Attention-Based Residual Inception Approach for Fruit Classification and Grading

First A. Author, Second B. Author, and Third C. Author

***Abstract*— Automated fruit inspection must recognise both the fruit type and its quality grade, yet most image-based systems treat these tasks separately and rarely explain their decisions. This study aimed to build a single explainable network that predicts type and grade jointly from ordinary colour photographs. The proposed Mean Squared LogCosh Residual Inception Network (MLRI-Net) enhances each image with a contextual Wiener filter, isolates the fruit with a graph-attention segmentation network trained on automatically generated masks, regularises training with CutMix and MixUp, and fuses texture descriptors and deep features with a residual-attention-inception image branch. Two classification heads are trained with a combined root mean squared logarithmic error and log-cosh loss, and layer-ordered saliency and layer-wise relevance maps explain the grading decisions. The method was evaluated on the public FruitNet and FruitQ datasets against four baselines over five training-data fractions and five-fold cross-validation. On FruitQ, MLRI-Net achieved the highest mean grading accuracy of 99.85% and significantly outperformed every baseline in paired t-tests. On FruitNet, it reached 97.20% grading accuracy, whereas a fused deep-feature support vector machine reached 99.12%, and removing the segmentation stage improved MLRI-Net. Removing CutMix and MixUp reduced FruitQ grading accuracy at every training fraction. The results show that joint, explainable type and grade prediction is effective, while the benefit of background isolation depends on the dataset.**

***Index Terms*— Explainable artificial intelligence, fruit quality grading, fruit type classification, image segmentation, residual inception network**

I. Introduction

FRUIT quality inspection determines the price, shelf life and export eligibility of fresh produce, and it remains largely manual in many supply chains. Manual sorting relies on visual judgement of colour, texture and surface defects, which makes it slow, subjective and difficult to scale to the volumes handled by packing houses [1], [2]. Computer vision offers an objective alternative because a single camera image already contains the cues that human graders use, and convolutional neural networks have become the dominant tool for extracting those cues from agricultural images [3]–[6]. Public image collections such as FruitNet and FruitQ now make it possible to study fruit type and quality recognition under realistic acquisition conditions [7], [8].

Manuscript received XX; revised XX. (Funding information to be supplied.)

Author names, affiliations and corresponding-author e-mail to be supplied.

The problem is significant because fruit type and quality grade are coupled decisions in practice: a grading line must first know which fruit it sees and then decide whether that fruit is good, partly damaged or spoiled. Visual quality cues differ strongly across fruit types, appear as small local texture changes, and are easily confused with shadows, hands and cluttered backgrounds [1]. Image noise and background clutter therefore degrade quality assessment, which is why recent agricultural pipelines add image enhancement and region isolation before classification [6], [9]. Non-imaging approaches such as hyperspectral firmness assessment provide valuable quality information, but they require specialised acquisition hardware that conventional sorting lines rarely provide [10].

Existing image-based techniques address parts of this problem. Same-domain transfer learning with EfficientNetV2 grades six FruitNet fruits through two sequential models [1], [11], fused pre-trained deep features with principal component analysis detect fresh and rotten produce [2], and residual networks with channel attention and inception modules classify plant diseases accurately [5], [12]. Handcrafted descriptors such as the centre-symmetric local multilevel pattern capture fine texture [13], while CutMix and MixUp regularise deep classifiers [14], [15]. Regression losses such as root mean squared logarithmic error and log-cosh loss have been surveyed for forecasting rather than classification [16], and explanation methods such as layer-ordered saliency and layer-wise relevance propagation have mainly been demonstrated on medical images [17]–[19]. The research problem is therefore a gap in integration: no single framework in the reviewed literature combines enhancement, background isolation, hybrid texture and deep features, joint type and grade prediction and decision explanations, and evaluates that combination on more than one public fruit-quality dataset.

This paper proposes the Mean Squared LogCosh Residual Inception Network (MLRI-Net), an explainable framework that predicts fruit type and quality grade from one colour image. The framework places a learned background isolation step and a hybrid feature representation in front of a residual-attention-inception classifier and trains both prediction heads with a combined logarithmic and log-cosh objective. The study evaluates the framework under controlled conditions against reproduced baselines from the literature, using model complexity analysis in the manner of lightweight medical networks [20].

The main contributions of this work are as follows. First, it formulates fruit type classification and quality grading as

a joint two-head problem solved by one network with a shared fused representation. Second, it trains a graph-attention segmentation network from automatically generated pseudo-masks, which removes the need for manually annotated fruit masks. Third, it fuses centre-symmetric local multilevel pattern texture features and ResNeXt-101 deep features with a residual-attention-inception image branch. Fourth, it adapts a combined root mean squared logarithmic error and log-cosh loss to probabilistic classification outputs. Fifth, it provides layer-ordered saliency and relevance-propagation explanations together with an evaluation protocol that includes data-fraction sweeps, ablation, cross-validation, paired statistical testing and complexity analysis on two public datasets.

The rest of this paper is organised as follows. Section II presents the motivation, the literature survey and the research gaps. Section III describes the proposed method and its components. Section IV reports the experimental procedure, the datasets, the parameter settings, the metrics and the results. Section V discusses the findings, compares them with published studies and states the limitations. Section VI concludes the paper and outlines future work.

II. Motivation

This research was motivated by the practical gap between accurate single-task fruit classifiers and the needs of an inspection line, which must name the fruit, grade it, cope with uncontrolled backgrounds and justify its decisions to operators. The literature survey below reviews ten studies that supplied the techniques combined in this work, and the research-gap list maps each unresolved issue to a component of the proposed method and to the evidence that tests it.

A. Literature Survey

Aldakhil and Almutairi [1] trained an EfficientNetV2 model to classify six FruitNet fruit types and then transferred its weights to a second model that grades quality; with AugMix, CutMix and MixUp augmentation they reported test accuracies of 99.49% for type and 99.42% for grade. The two tasks are learned by separate sequential models, the evaluation uses FruitNet only, and the study reports no visual explanation of the decisions, which the joint, explainable and two-dataset design of the present work targets. Yuan and Chen [2] fused GoogLeNet, DenseNet-201 and ResNeXt-101 features, reduced them with principal component analysis and detected freshness with classical classifiers, reaching 96.98% accuracy on twenty fresh and rotten categories. Their labels distinguish only fresh and rotten produce, so intermediate quality levels and handcrafted texture cues are not modelled, which the present work addresses with three-level grading and centre-symmetric texture features. Castillo-Gironés *et al.* [10] combined hyperspectral imaging, a variational autoencoder and a support vector machine to classify persimmon firmness into three categories with 81.3% accuracy. The approach depends on hyperspectral acquisition and a single fruit cultivar, whereas the present work uses ordinary colour images of up to eleven fruit types.

Sudhakar *et al.* [6] introduced the Contextual Image Enhancement Wiener Filter (CIEWF), generative augmentation, region-of-interest segmentation and a parallel multi-scale attention network with Grad-CAM explanations for cassava and groundnut leaf disease diagnosis. The pipeline was designed and tested on leaf disease images, so its enhancement stage had not been evaluated for fruit-quality grading, which the present work does. Liu *et al.* [9] proposed DLNet, a double-layer graph attention segmentation network built on a refined feature pyramid detector, and reported 80.9% box and 81.2% mask average precision for occluded green fruit. DLNet is trained with instance-level mask annotations, which fruit-quality datasets do not provide; the present work therefore trains a DLNet-style head from automatically generated pseudo-masks. Maurya *et al.* [5] combined a fine-tuned ResNet18, squeeze-and-excitation channel attention and an inception module in RAI-Net and achieved 97.88% accuracy on ten tomato-leaf disease classes. RAI-Net solves a single classification task with cross-entropy, and the authors identify the lack of real-life testing data as its main limitation; the present work extends the architecture to joint type and grade prediction and tests it on two public fruit datasets. Al-wajih and Ghazali [13] showed that sliding-window local binary pattern variants, including the centre-symmetric local multilevel pattern (CS-LMP), improve handwritten Arabic digit recognition by up to 21.43%. Their descriptors are used alone with shallow classifiers, so they do not exploit complementary deep features, which the present fusion design provides.

Jadon *et al.* [16] surveyed fourteen regression losses, including root mean squared logarithmic error (RMSLE) and log-cosh loss, and compared them on time-series forecasting tasks. The survey evaluates these losses for forecasting only, so their behaviour as classification objectives remained untested; the present work applies their combination to softmax outputs and evaluates it, although its contribution is not isolated by a dedicated loss ablation. Avram *et al.* [17] developed CathAI for automated coronary angiography interpretation and used Layer Ordered Visualization of Information (LOVI) saliency to show which pixels drive stenosis estimates. LOVI was demonstrated on angiograms, and its use for agricultural produce had not been reported, which the present work examines. Iqbal *et al.* [18] designed a lightweight convolutional network for coronary artery disease classification in magnetic resonance images, reached 99.35% accuracy and explained its predictions with Grad-CAM and layer-wise relevance propagation (LRP). Their explanations target clinical images, so relevance-based explanations of fruit-quality decisions remained unexplored, which the present work provides qualitatively and with a simple mask-overlap measure.

B. Research Gap

The survey identifies the following gaps; each is mapped to the component of MLRI-Net that targets it and to the evidence obtained in Section IV.

- *Joint prediction:* fruit type and grade are usually predicted by separate models [1]. MLRI-Net predicts both with two heads on one fused representation; Section IV shows

100% type accuracy and 99.85% mean grading accuracy on FruitQ, but lower type accuracy than all four baselines on FruitNet, so this gap is addressed only partly.

- *Annotation-free background isolation:* segmentation networks for fruit require manual mask annotations [9]. The DLNet-style head learns from pseudo-masks and reaches 0.875 and 0.944 intersection over union with them; however, the ablation shows that background isolation lowered FruitNet accuracy and had no measurable effect on FruitQ, so this gap remains open for grading performance.
- *Hybrid features:* handcrafted texture and deep features are seldom fused for grading [2], [13]. MLRI-Net fuses both; the experiments do not include a feature-level ablation, so the benefit of fusion is not isolated.
- *Robustness to limited data:* class imbalance and limited data weaken grading models [1]. CutMix and MixUp regularise MLRI-Net; removing them reduced FruitQ grading accuracy by up to 6.6 percentage points, which supports this component.
- *Loss design:* regression losses have not been evaluated as classification objectives [16]. MLRI-Net uses them; their effect is confounded with other differences from the cross-entropy RAI-Net baseline and therefore remains unresolved.
- *Explainability:* explanations of fruit-grading decisions are rarely provided [1], [2]. LOVI and LRP maps are produced for every grading decision; 57–78% of positive relevance fell on the fruit region in the examined samples, but no user study or faithfulness test was performed.
- *Expected value:* the practical value of an integrated, explainable grader lies in reducing manual inspection effort and in giving operators visual evidence for each decision; this value is expected rather than measured in the present study.

III. Proposed MLRI-Net Framework for Joint Fruit Type Classification and Quality Grading

Fig. 1 shows the block diagram of the proposed framework. A colour fruit image first passes through the Contextual Image Enhancement Wiener Filter (CIEWF), which suppresses sensor noise while preserving edges. A DLNet-style segmentation network then isolates the fruit from the background and produces a fruit-centred segmented image. During training, CutMix or MixUp mixes pairs of samples in every batch. Two complementary descriptors are then computed: a centre-symmetric local multilevel pattern (CS-LMP) texture vector and a ResNeXt-101 deep feature vector. The Mean Squared LogCosh Residual Inception Network (MLRI-Net) fuses these descriptors with a residual-attention-inception image branch and predicts the fruit type and the quality grade with two softmax heads trained by a combined RMSLE and log-cosh objective. Finally, Layer Ordered Visualization of Information (LOVI) and layer-wise relevance propagation (LRP) explain each grading decision. The following subsections describe these stages in execution order.

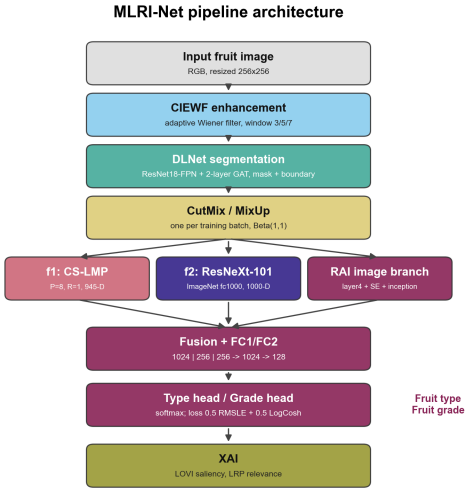


Fig. 1. Block diagram of the proposed MLRI-Net framework, from the input fruit image to the fruit type, the fruit grade and the explanation maps.

A. Fruit Image Acquisition and Stratified Subset Formation

Let the dataset be the set of n labelled images $D = \{I_1, I_2, \ldots, I_n\}$, where each image I_m carries a fruit-type label t_m from a set of T types and a quality-grade label g_m from a set of G grades. Both labels are read from the folder structure of the public datasets described in Section IV-B. Grade names are normalised so that equivalent labels share one name; for example, the FruitQ label Fresh is mapped to Good. The implementation first selects a reproducible subset: it keeps DATA_PERCENT $= 100\%$ of every type–grade class and then caps each class at 300 images with a fixed random seed. This cap bounds the cost of the many training runs of the evaluation protocol on a CPU-only computer while keeping every class represented. Every image is decoded as an 8-bit RGB array and resized to 256×256 pixels, which is the working resolution of the enhancement and segmentation stages.

The selected images, their labels and the class indices are written to a manifest that every later stage reads, so the enhancement, segmentation, feature extraction, training and explanation stages all refer to the same samples in the same order. Grades are ordered from best to worst quality (Good, Mild or Mixed, Bad or Rotten), which keeps the class index of each grade consistent across datasets. Because the cap is applied per type–grade class, the subset is more balanced than the original collections: in FruitNet, for example, the over-represented Good pomegranate class, which contains 5,940 images, contributes the same 300 images as the other full classes. The cap therefore changes the class distribution, and the experiments report results for the capped subsets rather than for the complete datasets.

B. Contextual Wiener Enhancement of Fruit Images

Fruit photographs taken with handheld cameras contain sensor noise and compression artefacts that can mimic small

surface blemishes. The CIEWF stage follows the contextual Wiener formulation of Sudhakar *et al.* [6], which is a locally adaptive minimum mean-square-error filter of the type introduced by Lee [21]. For a pixel (x_1, x_2) in channel c of image I_m, the enhanced value is

$$P_m(x_1,x_2) = \alpha + \frac{\max(\beta^2 - \beta_n^2, 0)}{\max(\beta^2, \beta_n^2)}\big(I_m(x_1,x_2) - \alpha\big) \quad (1)$$

where α is the local mean, β^2 is the local variance and β_n^2 is the noise variance. The local statistics are computed over a square window s of side w centred on the pixel:

$$\alpha = \frac{1}{|s|}\sum_{(u,v)\in s} I_m(u,v), \quad \beta^2 = \frac{1}{|s|}\sum_{(u,v)\in s} I_m^2(u,v) - \alpha^2 \quad (2)$$

where $|s| = w^2$ is the number of pixels in the window. When the noise variance is unknown, it is estimated as the mean of all local variances of the channel. In flat regions β^2 approaches β_n^2, the gain in (1) approaches zero and the pixel is pulled towards the local mean; on edges and lesions β^2 is large, the gain approaches one and the original detail is retained.

The original CIEWF description states that the filter adapts its window to the image content but does not specify the window size. The implemented variant therefore evaluates three windows, $w \in \{3, 5, 7\}$, for every image and keeps the window that leaves the lowest residual noise while retaining at least half of the edge energy. The residual noise is measured with the fast noise-variance estimator of Immerkær [22],

$$\hat{\sigma}_n = \sqrt{\frac{\pi}{2}}\,\frac{1}{6(W-2)(H-2)}\sum_{x_1,x_2}\big|\big(P_m^{gray} * N\big)(x_1,x_2)\big| \quad (3)$$

where P_m^{gray} is the grey-level enhanced image of width W and height H and N is the 3×3 Laplacian-difference mask with rows $(1, -2, 1)$, $(-2, 4, -2)$ and $(1, -2, 1)$. The edge-energy ratio is

$$E_{keep} = \frac{\mathrm{Var}\big(\nabla^2 P_m^{gray}\big)}{\mathrm{Var}\big(\nabla^2 I_m^{gray}\big)} \quad (4)$$

where ∇^2 denotes the Laplacian operator and Var the variance over all pixels. The adaptive rule makes the enhancement image-specific: strongly textured images receive a small window, and smooth, noisy images receive a large one. The selected window, the peak signal-to-noise ratio and the structural similarity with respect to the input are recorded for every image and reported in Section IV-E.

The peak signal-to-noise ratio (PSNR) between the input and the enhanced image measures how much the filter changed the image:

$$\mathrm{PSNR} = 10\log_{10}\left(\frac{255^2}{\frac{1}{3WH}\sum_{c,x_1,x_2}\big(I_m - P_m\big)^2}\right) \quad (5)$$

where the sum runs over the three colour channels and all pixels, and the result is expressed in decibels (dB). A high PSNR indicates that the filter removed little content, whereas a low PSNR indicates strong smoothing. A Wiener-type filter was chosen instead of a fixed Gaussian or median filter because its gain depends on the local signal-to-noise ratio, so it smooths uniform skin while leaving the sharp intensity changes of bruises and rot spots largely unchanged, which is the property that the subsequent texture descriptor depends on.

C. Pseudo-Mask-Supervised DLNet Background Isolation

Hands, trays, labels and cloth backgrounds share colours with some fruits and can bias a grading model. DLNet [9] isolates fruit with a double-layer graph attention head, but it is trained with manual instance masks that FruitNet and FruitQ do not provide. The proposed framework therefore trains a DLNet-style segmentation network from masks that are generated automatically. Because each image of the target datasets shows one fruit or a small group of fruits, the whole image acts as the region of interest and the detection stage of the original DLNet is not required.

1) Automatic Pseudo-Mask Generation: For a training image, two foreground cues are computed on the enhanced image resized to 128×128 pixels. GrabCut [23] is initialised with a rectangle that excludes a 6% border and runs four iterations; pixels labelled as certain or probable foreground form the mask M_{GC}. Otsu thresholding [24] of the HSV saturation channel forms a second mask M_S, because fruit skin is usually more saturated than the background. Both masks are cleaned by morphological opening and closing, components smaller than 2% of the image are removed and holes are filled. A mask is considered plausible when its foreground fraction lies between 3% and 97%. The pseudo-mask is

$$M^* = \begin{cases} M_{GC} \wedge M_S, & \mathrm{IoU}(M_{GC}, M_S) > 0.5 \\ M_{GC}, & \text{otherwise} \end{cases} \quad (6)$$

when both cues are plausible, where $\wedge$ is the pixel-wise logical AND and IoU is the intersection over union. If only one cue is plausible, that cue is used; if neither is plausible, a centred ellipse covering 35% of each image dimension is used so that training never stops. The boundary target B is the morphological gradient of M computed with a 3×3 elliptical structuring element. Pseudo-masks are generated for 1,000 randomly selected images of each dataset.

2) Double-Layer Graph Attention Segmentation Head: Fig. 2 shows the segmentation network. An ImageNet-pretrained ResNet18 encoder [3] produces features at one quarter, one eighth, one sixteenth and one thirty-second of the input resolution. A feature pyramid decoder merges them into a 96-channel map X_{roi} at one quarter resolution. A 3×3 convolution produces the head features h, and two graph attention layers model the image as a pixel graph, following the double-layer design of DLNet [9]. Each graph attention layer is implemented with the position-attention and channel-attention modules of the dual attention network [25]. For the position branch, every pixel is a node, and the adjacency matrix is computed from query and key projections:

$$A = \mathrm{softmax}\big((XW_q)(XW_k)^\top\big), \quad O = \gamma A(XW_g) + X \quad (7)$$

where $X \in \mathbb{R}^{N\times K}$ is the feature matrix of $N = H \times W$ pixel nodes with K channels, W_q and W_k are 1×1 projections to $K/8$ channels, W_g is the value projection, $A \in \mathbb{R}^{N\times N}$ is the learned adjacency and γ is a learnable scale initialised at

zero. The channel branch applies the same idea with channels as nodes, and the two outputs are summed and passed through a 3×3 convolution, batch normalisation and a ReLU, denoted $\sigma(\cdot)$. The first graph layer produces $g_0 = \mathrm{GAT}_0(h)$, from which the boundary is predicted. Its output is fused with the region features through a fully convolutional projection W_f:

$$O_f = g_0 W_f + X_{roi}, \quad O_1 = \mathrm{GAT}_1(O_f + g_0) \tag{8}$$

where O_f is the occlusion-aware fused feature and O_1 is the output of the second graph layer, from which the mask is predicted. Both predictions are produced by 1×1 convolutions and bilinear upsampling to the input size.

3) Segmentation Objective and Fruit-Centred Cropping: The segmentation network is trained with a mask term that combines binary cross-entropy and the Dice loss and with a weighted boundary term:

$$L_{seg} = \mathrm{BCE}(\hat{M}, M^*) + \left(1 - \frac{2\sum \hat{M}M^* + 1}{\sum \hat{M} + \sum M^* + 1}\right) + 0.5\ \mathrm{BCE}_{w=5}(\hat{B}, B^*) \tag{9}$$

where $\hat{M}$ and $\hat{B}$ are the predicted mask and boundary probabilities, the sums run over all pixels and BCE with $w = 5$ weights the rare boundary pixels five times more than background pixels. The encoder stem and first stage stay frozen, the rest is trained with Adam (learning rate 10^{-3}, weight decay 10^{-5}) for eight epochs with batch size 16 and random horizontal flips.

At inference, the predicted mask is thresholded at 0.5, components smaller than 15% of the largest component are removed, and an empty prediction falls back to the whole image. The segmented image S_m keeps the enhanced pixels inside the mask, sets the background to black, crops a square around the mask bounding box with an 8% margin and resizes the crop to 256×256 pixels. Cropping places the fruit at a consistent scale for the following feature extractors.

Two design choices make this stage practical. First, the graph attention layers operate on the decoder output at one quarter of the 128×128 input, so the position graph has $N = 32 \times 32 = 1{,}024$ nodes and its adjacency matrix stays small enough for CPU training. Second, pseudo-mask supervision trades annotation cost for label noise: the network learns the consensus of GrabCut and saturation thresholding and can generalise beyond individual pseudo-mask errors, but systematic errors of the cues, such as saturated background objects, can also be learned. The agreement between predicted masks and pseudo-masks is monitored during training and reported in Section IV-E.

D. Batch-Level CutMix and MixUp Regularisation

Quality classes are imbalanced, and some type–grade combinations contain few images. Following Aldakhil and Almutairi [1], one of CutMix [14] or MixUp [15] is chosen with equal probability for every training batch. For two samples (x_i, y_i) and (x_j, y_j) and a mixing ratio λ drawn from $\mathrm{Beta}(1,1)$, MixUp forms

$$x' = \lambda x_i + (1-\lambda)x_j, \quad y' = \lambda y_i + (1-\lambda)y_j \tag{10}$$

and CutMix forms

$$x' = M \odot x_i + (1-M) \odot x_j, \quad y' = \lambda' y_i + (1-\lambda')y_j \tag{11}$$

where M is a binary mask that removes a rectangle of relative area $1-\lambda$, $\odot$ is the element-wise product and λ' is the exact fraction of the area kept from x_i after the rectangle is clipped to the image. The type label and the grade label of a sample are mixed with the same ratio, so the two heads receive consistent soft targets.

The implementation applies the mixing to the cached feature representation rather than to raw pixels, because the frozen layers described in Section III-F.1 are computed once per image. CutMix cuts a rectangle from the 8×8 frozen ResNet18 feature maps, MixUp blends whole feature maps, and the corresponding texture and deep feature vectors are mixed with the same ratio. This manifold-level mixing reproduces the regularisation intent of the original methods at a fraction of the computational cost; image-level CutMix and MixUp examples are generated only for visual inspection in Section IV-E.

The $\mathrm{Beta}(1,1)$ distribution, which is uniform on $[0,1]$, was chosen because the source studies do not report the mixing parameter and a uniform ratio applies mild and strong mixing equally often. Mixing is disabled for the validation and test images, so the reported metrics refer to unmodified samples.

E. Hybrid Texture and Deep Feature Representation

1) Centre-Symmetric Local Multilevel Pattern Texture Descriptor: Bruises, wrinkles and rot spots change the local intensity structure of the fruit surface. The CS-LMP descriptor [13] encodes this structure by comparing centre-symmetric neighbour pairs at several quantisation levels. For $P = 8$ neighbours $g_0, \ldots, g_7$ at radius $R = 1$ around a pixel of the grey-level segmented image resized to 128×128 pixels, each pixel receives $P/2 = 4$ codes:

$$c_p(x,y) = s\big(g_p - g_{p+P/2}\big), \quad p = 0, 1, \ldots, \tfrac{P}{2} - 1 \tag{12}$$

where the quantiser s assigns the level index t of the interval that contains its argument:

$$s(z) = \begin{cases} 1, & z \le -d_2 \\ 2, & -d_2 < z \le -d_1 \\ 3, & -d_1 < z \le d_1 \\ 4, & d_1 < z \le d_2 \\ 5, & z > d_2 \end{cases} \tag{13}$$

with thresholds $d_1 = 5$ and $d_2 = 15$ grey levels, which gives $2m + 1 = 5$ levels for $m = 2$. Following the sliding-window idea of the original study, the image is divided into a 4×4 grid of cells; each cell contributes a normalised 20-bin histogram of the four direction codes over five levels. A global histogram of the combined pattern

$$C(x,y) = \sum_{p=0}^{P/2-1} \big(c_p(x,y) - 1\big)\, 5^p \tag{14}$$

adds $5^4 = 625$ bins. The resulting texture vector f_1 has $16 \times 20 + 625 = 945$ elements, and every histogram sums

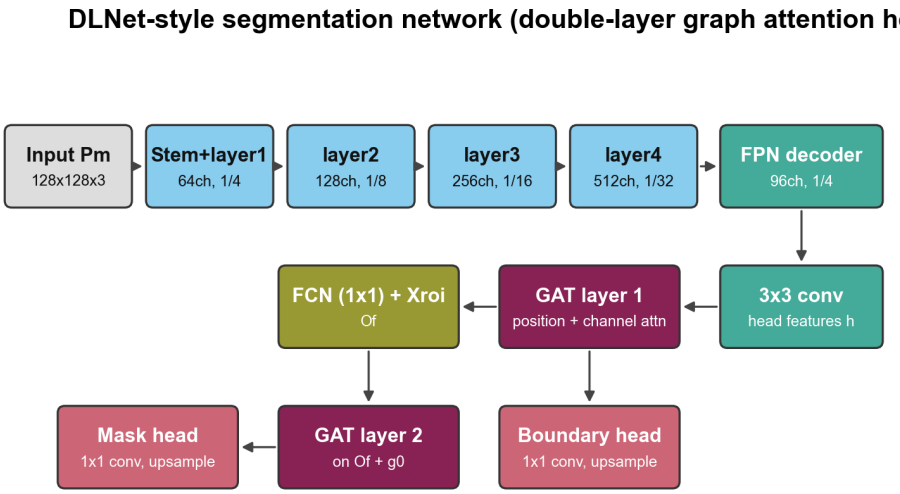


Fig. 2. Architecture of the DLNet-style segmentation network with the double-layer graph attention head used for background isolation.

to one. Compared with binary patterns, the multilevel quantiser distinguishes weak and strong intensity changes, which separates gentle colour gradients from sharp lesion boundaries.

2) ResNeXt-101 Deep Descriptor: The second descriptor follows the deep-feature strategy of Yuan and Chen [2]. The segmented image is resized to 224×224 pixels, normalised with ImageNet statistics and passed through an ImageNet-pretrained ResNeXt-101 (32×8d) [26]. The 1000-dimensional output of the final fully connected layer (fc1000) forms the deep vector f_2. ResNeXt aggregates many parallel residual transformations of equal topology, which gives a rich high-level description at moderate design complexity. The feature representation of image m is $F_m = \{f_1, f_2\}$. Both vectors are standardised with the mean and standard deviation of the training split only, so no information from the test images enters the model:

$$\tilde{f}_{r,d} = \frac{f_{r,d} - \mu_{r,d}}{s_{r,d} + 10^{-6}}, \quad r \in \{1, 2\} \tag{15}$$

where $f_{r,d}$ is element d of vector f_r, and $\mu_{r,d}$ and $s_{r,d}$ are the mean and standard deviation of that element over the training images. The two descriptors are complementary by design. CS-LMP is a local, rotation-sensitive histogram that responds to fine surface patterns but carries no semantic knowledge, whereas the ResNeXt-101 vector summarises shape, colour and object-level appearance learned from ImageNet but is computed at a fixed 224×224 resolution that can blur small defects. Fusing them lets the classifier combine both views; the experiments do not contain a feature-level ablation, so this complementarity is a design argument rather than a measured effect.

F. Mean Squared LogCosh Residual Inception Network

MLRI-Net is the core contribution of this work. Its name describes its two defining elements: a residual-attention-inception image branch derived from RAI-Net [5], and a mean squared logarithmic and log-cosh training objective adapted from regression losses [16]. Fig. 3 shows the network. The image branch, the fusion stage and the two prediction heads are described below.

1) Residual-Attention-Inception Image Branch: The image branch receives the segmented image resized to 128×128 pixels, the input size that RAI-Net found most effective [5]. An ImageNet-pretrained ResNet18 [3] processes it up to its third stage, which yields a $256 \times 8 \times 8$ feature map. These layers are frozen, so their output is computed once per image and reused by every training run. The fourth residual stage is trainable and produces a $512 \times 4 \times 4$ map F. Each residual block computes $x_{l+1} = \mathcal{F}(x_l) + x_l$, where $\mathcal{F}(\cdot)$ is a sequence of convolution, batch normalisation and ReLU operations.

Freezing the first three stages follows the transfer-learning practice of RAI-Net, which also fine-tunes only the last layers of ResNet18 [5]. The early stages encode generic edges and colour blobs that transfer well from ImageNet to fruit images, whereas the fourth stage, which carries the most class-specific information, is adapted to the grading task. This choice also keeps the number of trainable parameters moderate for training sets of a few thousand images.

Squeeze-and-excitation channel attention [12] then reweights the channels of F:

$$\begin{aligned} F_{avg} &= \frac{1}{HW} \sum_{i=1}^{H} \sum_{j=1}^{W} F(:, i, j), \\ M_c &= \sigma_s\big(W_2\, \delta(W_1 F_{avg} + b_1) + b_2\big) \end{aligned} \tag{16}$$

where $H = W = 4$ here, W_1 reduces the 512 channels by a ratio of 16, W_2 restores them, b_1 and b_2 are biases, δ is the ReLU function and σ_s is the sigmoid function. As in RAI-Net, the attended map is added to the residual output:

$$y_{CA} = F \odot M_c + F \tag{17}$$

so that attention can emphasise informative channels without

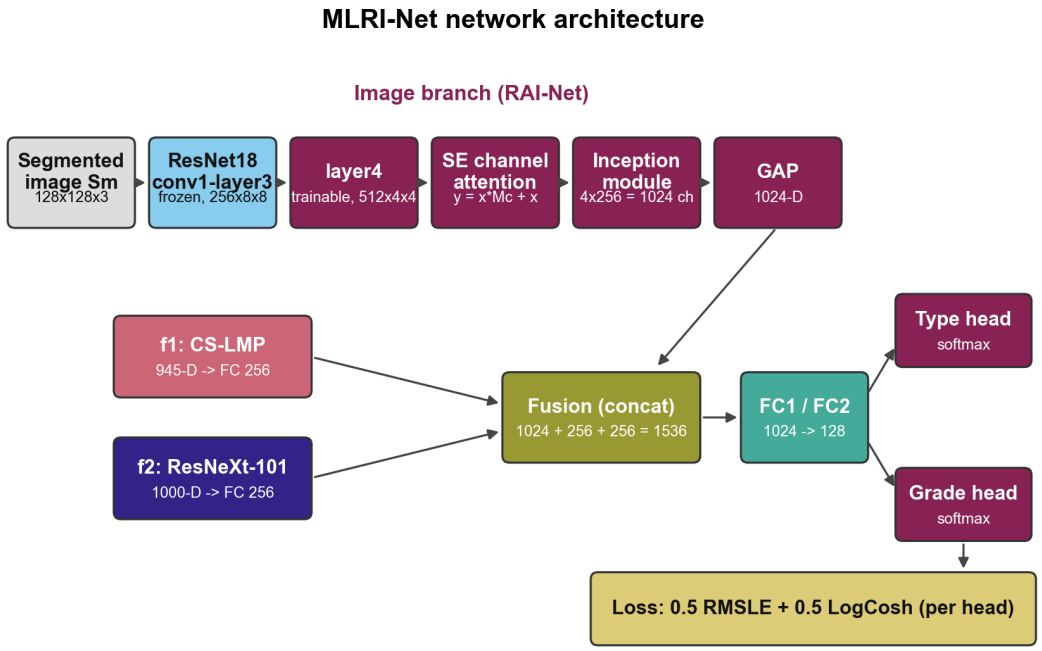


Fig. 3. Network architecture of MLRI-Net: residual-attention-inception image branch, fusion with the CS-LMP and ResNeXt-101 descriptors, joint type and grade heads and the combined loss.

removing information. An inception module [4] then extracts multi-scale patterns from y_{CA}:

$$y_{inc} = \text{concat}\big(C_1(y_{CA}),\, C_3(C_1(y_{CA})),\, C_5(C_1(y_{CA})),\, C_1(\text{MP}_3(y_{CA}))\big) \quad (18)$$

where C_k denotes a $k \times k$ convolution followed by batch normalisation and ReLU, and MP_3 is a 3×3 max pooling with stride 1. Each branch outputs 256 channels, so y_{inc} has 1024 channels, and global average pooling turns it into the 1024-dimensional image vector v.

2) Feature Fusion and Joint Type–Grade Heads: The texture and deep vectors are projected to 256 dimensions each and concatenated with the image vector:

$$z = \big[v;\, \phi_1(f_1);\, \phi_2(f_2)\big] \in \mathbb{R}^{1536} \quad (19)$$

where ϕ_1 and ϕ_2 are fully connected layers with ReLU activation and dropout 0.3. Two shared fully connected layers, FC1 with 1024 units and FC2 with 128 units, map z to the embedding e. Two linear heads then produce the type and grade probabilities:

$$\hat{y}^{type} = \text{softmax}(W_t e + b_t), \quad \hat{y}^{grade} = \text{softmax}(W_g e + b_g) \quad (20)$$

where W_t, b_t, W_g and b_g are the head parameters. Sharing e couples the two tasks, so cues learned for one task can inform the other, while the separate heads allow different class sets: six types and three grades for FruitNet, and eleven types and three grades for FruitQ.

The image vector keeps its full 1024 dimensions, whereas each descriptor is compressed to 256 dimensions, so the image branch provides two thirds of the fused vector. This weighting keeps the end-to-end trained branch dominant and lets the fixed descriptors act as auxiliary evidence. The 128-dimensional embedding e is also the representation visualised with UMAP in Section IV-E.

3) Mean Squared LogCosh Objective: Instead of cross-entropy, MLRI-Net is trained with a combination of two regression losses surveyed by Jadon *et al.* [16], applied to the softmax probabilities of each head. For a head with predicted probabilities $\hat{y}$ and one-hot or mixed targets y over N elements (samples $\times$ classes), the RMSLE term is

$$L_R = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\big(\log(y_i+1) - \log(\hat{y}_i+1)\big)^2} \quad (21)$$

and the log-cosh term is

$$L_{LC} = \frac{1}{N}\sum_{i=1}^{N}\log\big(\cosh(\hat{y}_i - y_i)\big) \quad (22)$$

where the mean replaces the sum of the original definition so that both terms have the same scale. The loss of one head and the total training loss are

$$L_{Tot} = \lambda_1 L_R + \lambda_2 L_{LC}, \quad L = L_{Tot}^{type} + L_{Tot}^{grade} \quad (23)$$

with scaling factors $\lambda_1 = \lambda_2 = 0.5$. Log-cosh is numerically evaluated as $d + \text{softplus}(-2d) - \log 2$ for $d = \hat{y} - y$, which avoids overflow.

The gradients of the two terms with respect to a predicted

probability explain their behaviour:

$$\frac{\partial L_{LC}}{\partial \hat{y}_i} = \frac{1}{N}\tanh(\hat{y}_i - y_i), \qquad \frac{\partial L_R}{\partial \hat{y}_i} = \frac{\log(\hat{y}_i+1) - \log(y_i+1)}{N L_R(\hat{y}_i+1)} \tag{24}$$

The log-cosh gradient is bounded by $1/N$ in magnitude, so no single sample can dominate an update, and the RMSLE gradient is normalised by the current loss value L_R, which keeps its scale stable as training progresses. Because both losses act on probabilities in $[0, 1]$, their gradients are smaller than those of cross-entropy for confidently wrong predictions; the adaptive step size of Adam compensates for this scale difference.

The design rationale is as follows. Log-cosh behaves like a squared error for small residuals and like an absolute error for large ones, which limits the influence of ambiguous or mislabelled samples; the logarithmic RMSLE term compresses large probability errors and penalises under-estimation of the true class more than over-estimation. Both terms operate directly on probabilities, so they accept the soft targets produced by CutMix and MixUp without modification. These properties are design arguments; the experiments do not include a cross-entropy variant of MLRI-Net, so the effect of the loss alone is not measured.

4) Optimisation and Training Algorithm: MLRI-Net is optimised with Adam ($\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 10^{-8}$) at a learning rate of 10^{-3} with batch size 32. A ReduceLROnPlateau schedule divides the learning rate by ten when the validation loss stops improving, and training stops after four epochs without improvement or after 15 epochs; the weights of the best validation epoch are restored. Algorithm 1 summarises the training procedure.

5) Reproducibility and Leakage Controls: Every random choice of the method, including subset selection, data splits, weight initialisation, batch order and mixing ratios, is driven by a fixed seed of 42, and the split of every experiment is saved to disk and reused by all methods. Only the training part of a split is used to fit the feature scalers and to select the best epoch, and the test part is touched once, after training, to compute the metrics. The pseudo-masks and the segmentation network use no grade or type labels, so training the segmenter on images that later appear in a test split cannot leak class information. The models trained on the 80% fraction are stored as reference models and are reused for the confusion matrices, the embedding visualisation, the explanation maps and the end-to-end sample outputs, so that all qualitative results refer to the same trained networks as the quantitative results.

Algorithm 1 Training procedure of MLRI-Net (one training run).

Require: training indices D_{tr}, validation indices D_{val}, cached maps L_3 ($256\times8\times8$), vectors f_1, f_2, labels (t, g), epochs $E = 15$, patience $P = 4$, batch size $B = 32$
Ensure: trained MLRI-Net parameters Θ^*

```
fit z-score scalers on f1[D_tr], f2[D_tr]; standardise f1, f2 for D_tr and D_val
initialise layer4 from ImageNet ResNet18; initialise SE, inception, fusion, FC1, FC2, heads
for epoch = 1, ..., E do
    for each mini-batch b of size B drawn from a random permutation of D_tr do
        build one-hot targets y_type, y_grade
        draw CutMix or MixUp with p = 0.5 and λ ~ Beta(1, 1); mix L3[b], f1[b], f2[b], y_type, y_grade
        (ŷ_type, ŷ_grade) ← MLRI-Net(L3[b], f1[b], f2[b])
        L ← 0.5 RMSLE(ŷ_type, y_type)
              + 0.5 LogCosh(ŷ_type, y_type)
              + 0.5 RMSLE(ŷ_grade, y_grade)
              + 0.5 LogCosh(ŷ_grade, y_grade)
        update Θ with Adam (lr 10^-3, β1 0.9, β2 0.999, ε 10^-8)
    end for
    compute validation loss on D_val (no mixing); step ReduceLROnPlateau
    if validation loss improved: Θ* ← Θ else stale ← stale +1
    if stale = P: break
end for
return Θ*
```

G. Layer-Ordered and Relevance-Based Grading Explanations

To make the grading decisions inspectable, two explanation methods are applied to the image branch with the texture and deep vectors held constant. The first follows the Layer Ordered Visualization of Information (LOVI) used in CathAI [17]. Because the original publication describes LOVI only at the level of its visual output, this work implements it as a gradient-weighted activation map computed at every stage of the image branch in layer order. For a feature map A_ℓ of stage ℓ and the logit y_k of the predicted grade k,

$$S_\ell = \mathrm{norm}\left(\mathrm{up}\left(\mathrm{ReLU}\left(\sum_c \frac{\partial y_k}{\partial A_\ell^c} \odot A_\ell^c\right)\right)\right), \qquad S = \mathrm{norm}\left(\frac{1}{|\mathcal{L}|}\sum_{\ell\in\mathcal{L}} S_\ell\right) \tag{25}$$

where c indexes channels, $\mathrm{up}(\cdot)$ is bilinear upsampling to 128×128 pixels, $\mathrm{norm}(\cdot)$ rescales to $[0, 1]$ and $\mathcal{L}$ contains the six stages layer1, layer2, layer3, layer4, attention and inception. The sequence $S_1, \ldots, S_6$ shows how evidence evolves from low-level edges to class-specific regions, and S is the combined overlay.

Ordering the maps by layer serves a diagnostic purpose. If a decision is driven by the fruit, the evidence should move from scattered edge responses in the first stages to compact regions on the fruit surface in the attention and inception stages; evidence that stays on the background in late stages indicates that the model relies on context rather than on the produce.

The second method is layer-wise relevance propagation [19], as used by Iqbal *et al.* [18]. Relevance R is propagated

from the predicted grade logit to the input pixels. Convolutions use the z^+ rule and fully connected layers use the ϵ rule:

$$R_i = \sum_j \frac{a_i w_{ij}^+}{\sum_{i'} a_{i'} w_{i'j}^+} R_j, \quad R_i = \sum_j \frac{a_i w_{ij}}{z_j + \epsilon\,\mathrm{sign}(z_j)} R_j \tag{26}$$

where a_i is the input activation of neuron i, w_{ij} the weight to neuron j, w^+ its positive part, z_j the pre-activation of neuron j and $\epsilon = 10^{-6}$. Both rules are implemented through modified gradients, the squeeze-and-excitation gates are treated as constants, and the pixel relevance map is the channel sum of the input multiplied by its modified gradient. The map is scaled by the 99.5th percentile of its absolute values so that a few extreme pixels do not hide the overall pattern.

The two explanations are complementary. LOVI shows where each stage of the network looks, and LRP decomposes the final grade score into signed pixel contributions. In Section IV-E, the share of positive relevance and saliency inside the predicted fruit mask is reported as a simple check that the explanations concentrate on the fruit rather than on the background:

$$\rho = \frac{\sum_{(x,y)\in\Omega} \max\big(R(x,y),0\big)}{\sum_{(x,y)} \max\big(R(x,y),0\big)} \tag{27}$$

where Ω is the set of pixels inside the predicted fruit mask and R is the relevance or saliency map. A value of ρ above the mask's area fraction indicates that evidence concentrates on the fruit. This measure checks localisation only; it does not establish that the explanation is faithful to the model's reasoning.

H. Computational Design for CPU-Only Deployment

The framework is designed to run on a laptop without a graphics processor. All heavy, frozen components, namely the CIEWF, the segmentation network at inference, the frozen ResNet18 stages and ResNeXt-101, are executed once per image in a resumable caching pass, and their outputs are stored on disk in half or single precision. Every training run then loads only the cached $256 \times 8 \times 8$ maps and the two vectors, so the 75 training runs per dataset of the evaluation protocol reuse the same preprocessing. The trainable part of MLRI-Net (the fourth ResNet18 stage, attention, inception, fusion and heads) operates on 8×8 inputs, which keeps a training epoch below one minute on the test computer. The complete inference pipeline, including the segmentation network, the frozen backbones and ResNeXt-101, requires 114.9 million parameters and about 36.0 giga floating-point operations per image, as counted in Section IV-G.

IV. Experimental Procedure

This section describes how MLRI-Net was tested and reports the observed results. All numbers come from the saved records of a single complete run of the implementation with a fixed seed of 42.

A. Experimental Setup

All experiments ran on a laptop with a 13th-generation Intel Core i5-1334U processor (12 logical cores), 15.7 GB of RAM and integrated Intel Iris Xe graphics, under Windows 11 Home. No graphics processor was used for computation. The implementation is written in Python 3.10.11 with PyTorch 2.5.1 and torchvision 0.20.1 (CPU builds), scikit-learn 1.6.1, NumPy 1.26.4, SciPy 1.15.3, OpenCV 4.10.0, scikit-image 0.24.0, matplotlib 3.8.4 and umap-learn 0.5.7. ImageNet weights for all backbones were taken from torchvision. The caching pass ran at about 1.45 s per image on FruitNet and 1.32 s per image on FruitQ.

B. Datasets and Data Partitioning

Two public fruit-quality datasets were used, and each was processed independently. FruitNet [7] contains 19,526 processed images of six Indian fruits photographed with a mobile phone against varied backgrounds and grouped into Good, Bad and Mixed quality; Mixed images show several fruits of different quality. FruitQ [8] contains images of eleven fruits extracted from time-lapse videos and labelled Good (Fresh), Mild or Rotten; the FruQ-multi partition with 9,370 images was used. Both datasets are public and have been used in prior grading studies [1], but the present experiments use a capped subset and their own splits, so the reported numbers are not directly comparable with published benchmark results. Table I summarises the data used.

For every training fraction $f \in \{50, 60, 70, 80, 90\}\%$, the subset was split with stratification by the type–grade combination: $f\%$ for training, of which 10% served as validation for early stopping, and the remaining $(100-f)\%$ for testing. Five-fold stratified cross-validation was additionally performed. All methods used identical split files, and the scalers of the feature vectors were fitted on the training part only.

C. Parameter Settings

Table II lists the settings used for MLRI-Net in the reported run; the baselines kept the settings of their source studies where these were stated.

Four baselines were reproduced from their source studies. B1 uses frozen EfficientNetV2-S [11] features with a type model and a grade model initialised from it, as in same-domain transfer learning [1] (AdamW, learning rate 0.01, weight decay 10^{-2}, batch 64, cross-entropy, oversampling and MixUp). B2 fuses GoogLeNet [4], DenseNet-201 [27] and ResNeXt-101 features, keeps 95% of the variance with principal component analysis and classifies with a radial-basis support vector machine ($C = 10$) [2]. B3 is RAI-Net on the raw image with cross-entropy and stochastic gradient descent (learning rate 0.001, momentum 0.9, batch 128) [5]. B4 is a simplified MSFNet-CAM [6] that fuses frozen ResNet-50 [3], DenseNet-121 and Swin-T [28] maps of the segmented image with coordinate attention [29] (Adam, learning rate 10^{-4}). Training the large backbones end to end was not feasible on the CPU-only computer, so B1, B3 and B4 keep their backbones frozen, as MLRI-Net does.

TABLE I
DATASETS USED IN THE EXPERIMENTS

Property	FruitNet	FruitQ (FruQ-multi)
Fruit types	6 (apple, banana, guava, lime, orange, pomegranate)	11 (banana, cucumber, grape, kaki, papaya, peach, pear, pepper, strawberry, tomato, watermelon)
Grades	Good, Mixed, Bad	Good, Mild, Rotten
Images available	19,526	9,370
Images used (cap 300 per type–grade class)	4,674	6,634
Images per grade (used)	Good 1,800; Mixed 1,074; Bad 1,800	Good 2,085; Mild 1,998; Rotten 2,551
Split at 80% training fraction	3,365 train / 374 validation / 935 test	4,776 train / 531 validation / 1,327 test

TABLE II
EXPERIMENTAL PARAMETERS OF MLRI-NET

Parameter	Value
Epochs	up to 15, early stopping patience 4 (validation loss)
Optimizer	Adam with ReduceLROnPlateau (factor 0.1)
Learning rate	1×10^{-3}
Beta 1 / Beta 2 / Epsilon	0.9 / 0.999 / 1×10^{-8}
Batch size	32 (sensitivity analysis: 16, 32, 64, 128)
Kernel sizes	7×7 stem; 3×3 residual; 1×1, 3×3, 5×5 inception
Loss	0.5 RMSLE + 0.5 log-cosh per head, summed over the two heads
Activation	ReLU (hidden layers), sigmoid (attention gate), softmax (outputs)
Dropout	0.3
Input sizes	256×256 (CIEWF, segmentation), 128×128 (image branch), 224×224 (ResNeXt-101)
CIEWF windows / CS-LMP thresholds	3, 5, 7 (adaptive) / $d_1 = 5$, $d_2 = 15$
Segmentation network	1,000 pseudo-masks, 8 epochs, batch 16, Adam 1×10^{-3}
Mixing	CutMix or MixUp per batch, $\lambda \sim \mathrm{Beta}(1, 1)$

D. Evaluation Metrics

Each head was evaluated separately. For class k, TP_k, FP_k and FN_k denote true positives, false positives and false negatives, and K is the number of classes. Accuracy, macro precision, macro recall and macro F1-score are

$$\mathrm{Acc} = \frac{\sum_k TP_k}{n_{test}}, \quad P_k = \frac{TP_k}{TP_k + FP_k}, \quad R_k = \frac{TP_k}{TP_k + FN_k} \tag{28}$$

where n_{test} is the number of test images, and

$$F_1 = \frac{1}{K} \sum_{k=1}^{K} \frac{2P_k R_k}{P_k + R_k} \tag{29}$$

is the unweighted mean over classes, which gives minority grades equal weight. The Matthews correlation coefficient [30] in its multiclass form is

$$\mathrm{MCC} = \frac{cs - \sum_k p_k t_k}{\sqrt{\left(s^2 - \sum_k p_k^2\right)\left(s^2 - \sum_k t_k^2\right)}} \tag{30}$$

where c is the number of correct predictions, s the number of samples, p_k the number of times class k was predicted and t_k the number of times it occurred. Threshold-independent performance is measured with the macro one-vs-rest area under the receiver operating characteristic curve (AUC) and the macro area under the precision–recall curve (AUPR), averaged over the classes present in the test set. All five metrics are dimensionless and lie in $[0, 1]$, with MCC in $[-1, 1]$; tables report them as fractions and the text as percentages. Training time is reported in seconds (s) and computational cost in floating-point operations (FLOPs).

E. Experimental Results

Fig. 4 shows the processing stages for representative test images: the input, the CIEWF output, the DLNet mask, the segmented fruit, the CutMix and MixUp examples and the CS-LMP code map. CIEWF produced a mean peak signal-to-noise ratio of 34.4 dB on FruitNet and 38.0 dB on FruitQ relative to the input, with mean structural similarity of 0.826 and 0.958. The adaptive rule chose the 7×7 window for 77.3% of FruitNet images and 69.4% of FruitQ images, and the estimated noise level fell from 0.0105 to 0.0028 on FruitNet and from 0.0039 to 0.0021 on FruitQ. The lower similarity on FruitNet reflects its noisier phone images, on which the filter removed more content.

Fig. 5 shows the segmentation training. Intersection over union with the pseudo-masks rose from 0.693 to 0.875 on FruitNet and from 0.801 to 0.944 on FruitQ within eight epochs. The masks covered on average 22.4% of a FruitNet image and 29.7% of a FruitQ image. Fig. 4 also shows the main failure mode: on FruitNet, some masks cover only part of a fruit or of a group of fruits, and a cyan cloth was kept as foreground in one sample, so the crop occasionally discards useful fruit area.

Fig. 6 shows the training curves of MLRI-Net at the 80% training fraction; validation grading accuracy exceeded 95% by the third epoch on FruitNet and 98.9% from the first epoch on FruitQ, and training ended after 15 and 13 epochs, respectively. Fig. 7 shows the grade confusion matrices. On FruitNet, MLRI-Net recognised 97.5% of Good, 96.7% of Mixed and 99.2% of Bad test images, and most errors confused Good with Mixed. On FruitQ, the recall was 100% for Good, 99.5% for Mild and 99.8% for Rotten. Fig. 8 shows the type confusion matrices; FruitQ types were always correct, whereas FruitNet errors occurred mainly between visually similar fruits, most often limes predicted as oranges and apples predicted as guavas or limes. The UMAP projection [31] of the 128-dimensional embedding in Fig. 9 forms compact clusters per fruit type and largely separates grades within each type, which is consistent with the shared representation serving both heads.

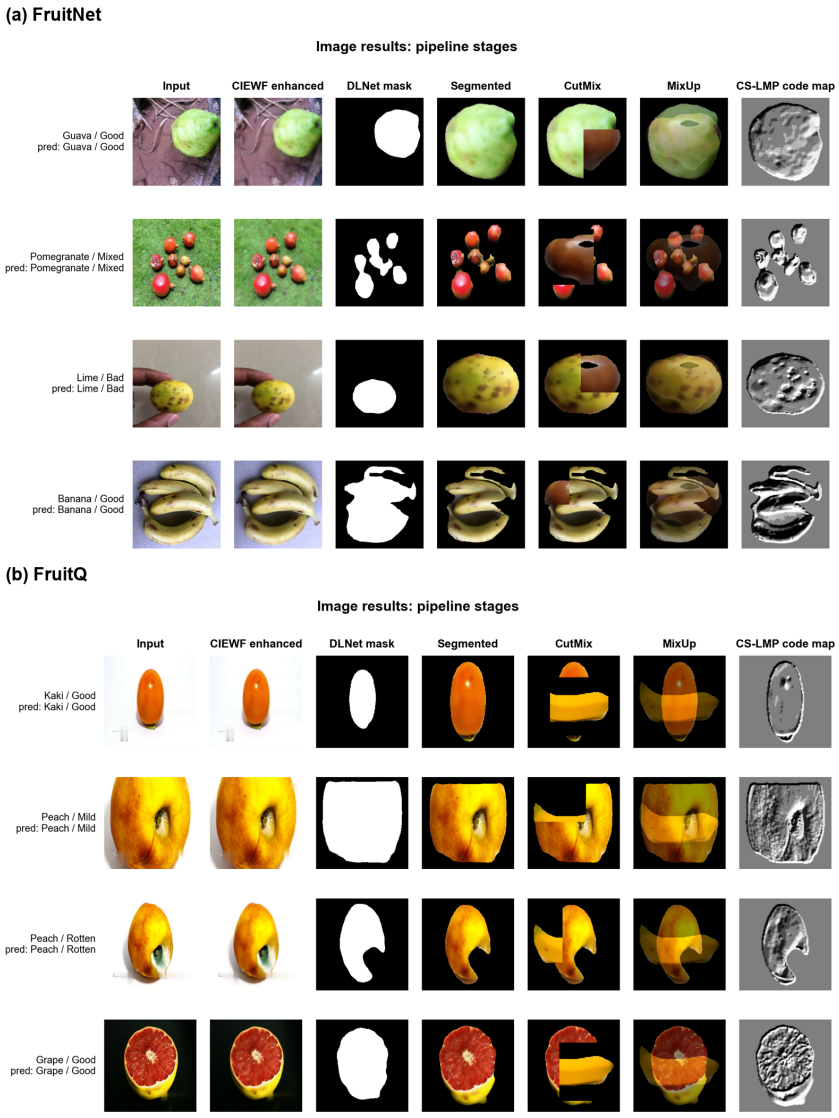


Fig. 4. Step-by-step outputs for representative test images: input, CIEWF enhancement, DLNet mask, segmented fruit, CutMix and MixUp examples and CS-LMP code map, with true and predicted labels.

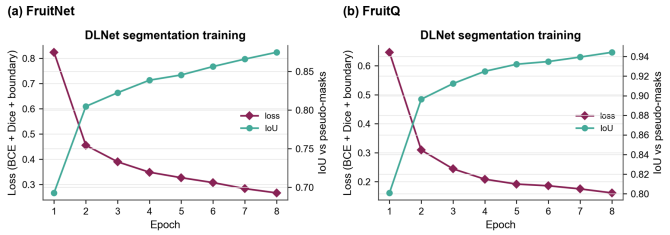


Fig. 5. Training loss and intersection over union with the pseudo-masks of the DLNet-style segmentation network.

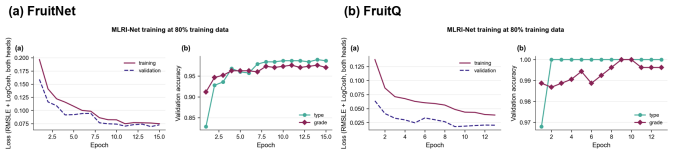


Fig. 6. Training and validation loss and validation accuracy of MLRI-Net at the 80% training fraction.

Figs. 10 and 11 present the explainability results. In the LOVI sequence, early stages respond to edges and texture across the whole crop, whereas the attention and inception stages concentrate on fewer regions; for one FruitNet pomegranate sample the inception-stage map was empty because no channel increased the grade score. On average, 66–68% of the layer3, layer4 and attention saliency fell inside the fruit mask on FruitNet. LRP placed 64.6–74.9% of the positive relevance inside the fruit mask on FruitNet, whose masks covered 15.2–46.5% of these images, and 57.4–78.0% on FruitQ. Relevance concentrated along fruit contours and on dark blemishes of Bad and Rotten samples. These observations show that the explanations focus mainly on the fruit, but no quantitative faithfulness test or expert assessment was performed.

Twelve further test images, six per dataset, were processed end to end and saved with all intermediate outputs, feature

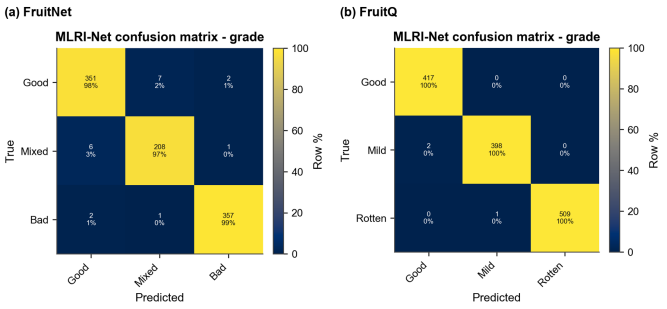


Fig. 7. Grade confusion matrices of MLRI-Net at the 80% training fraction (counts and row percentages).

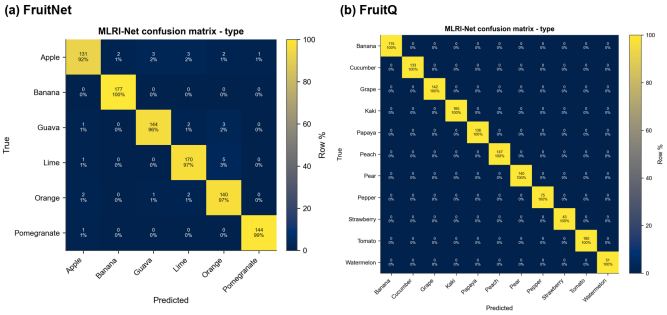


Fig. 8. Fruit-type confusion matrices of MLRI-Net at the 80% training fraction.

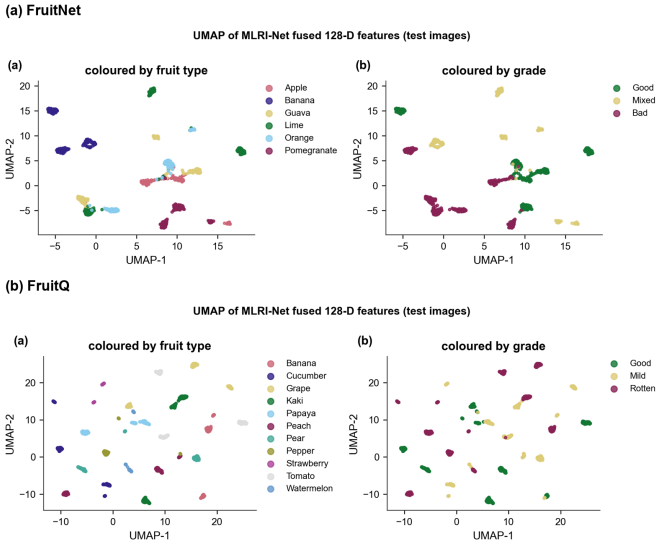


Fig. 9. UMAP projection of the MLRI-Net embedding of the test images, coloured by fruit type and by grade.

visualisations and the predictions of every method; MLRI-Net predicted the correct type and grade for all twelve.

F. Comparative Analysis with Baseline Methods

Fig. 12 plots grading accuracy, MCC and F1-score against the training fraction, and Fig. 13 plots fruit-type accuracy. On FruitNet, B2 achieved the highest grading accuracy at every fraction, from 98.55% at 50% to 99.79% at 90%. MLRI-Net improved steadily with more training data, from 96.28% to 98.08%, matched B3 at 80% (97.97%) and outperformed B1 at every fraction from 60% onwards. On FruitQ, all methods exceeded 99.3% grading accuracy; MLRI-Net achieved the highest or joint-highest accuracy at the 50%, 70% and 90% fractions (99.73%, 99.80% and 99.70%) and tied with B3 at 80% (99.77%), whereas B3 led at 60%. For fruit type, every method reached 100% on FruitQ at every fraction. On FruitNet, MLRI-Net reached 95.93–97.51% type accuracy and was below B2 and B4 at every fraction and below B1 at four of the five fractions.

Table III reports all metrics at the 80% training fraction, which is also shown as a bar chart in Fig. 14; the MCC and F1-score curves over all fractions follow the same ordering as accuracy and are provided with the supplementary graphs.

G. Ablation, Sensitivity, Confidence and Complexity Analyses

Fig. 15 shows the ablation as raincloud plots [32] of bootstrap distributions. Removing CutMix and MixUp reduced FruitQ grading accuracy at every fraction, by 1.24 to 6.63 percentage points (for example 99.70% versus 93.07% at 90%), and reduced FruitNet grading accuracy at every fraction by 0.26 to 2.78 points. Removing the DLNet stage had the opposite effect on FruitNet: grading accuracy rose to 97.81–99.57% and type accuracy to 98.40–99.15%, up to 2.9 points above the full model. On FruitQ, the variant without DLNet differed from the full model by at most 0.27 points. The segmentation stage therefore did not improve accuracy in these experiments, consistent with the partial masks observed in Section IV-E.

The radar plots in Fig. 16 summarise the batch-size sensitivity. On FruitNet, grading accuracy ranged from 96.28% to 99.36% across batch sizes 16, 32, 64 and 128, and batch size 64 was more accurate than the default of 32 at all five fractions, whereas batch size 128 was better at only two. On FruitQ, all batch sizes stayed between 99.40% and 100%, so the model was insensitive to this parameter. The confidence analysis in Fig. 17, drawn as violin and swarm plots [33], shows that accuracy increased with the maximum softmax probability for all methods and that the highest-confidence bin (0.97–1.00) held the largest share of MLRI-Net test predictions: 447 of 935 on FruitNet and 1,004 of 1,327 on FruitQ. Fig. 18 relates accuracy to computational cost in the manner of Asif *et al.* [20]. MLRI-Net requires 114.9 million parameters and 36.0 GFLOPs per image, mostly because of ResNeXt-101 and the segmentation network, compared with 13.5 million parameters and 1.2 GFLOPs for B3, 22.1 million and 5.7 GFLOPs for B1, 70.9 million and 21.8 GFLOPs for B4 and 115.4 million and 44.4 GFLOPs for B2. These counts include convolution and linear layers only, so the attention products of Swin-T and of the graph attention layers are not counted.

H. Statistical Findings and Experimental Discussion

Five-fold cross-validation provides the paired observations for statistical testing. Fig. 19 shows the fold-wise AUC curves [34], Fig. 20 the AUPR distributions [35] and Fig. 21 the fold scores with paired t-test results [36]. The deviation plot in Fig. 22 shows how far each fold's grading accuracy deviates

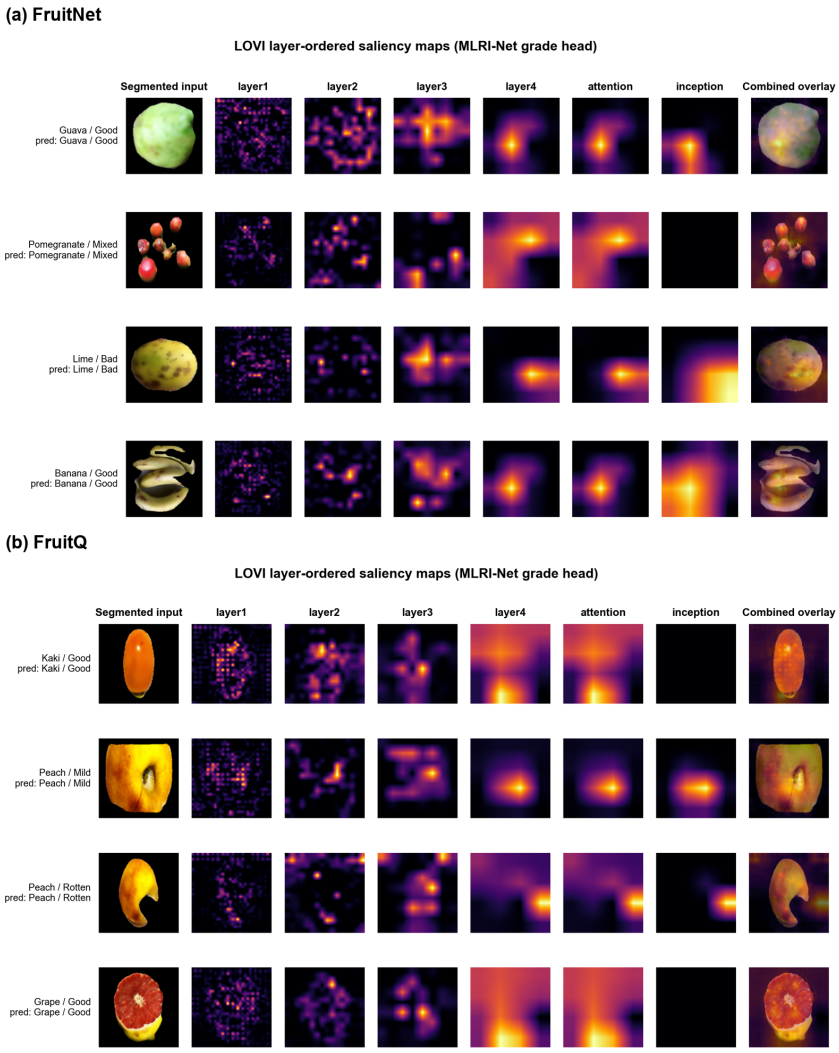


Fig. 10. LOVI layer-ordered saliency maps of the MLRI-Net grade head for representative test images.

TABLE III
TEST RESULTS AT THE 80% TRAINING FRACTION (935 FRUITNET AND 1,327 FRUITQ TEST IMAGES). TRAINING TIME EXCLUDES THE SHARED FEATURE-CACHING PASS

Dataset	Method	Grade Acc	Grade MCC	Grade F1	Grade AUC	Type Acc	Train time (s)
FruitNet	B1 EfficientNetV2-S TL	0.9722	0.9575	0.9719	0.9974	0.9786	10
FruitNet	B2 Deep features + PCA + SVM	0.9925	0.9885	0.9923	0.9979	0.9957	28
FruitNet	B3 RAI-Net (CE)	0.9797	0.9688	0.9793	0.9979	0.9786	863
FruitNet	B4 MSFNet-CAM	0.9840	0.9755	0.9831	0.9980	0.9775	116
FruitNet	MLRI-Net (proposed)	0.9797	0.9688	0.9777	0.9964	0.9690	615
FruitQ	B1 EfficientNetV2-S TL	0.9947	0.9920	0.9946	0.9999	1.0000	12
FruitQ	B2 Deep features + PCA + SVM	0.9955	0.9932	0.9953	1.0000	1.0000	27
FruitQ	B3 RAI-Net (CE)	0.9977	0.9966	0.9978	1.0000	1.0000	1,129
FruitQ	B4 MSFNet-CAM	0.9932	0.9898	0.9932	0.9999	1.0000	101
FruitQ	MLRI-Net (proposed)	0.9977	0.9966	0.9976	1.0000	1.0000	697

from the method's mean. On FruitNet, MLRI-Net deviated by up to ± 1.6 percentage points between folds, the largest spread of all methods, whereas on FruitQ every method stayed within ± 0.3 points. MLRI-Net achieved a mean grading accuracy of $99.85 \pm 0.08\%$ on FruitQ, higher than every baseline, and paired t-tests over the five folds gave $p = 0.0027$ (B1), 0.0026 (B2), 0.0186 (B3) and 0.0030 (B4). On FruitNet, its mean grading accuracy of $97.20 \pm 1.11\%$ was significantly lower than that of B2 (99.12%, $p = 0.0073$) and B4 (97.67%, $p = 0.0376$) and not significantly different from B1 ($p = 0.756$) or B3 ($p = 0.240$). Each test compares the five paired fold scores of two methods and assumes approximately normal fold differences; with five folds the tests have limited power, and no correction for multiple comparisons was applied.

The experiments therefore support the objective on FruitQ, where joint prediction achieved perfect type recognition and

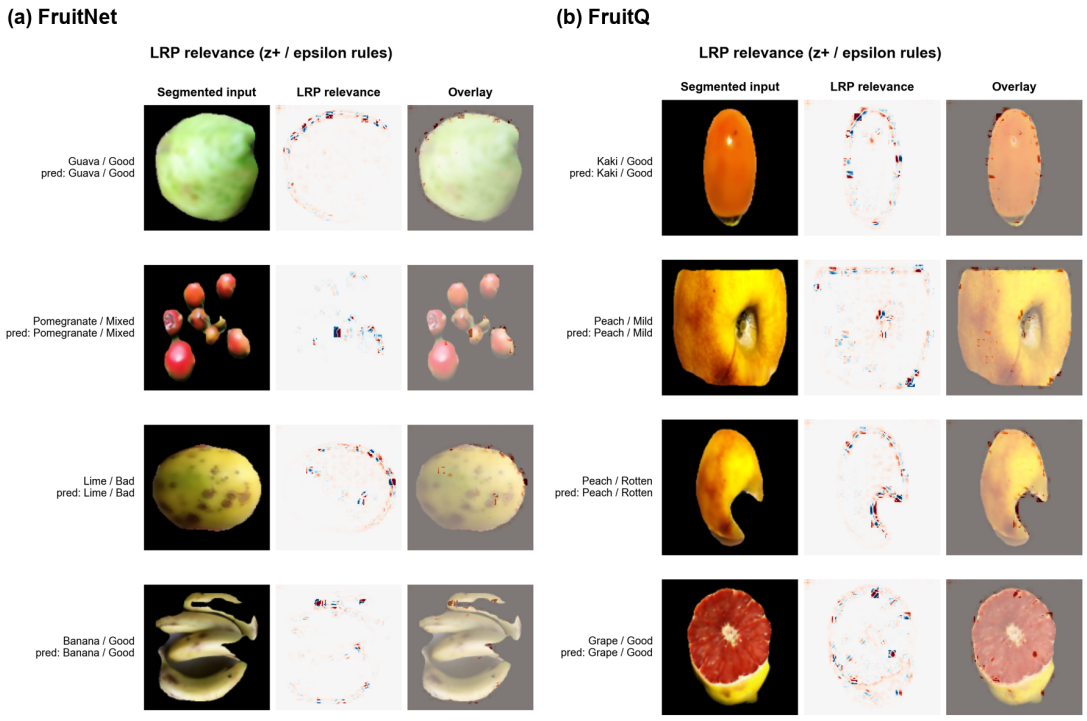


Fig. 11. LRP relevance maps (z^+ and ϵ rules) of the MLRI-Net grade head.

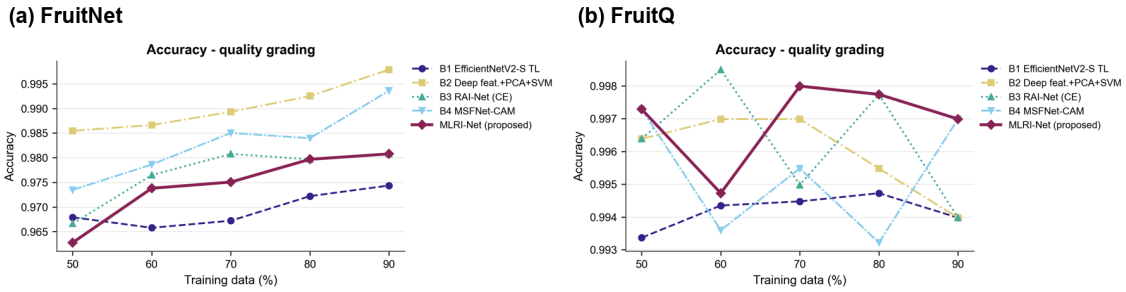


Fig. 12. Grading accuracy of all methods versus the training-data percentage.

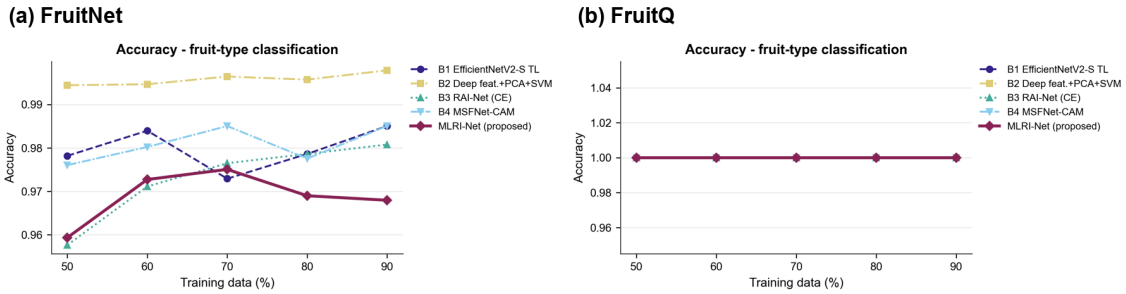


Fig. 13. Fruit-type accuracy of all methods versus the training-data percentage.

the best grading performance, and only partly on FruitNet. The likely reason for the difference is the acquisition setting: FruitQ frames show single fruits from fixed time-lapse viewpoints, whereas FruitNet phone images contain hands, groups of fruits and varied backgrounds, which made the pseudo-mask segmentation less reliable. Because MLRI-Net classifies the cropped segmentation output, segmentation errors propagate to both heads, which is consistent with the ablation result and the larger fold variance on FruitNet.

V. Discussion

The main achievement of this study is a single explainable network that predicts fruit type and quality grade together and that was evaluated with a complete protocol on two public datasets. Table IV summarises the five-fold cross-validation results by dataset. On FruitQ, MLRI-Net achieved the highest

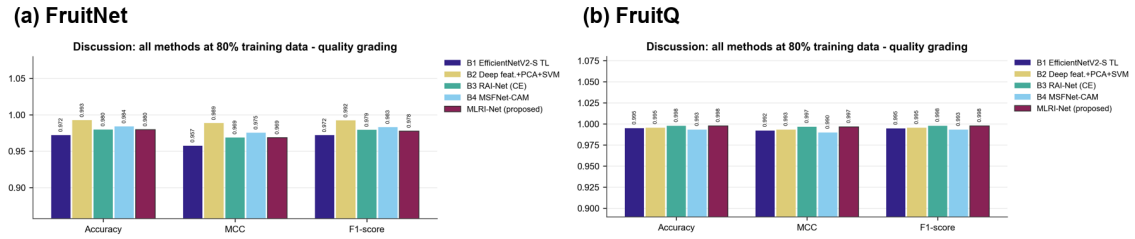


Fig. 14. Grading accuracy, MCC and F1-score of all methods at the 80% training fraction.

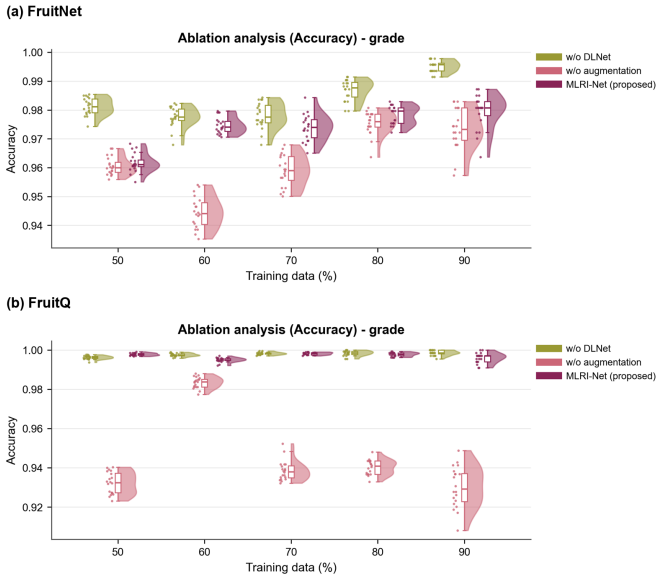


Fig. 15. Ablation of MLRI-Net for grading: without DLNet, without augmentation and the full model (raincloud plots of 20 bootstrap resamples of the test set).

TABLE IV
FIVE-FOLD CROSS-VALIDATION SUMMARY (MEAN OVER FOLDS; p FROM PAIRED T-TESTS AGAINST MLRI-NET)

Dataset	Method	Grade Acc (mean ± SD)	Grade MCC	Grade F1	Type Acc	p (grade Acc)
FruitNet	B1	0.9707 ± 0.0047	0.9552	0.9693	0.9822	0.756
FruitNet	B2	0.9912 ± 0.0052	0.9866	0.9912	0.9964	0.007
FruitNet	B3	0.9771 ± 0.0048	0.9650	0.9765	0.9807	0.240
FruitNet	B4	0.9767 ± 0.0089	0.9644	0.9758	0.9842	0.038
FruitNet	MLRI-Net	0.9720 ± 0.0111	0.9571	0.9705	0.9722	–
FruitQ	B1	0.9944 ± 0.0020	0.9916	0.9942	1.0000	0.003
FruitQ	B2	0.9959 ± 0.0010	0.9939	0.9958	1.0000	0.003
FruitQ	B3	0.9965 ± 0.0007	0.9948	0.9964	1.0000	0.019
FruitQ	B4	0.9935 ± 0.0015	0.9902	0.9934	1.0000	0.003
FruitQ	MLRI-Net	0.9985 ± 0.0008	0.9977	0.9984	1.0000	–

mean grading accuracy, MCC and F1-score, and the differences from all four baselines were statistically significant. On FruitNet, the fused deep-feature support vector machine (B2) remained clearly superior, and MLRI-Net ranked with the residual baselines.

Table IV shows that the gap between datasets is larger than the gap between methods. FruitQ grading is close to saturation for every method, so the significant improvement of MLRI-Net, about 0.2–0.5 percentage points, matters mainly in the error rate, which fell from 0.35–0.65% for the baselines to 0.15%. On FruitNet, MLRI-Net had the largest fold variance, which indicates sensitivity to the composition of the training data.

Table V places the results beside published studies. These values come from different datasets, splits and labels and are therefore not directly comparable; they indicate the performance range reported for related tasks.

The FruitNet result of MLRI-Net is below the 99.42% grading accuracy reported by Aldakhil and Almutairi [1], who fine-tuned the full EfficientNetV2 network on all FruitNet images; the reproduced B1 baseline with a frozen backbone reached 97.07%, which suggests that freezing backbones on a CPU-only computer cost accuracy for all neural methods in this study.

Several explanations for the observed behaviour are plausible but were not tested directly. The improvement on FruitQ may come from the fused texture and deep features, which capture the gradual surface changes of time-lapse decay, and from CutMix and MixUp, whose removal caused the largest accuracy loss. The weaker FruitNet performance is more clearly linked to background isolation, because the ablation without DLNet outperformed the full model and B2, which uses uncropped images, performed best. The regression-based loss trained stably on both datasets, but its contribution cannot be separated from the other components without a cross-entropy variant of MLRI-Net.

The findings also relate to the challenges identified in the literature survey. Relevance-based explanations, which Iqbal *et al.* [18] and Avram *et al.* [17] used for clinical images, transferred to fruit grading without modification and highlighted lesions and contours, which supports their use as an inspection aid. The DLNet idea [9], however, did not transfer as easily: without instance-level annotations, the learned masks inherited the errors of the automatic cues. For practice, the results suggest that a grading line with controlled backgrounds can use the full framework, whereas uncontrolled phone images need either verified masks or the variant without segmentation, which was the most accurate MLRI-Net configuration on FruitNet.

From a theoretical perspective, the contrast between the datasets is consistent with the bias–variance view of feature pipelines. B2 combines three frozen ImageNet backbones with a strongly regularised classifier, which suits small, heterogeneous data such as FruitNet, whereas MLRI-Net adds trainable layers and a segmentation step that increase variance when the input is noisy. On the cleaner FruitQ images, the additional

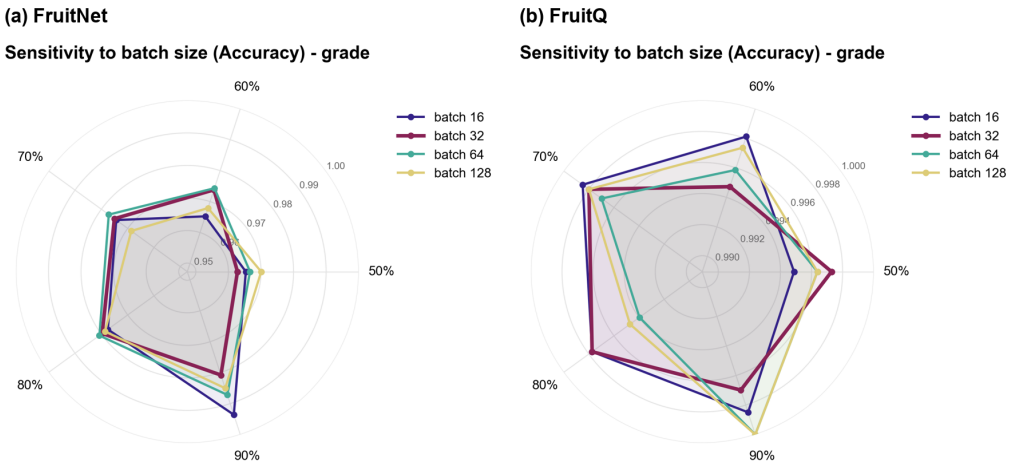


Fig. 16. Sensitivity of MLRI-Net grading accuracy to the batch size across training fractions.

TABLE V
RESULTS REPORTED IN RELATED STUDIES AND IN THIS WORK (NOT DIRECTLY COMPARABLE)

Study	Task and data	Reported result
Aldakhil and Almutairi [1]	Type and grade, full FruitNet, 70/20/10 split	99.49% type, 99.42% grade accuracy
Yuan and Chen [2]	Fresh/rotten, 20 produce categories	96.98% accuracy
Castillo-Gironés *et al.* [10]	Persimmon firmness, hyperspectral, 3 classes	81.3% accuracy
Maurya *et al.* [5]	Tomato leaf disease, 10 classes	97.88% accuracy
This work, MLRI-Net	Type and grade, FruitNet subset, 5-fold CV	97.22% type, 97.20% grade accuracy
This work, MLRI-Net	Type and grade, FruitQ subset, 5-fold CV	100% type, 99.85% grade accuracy

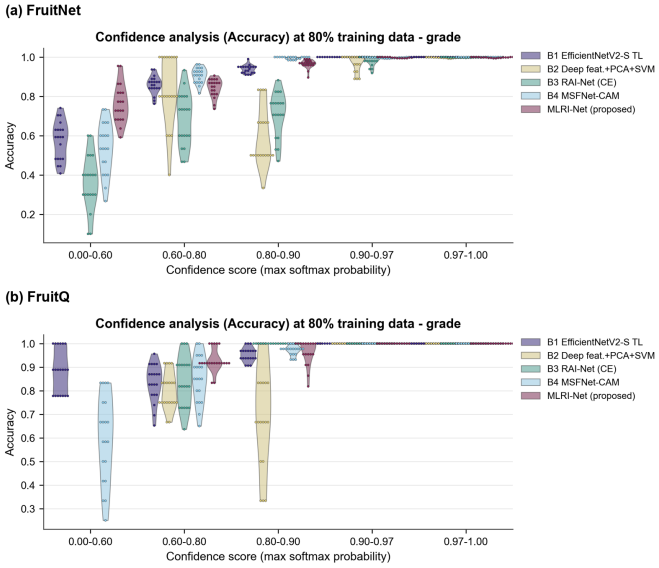


Fig. 17. Confidence analysis: grading accuracy per confidence bin (violin and swarm plots of bootstrap resamples) at the 80% training fraction.

capacity of MLRI-Net became an advantage.

The strengths of the approach are joint prediction, annotation-free segmentation training, explanation maps for every decision and a complete statistical evaluation. Its limitations are the dataset-dependent value of segmentation, a higher computational cost than the residual baselines, frozen backbones, capped dataset subsets, the absence of loss and feature ablations, and explanation assessments that remain qualitative.

VI. CONCLUSION

This study set out to predict fruit type and quality grade jointly and explainably from ordinary colour images. The proposed MLRI-Net enhances images with an adaptive contextual Wiener filter, isolates the fruit with a graph-attention segmentation network trained on automatically generated masks, regularises training with CutMix and MixUp, fuses multilevel texture and ResNeXt-101 features with a residual-attention-inception branch and trains two prediction heads with a combined logarithmic and log-cosh loss. In five-fold cross-validation on FruitQ, it achieved 100% type accuracy and the best grading accuracy of 99.85% (MCC 0.9977), significantly above all four baselines. On FruitNet, it reached 97.20% grading accuracy, below a fused deep-feature support vector machine at 99.12%. Ablation confirmed the value of mixing augmentation, whose removal cut FruitQ grading accuracy by up to 6.63 points, but showed that background isolation reduced accuracy on FruitNet. Layer-ordered saliency and relevance maps placed most evidence on the fruit. Future work will train segmentation with a small set of verified masks, fine-tune the backbones on graphics hardware, isolate the loss and feature contributions and test the grader on images from an operating packing line.

REFERENCES

[1] L. A. Aldakhil and A. A. Almutairi, “Multi-fruit classification and grading using a same-domain transfer learning approach,” *IEEE Access*, vol. 12, pp. 44960–44971, Mar. 2024, doi: 10.1109/ACCESS.2024.3379276.

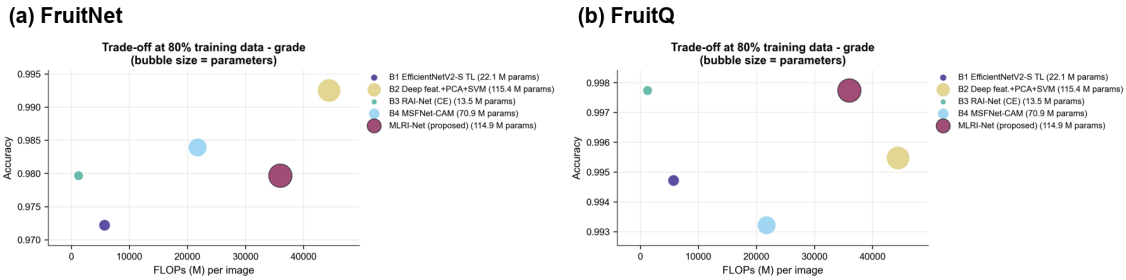


Fig. 18. Trade-off between grading accuracy and FLOPs per image; bubble size indicates the number of parameters.

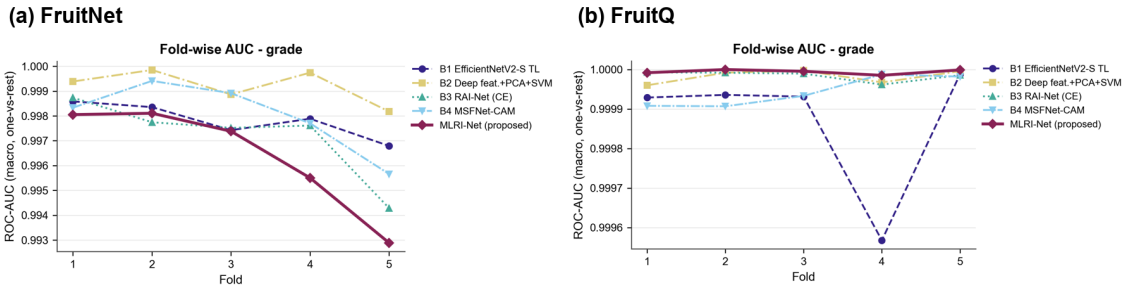


Fig. 19. Fold-wise macro AUC of the grade head in five-fold cross-validation.

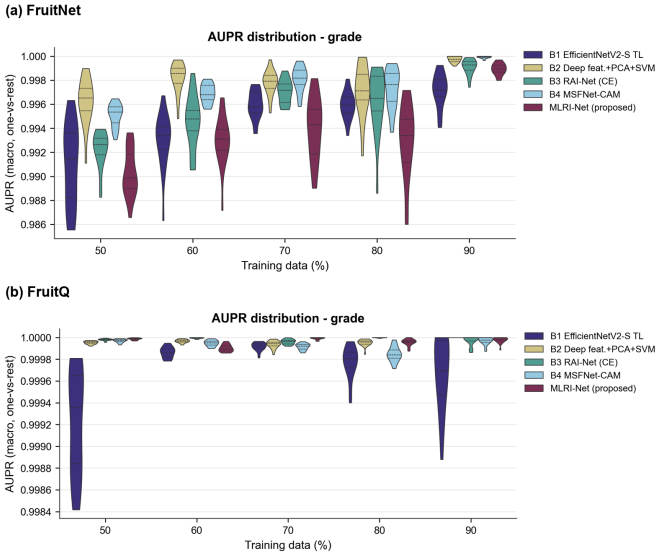


Fig. 20. Distribution of the macro AUPR of the grade head across training fractions (violin plots of bootstrap resamples).

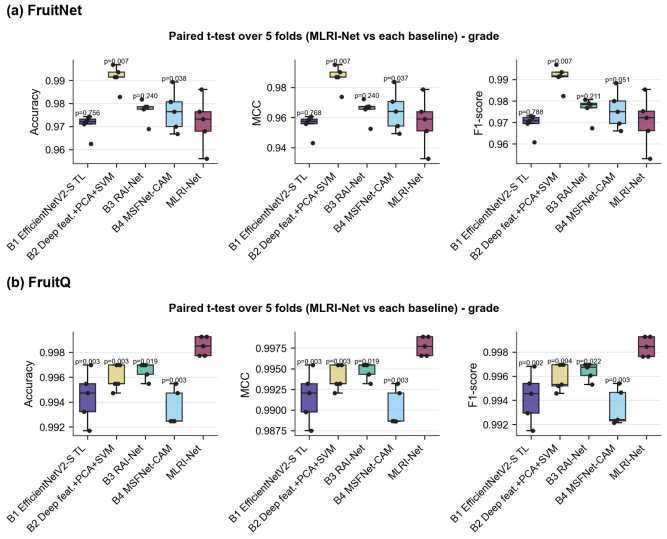


Fig. 21. Fold scores of the grade head with paired t-test p-values of each baseline against MLRI-Net.

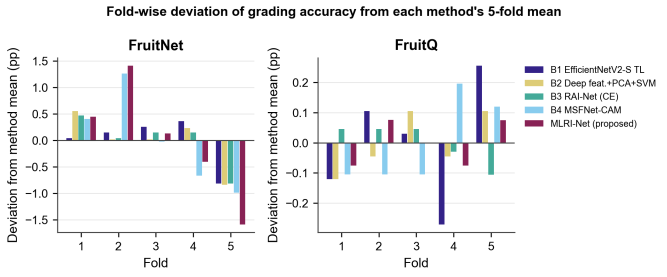


Fig. 22. Fold-wise deviation of grading accuracy from each method's five-fold mean, in percentage points.

[2] Y. Yuan and X. Chen, "Vegetable and fruit freshness detection based on deep features and principal component analysis," *Curr. Res. Food Sci.*, vol. 8, 2024, Art. no. 100656, doi: 10.1016/j.crfs.2023.100656.
[3] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2016, pp. 770–778, doi: 10.1109/CVPR.2016.90.
[4] C. Szegedy *et al.*, "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2015, pp. 1–9, doi: 10.1109/CVPR.2015.7298594.
[5] R. Maurya, L. Rajput, and S. Mahapatra, "RAI-Net: Tomato plant disease classification using residual-attention-inception network," *IEEE Access*, vol. 13, pp. 64832–64840, Apr. 2025, doi: 10.1109/ACCESS.2025.3559804.
[6] R. Sudhakar, K. Nithya, C. R. Dhivyaa, and C. Sharmila, "High-performance parallel multi-scale attention network with explainable AI for intelligent diagnosis of leaf diseases in agricultural systems," *Sci.*

Rep., vol. 15, 2025, Art. no. 42060, doi: 10.1038/s41598-025-26144-4.
[7] V. Meshram and K. Patil, "FruitNet: Indian fruits image dataset with quality for machine learning applications," *Data Brief*, vol. 40, 2022, Art. no. 107686, doi: 10.1016/j.dib.2021.107686.
[8] O. O. Abayomi-Alli, R. Damaševičius, S. Misra, and A. Abayomi-Alli, "FruitQ: A new dataset of multiple fruit images for freshness evaluation," *Multimedia Tools Appl.*, 2024, doi: 10.1007/s11042-023-16058-6.
[9] J. Liu, Y. Zhao, W. Jia, and Z. Ji, "DLNet: Accurate segmentation of green fruit in obscured environments," *J. King Saud Univ.—Comput. Inf. Sci.*, vol. 34, pp. 7259–7270, 2022, doi: 10.1016/j.jksuci.2021.09.023.
[10] S. Castillo-Gironés, Á. González, S. Munera, M. Martínez-Sober, S. Cubero, N. Q. Vilhena, I. Blanco-Álvarez, and J. Gómez-Sanchis, "Firmness classification in 'Rojo Brillante' persimmon fruit after cold storage using hyperspectral imaging and deep learning with variational autoencoders," *LWT*, vol. 229, 2025, Art. no. 118163, doi: 10.1016/j.lwt.2025.118163.
[11] M. Tan and Q. V. Le, "EfficientNetV2: Smaller models and faster training," in *Proc. Int. Conf. Mach. Learn. (ICML)*, PMLR vol. 139, 2021, pp. 10096–10106.
[12] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2018, pp. 7132–7141, doi: 10.1109/CVPR.2018.00745.
[13] E. Al-wajih and R. Ghazali, "An enhanced LBP-based technique with various size of sliding window approach for handwritten Arabic digit recognition," *Multimedia Tools Appl.*, 2021, doi: 10.1007/s11042-021-10762-x.
[14] S. Yun, D. Han, S. J. Oh, S. Chun, J. Choe, and Y. Yoo, "CutMix: Regularization strategy to train strong classifiers with localizable features," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, 2019, pp. 6023–6032, doi: 10.1109/ICCV.2019.00612.
[15] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, "mixup: Beyond empirical risk minimization," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2018.
[16] A. Jadon, A. Patil, and S. Jadon, "A comprehensive survey of regression based loss functions for time series forecasting," 2022, *arXiv:2211.02989*.
[17] R. Avram *et al.*, "CathAI: Fully automated coronary angiography interpretation and stenosis estimation," *npj Digit. Med.*, vol. 6, 2023, Art. no. 142, doi: 10.1038/s41746-023-00880-1.
[18] T. Iqbal, A. Khalid, and I. Ullah, "Explaining decisions of a light-weight deep neural network for real-time coronary artery disease classification in magnetic resonance imaging," *J. Real-Time Image Process.*, vol. 21, 2024, Art. no. 31, doi: 10.1007/s11554-023-01411-7.
[19] S. Bach, A. Binder, G. Montavon, F. Klauschen, K.-R. Müller, and W. Samek, "On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation," *PLoS ONE*, vol. 10, no. 7, 2015, Art. no. e0130140, doi: 10.1371/journal.pone.0130140.
[20] S. Asif *et al.*, "MultiResFF-Net: Multilevel residual block-based lightweight feature fused network with attention for gastrointestinal disease diagnosis," *Int. J. Intell. Syst.*, vol. 2025, 2025, Art. no. 1902285, doi: 10.1155/int/1902285.
[21] J.-S. Lee, "Digital image enhancement and noise filtering by use of local statistics," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-2, no. 2, pp. 165–168, 1980, doi: 10.1109/TPAMI.1980.4766994.
[22] J. Immerkær, "Fast noise variance estimation," *Comput. Vis. Image Underst.*, vol. 64, no. 2, pp. 300–302, 1996, doi: 10.1006/cviu.1996.0060.
[23] C. Rother, V. Kolmogorov, and A. Blake, "GrabCut: Interactive foreground extraction using iterated graph cuts," *ACM Trans. Graph.*, vol. 23, no. 3, pp. 309–314, 2004, doi: 10.1145/1015706.1015720.
[24] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE Trans. Syst., Man, Cybern.*, vol. 9, no. 1, pp. 62–66, 1979, doi: 10.1109/TSMC.1979.4310076.
[25] J. Fu, J. Liu, H. Tian, Y. Li, Y. Bao, Z. Fang, and H. Lu, "Dual attention network for scene segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2019, pp. 3146–3154, doi: 10.1109/CVPR.2019.00326.
[26] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He, "Aggregated residual transformations for deep neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2017, pp. 1492–1500, doi: 10.1109/CVPR.2017.634.
[27] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2017, pp. 4700–4708, doi: 10.1109/CVPR.2017.243.
[28] Z. Liu *et al.*, "Swin Transformer: Hierarchical vision transformer using shifted windows," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, 2021, pp. 10012–10022.
[29] Q. Hou, D. Zhou, and J. Feng, "Coordinate attention for efficient mobile network design," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2021, pp. 13713–13722.
[30] B. W. Matthews, "Comparison of the predicted and observed secondary structure of T4 phage lysozyme," *Biochim. Biophys. Acta—Protein Struct.*, vol. 405, no. 2, pp. 442–451, 1975, doi: 10.1016/0005-2795(75)90109-9.
[31] L. McInnes, J. Healy, and J. Melville, "UMAP: Uniform manifold approximation and projection for dimension reduction," 2018, *arXiv:1802.03426*.
[32] V. H. Jasodanand, S. S. Kowshik, S. Puducheri, M. F. Romano, L. Xu, R. Au, and V. B. Kolachalama, "AI-driven fusion of multimodal data for Alzheimer's disease biomarker assessment," *Nat. Commun.*, vol. 16, 2025, Art. no. 7407, doi: 10.1038/s41467-025-62590-4.
[33] E. Kalita, H. El Aouifi, A. Kukkar, S. Hussain, T. Ali, and S. Gaftandzhieva, "LSTM-SHAP based academic performance prediction for disabled learners in virtual learning environments: A statistical analysis approach," *Soc. Netw. Anal. Min.*, vol. 15, 2025, Art. no. 65, doi: 10.1007/s13278-025-01484-1.
[34] E. Srivastava, S. Mohan, T. K. Gandhi, A. K. Choudhury, and S. Kumar, "Attention-driven graph-based machine learning for non-invasive diagnosis of NAFLD," *Intell.-Based Med.*, vol. 12, 2025, Art. no. 100288, doi: 10.1016/j.ibmed.2025.100288.
[35] Y. Wang, Z. Wang, T. Wang, J. Hu, Z. You, and J. Peng, "HiGLDP: A hierarchical graph neural network for predicting lncRNA-disease associations through multi-omic integration," *BMC Biol.*, vol. 24, 2026, Art. no. 80, doi: 10.1186/s12915-026-02557-z.
[36] J.-H. Jang, Y.-Y. Jo, S. Kang, J. M. Son, H. S. Lee, J.-M. Kwon, and M. S. Lee, "A novel XAI framework for explainable AI-ECG using generative counterfactual XAI (GCX)," *Sci. Rep.*, vol. 15, 2025, Art. no. 23608, doi: 10.1038/s41598-025-08080-5.