-00099-6
30. Yin S, Fu C, Zhao S, Xu T, Wang H, Sui D, Shen Y, Li K, Sun X, Chen E (2024) Woodpecker: hallucination correction for multimodal large language models. Sci China Inf Sci 67(12):220105. https://doi.org/10.1007/s11432-024-4251-x
31. Geigle G, Timofte R, Glavaš G (2024) Does object grounding really reduce hallucination of large vision-language models? In: Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing (EMNLP), Miami, FL, USA, pp 2728–2742. https://doi.org/10.18653/v1/2024.emnlp-main.159
32. Brodersen KH, Ong CS, Stephan KE, Buhmann JM (2010) The balanced accuracy and its posterior distribution. In: 20th International Conference on Pattern Recognition (ICPR), Istanbul, Turkey. IEEE, pp 3121–3124. https://doi.org/10.1109/ICPR.2010.764
33. Cohen J (1960) A coefficient of agreement for nominal scales. Educ Psychol Meas 20(1):37–46. https://doi.org/10.1177/001316446002000104
34. Powers DMW (2011) Evaluation: from precision, recall and F-measure to ROC, informedness, markedness & correlation. Int J Mach Learn Technol 2(1):37–63

Statements and Declarations

Funding This research was funded by the National Science and Technology Council (NSTC) of the Republic of China under Grant No. NSTC 114-2222-E-167-004 and NSTC 115-2221-E-167-029.

Competing Interests The authors declare no competing interests.

Ethics Approval Not applicable. This study did not involve human participants or animals.

Consent to Participate Not applicable.

Consent for Publication Not applicable.

Author Contributions Cheng-Hui Chen conceived the study, developed the methodology and software, designed and conducted the experiments, analyzed the results, and wrote and revised the manuscript. Tsung-Hua Tsai assisted with the experiments. Both authors read and approved the final manuscript.

Data Availability Due to confidentiality agreements and privacy concerns, the data used in this study cannot be publicly shared. However, interested researchers may request access under appropriate conditions by contacting the corresponding author.

Code Availability The code is available from the corresponding author upon reasonable request.