



Journal of Smart Algorithms and Applications JSAA

ISSN: 3070-4189/© 2026, The Authors. (CC BY 4.0) License.

Journal Homepage

<https://pub.scientificirg.com/index.php/JSAA>



Trustworthy Coral Bleaching Classification with Convolutional and Transformer Networks

Asmaa A. Attwa^{a,1} , Ola Farid^b , Muthmainnah Muthmainnah^c , and Luis Cardoso^d

^a Faculty of Applied Health Sciences Technology, Beni-Suef University, Beni-Suef City, 62511, Egypt; E-mail: asmaa.abdelhamed@fahst.bsu.edu.eg

^b Computer Science Department, Faculty of Science, Beni-Suef University, Beni-Suef City, 62511, Egypt; E-mail: ola3131313@gmail.com

^c Universitas Al Asyriah Mandar, Indonesia; E-mail: muthmainnahunasman@gmail.com

^d Polytechnic Institute of Portalegre and Centre for Comparative Studies of the University of Lisbon, Portugal; E-mail: lmcardoso@ipportalegre.pt

ABSTRACT

Automated classification of healthy and bleached coral in underwater photographs can scale reef monitoring, but accuracy reported on public image collections is inflated when near-duplicate frames are shared between training and evaluation partitions. This study introduces a reusable leakage-mitigated, scene-aware benchmarking protocol and applies it to the Coral Reefs Images collection of 10,382 photographs. A single average hash flagged a same-class near-duplicate candidate in the original training partition for 418 of 720 native validation and test images (58.1%), against approximately 29% under a stricter multi-hash criterion. Scene grouping, group-stratified partitioning, and an embedding-based purge yielded 1,710 test images from 112 scene groups. Because the purge relocates individual images, the partitions are leakage-mitigated but not strictly scene-disjoint. Seven convolutional, transformer, and hybrid architectures, each tuned with 5 to 9 completed Optuna trials and trained with three seeds, reached mean test accuracy of 93.0% to 95.6% and ROC-AUC of 0.969 to 0.984. Scene-cluster bootstrap intervals for accuracy were 8 to 12 percentage points wide, and only one of 21 architecture pairs remained significant after joint Holm correction, whereas McNemar tests, which ignore scene clustering, were overconfident. The wide intervals indicate substantial uncertainty and limited evidence for distinguishing small differences, so the data do not support a reliable architecture ranking. A noise and blur augmentation arm, evaluated for four architectures, raised mean accuracy by 4.5 and 4.1 percentage points under strong noise and heavy blur (descriptive, untested) at a clean cost of 0.56 percentage points. Grad-CAM maps achieved lower deletion AUC and higher insertion AUC than smooth random maps for all models, which shows relative map utility and not biological faithfulness. Corner and caption occlusion changed few predictions, but matching dropout during training means this does not establish the absence of logo or caption shortcuts. An independent audit still found neighbors above 0.95 cosine similarity for 10.9% of test images, so the reported accuracies are within-collection estimates that may be optimistic for independently collected reef imagery.

PAPER INFORMATION

HISTORY

Received: 8 August 2026

Revised: 28 September 2026

Accepted: 16 October 2026

Online: 30 October 2026

MSC

68T07; 68T45; 68U10;

KEYWORDS

Scene-Aware;
Convolutional Neural
Networks;
Vision Transformers;
Coral Bleaching;
Explainability Analysis.

¹Corresponding author at Faculty of Applied Health Sciences Technology, Beni-Suef University, Beni-Suef City, 62511, Egypt; E-mail: asmaa.abdelhamed@fahst.bsu.edu.eg

1. INTRODUCTION

Coral reefs are diverse and economically important ecosystems, and marine heatwaves have turned mass bleaching from an occasional event into a recurrent stressor on reefs worldwide [1, 2, 3]. Bleaching is a stress response in which the coral host expels its algal symbionts, leaving pale tissue over the white skeleton [1]. Because the color change is visible in ordinary photographs, image-based screening is a practical complement to in-water surveys, particularly for monitoring programs that collect thousands of frames per site visit from divers, towed cameras, and autonomous vehicles.

Computer vision has been applied to reef imagery for more than a decade. Early systems paired color and texture descriptors with supervised classifiers [4, 5], and deep convolutional networks trained with augmentation and transfer learning later improved accuracy on coral texture benchmarks [6]. Recent studies address bleaching or coral health classification directly with local perception blocks [7], convolutional autoencoders [8], two-stream networks with post hoc explanations [9], and hybrid pipelines that combine transformer features with a YOLOv8 detector [10]. These studies rely on different image collections, partitioning rules, and metrics, so their reported accuracies are not directly comparable.

A further threat to valid comparison is overlap between training and evaluation images. Leakage was formalized in data mining as the use of information about the prediction target that should not legitimately be available at training time [11], and it has been documented across many scientific fields that use machine learning [12]. Near-duplicates between partitions inflate test accuracy in medical imaging [13] and occur in the CIFAR benchmarks [14]. Underwater collections are especially exposed, since frames exported from continuous video differ only by small camera motion or compression noise, and an automatic export has no reason to keep neighboring frames within one partition. In addition, networks can exploit unintended cues that correlate with labels [15], which is a concern for frames that carry channel logos or captions.

The public Coral Reefs Images collection [16] follows the Roboflow export convention, includes rotated and noise-augmented copies, and contains frames with channel logos and caption text. In this paper, a near-duplicate is defined operationally as an image whose perceptual-hash distance to another image falls below a fixed threshold, and a scene group is a set of images linked directly or transitively by such similarity. Scene groups approximate, but do not guarantee, disjoint reef scenes after the image-level purge. A first audit with a single average hash identified a same-class near-duplicate candidate in the original (native) training partition for 418 of the 720 native validation and test images. Accuracy on the native splits therefore provides limited evidence of generalization to unseen reef scenes, the practical question in ecological monitoring.

The contributions of this study are as follows.

- A reusable leakage-mitigated, scene-aware benchmarking protocol for image classification in clustered ecological image collections, comprising a multi-hash, flip-aware audit with a threshold sweep, an embedding-based scene graph, a similarity purge, and independent post-split audits with two networks and a baseline that uses no pixels.
- A scene-grouped benchmark of seven architectures, each allotted the same nominal Optuna budget and trained with three seeds, evaluated on 1,710 test images from 112 scene groups, with a second augmentation regime evaluated for four of the architectures.
- Scene-aware statistics, comprising cluster bootstrap intervals, a two-sided cluster sign-flip test for paired accuracy, and joint Holm correction, together with a direct comparison against the McNemar test.
- A quantitative explainability study based on deletion and insertion curves, occlusion agreement, parameter randomization, and saliency mass in corners, caption bands, and black borders, complemented by a 17-condition corruption sweep and an efficiency profile measured on one GPU.

The principal finding is that high apparent accuracy does not translate into a reliable architecture ranking once scene-level dependence is modeled. The protocol mitigates but does not eliminate leakage, and the corruption and explainability analyses are secondary evaluations of robustness and interpretability, not definitive tests of biological reliance.

The remainder of this paper is organized as follows. Section 2 reviews related work. Section 3 describes the data, leakage control, models, training, evaluation, and explainability methods. Section 4 reports the results, Section 5 discusses implications, limitations, and future work, and Section 6 concludes.

2. RELATED WORK

Automated annotation of reef survey images began with color and texture descriptors and supervised classifiers [4], which were later benchmarked against the variability among human experts [5]. Fine-tuned convolutional networks with data augmentation then improved accuracy on coral texture benchmarks [6]. For the healthy versus bleached task, ML-Net uses parallel local convolutions to preserve color and texture detail and reports 86.35% accuracy on the Bleached and Unbleached Corals Classification dataset [7]. Other proposals include a deep convolutional autoencoder for bleaching detection [8], a

two-stream network with Grad-CAM and LIME explanations for coral reef health classification [9], and a pipeline that combines CNN and transformer features with YOLOv8 for coral image classification [10]. A larger Indo-Pacific dataset of more than 20,000 photo transect images supports multilabel condition and stressor classification, for which an ensemble method was proposed after seven architectures were tested [17]. **Table 1** summarizes these studies together with the leakage literature.

Convolutional networks remain a standard choice for image classification [18], while vision transformers replace convolution with global self-attention over image patches [19, 20, 21]. Analyses of internal representations indicate that transformers aggregate global context earlier than convolutional networks [22], and ensembles that combine both families have been studied for plankton imagery [23]. Modernized convolutional designs [24] and hybrid attention designs [25] narrow the gap between the two families. Pretrained weights and transfer learning remain standard where collections are small relative to ImageNet [26, 27, 28], and fine-tuning schedules that first fit a linear head can preserve pretrained features [29].

Evaluation validity is a separate concern. Leakage in imaging typically takes the form of near-duplicate samples across partitions, which inflates accuracy [13] and affects even canonical benchmarks [14]. Perceptual hashing maps visually similar images to nearby binary codes and is a standard low-cost detector [30, 31]. Hash distances alone miss rotated, cropped, or recolored copies, which motivates complementary embedding-based checks. Statistical comparison of classifiers on a single test set is commonly based on the McNemar test [32, 33], which treats test images as independent. When images cluster into scenes, resampling at the level of clusters is the standard remedy for interval estimation [34, 35].

Explanations from gradient-based maps [36] and from transformer-specific attributions [37, 38] are often presented qualitatively. Sanity checks that randomize model parameters [39] and perturbation-based deletion and insertion metrics [40] provide quantitative tests. Robustness to image corruptions has been benchmarked systematically for natural images [41], whereas underwater imagery adds attenuation, scattering, and color loss [42]. None of the coral studies summarized in **Table 1** combines scene-grouped partitioning, multi-seed training, scene-aware statistics, quantitative explanation tests, and a corruption sweep in one protocol.

Table 1: Related Work on Coral Image Analysis and Evaluation Leakage

Study	Year	Task and data	Method	Limitation
Beijbom et al. [4, 5]	2012, 2015	Coral reef survey image annotation	Color and texture descriptors with SVM classifiers	Handcrafted features; no deep transfer learning
Gómez-Ríos et al. [6]	2019	Coral texture classification (EILAT, RSMAS)	ResNet, Inception, and DenseNet variants with augmentation and transfer learning	Species texture patches, not bleaching state; two small datasets
Wang et al. [7]	2024	Healthy versus bleached coral	ML-Net with multi-local perception blocks	Reports 86.35% accuracy on its own dataset; not comparable with other collections or split protocols
Gautam et al. [8]	2024	Bleaching detection	Deep convolutional autoencoder model	Results reported on its own data; not comparable with other collections or split protocols
Shao et al. [17]	2024	Multilabel reef condition and stressors, over 20,000 images	Seven architectures tested, ensemble method proposed	Multilabel task; F1 and match ratio are not comparable with binary accuracy
Ghaffar and Choi [9]	2025	Coral reef health classification	Two-stream CNN with Grad-CAM and LIME	Accuracy reported on its own data; not comparable with other collections or split protocols
Bhanu and Saravanan [10]	2026	Coral image classification	CNN and transformer features with YOLOv8	Accuracy reported on its own data; not comparable with other collections or split protocols
Tampu et al. [13]	2022	OCT image classification	Correct versus leaky partitioning compared	Not coral imagery; demonstrates accuracy inflation by leakage
Barz and Denzler [14]	2020	CIFAR-10 and CIFAR-100	Near-duplicate detection and replacement (ciFAIR)	Not coral imagery; duplicates in 3.3% and 10% of test images, with a 9% to 14% relative accuracy drop on the duplicate-free test set

3. MATERIALS AND METHODS

3.1 Pipeline Overview

The pipeline in **Figure 1** has seven stages. Images are decoded once into a uint8 cache (Stage 1). A multi-hash audit measures cross-partition overlap in the native splits, scene groups are built from hash, source-frame, and embedding edges to

form group-stratified partitions, and the partitions are purged of residual neighbors and audited with ResNet-50 and with an independent ConvNeXt-T (Stage 2). Training augmentation (Stage 3), seven architectures (Stage 4), and tuning and training with three seeds under two augmentation arms (Stage 5) follow. The test partition is then evaluated with ensembles and scene-aware statistics (Stage 6), and corruption sweeps, explanation tests, and efficiency profiling complete the evaluation (Stage 7). Three partition terms are used throughout. The native split is the original Roboflow train, validation, and test folders. The grouped split is the output of group-stratified partitioning. The purged split is the grouped split after the embedding-based purge.

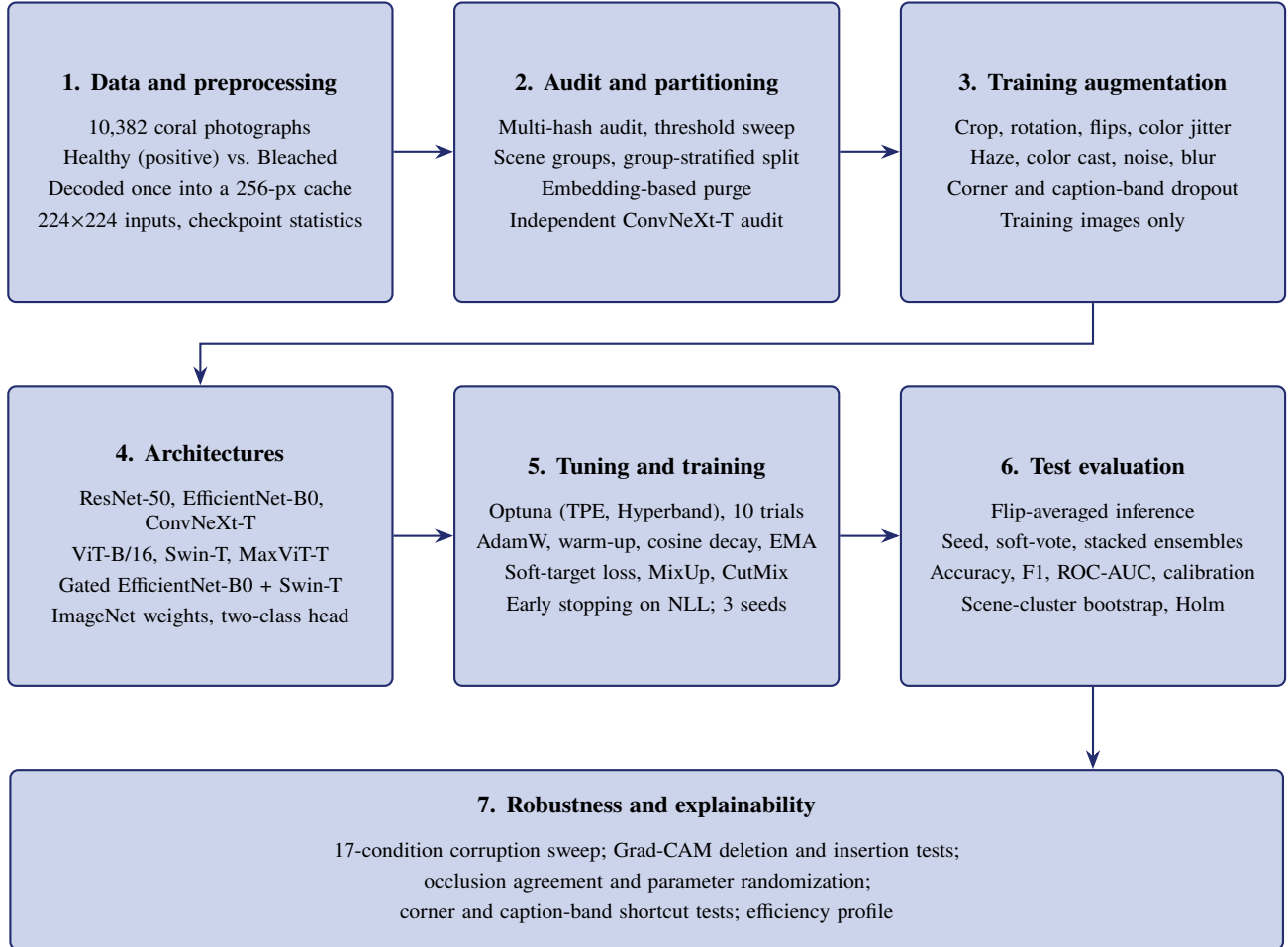


Figure 1: Leakage-Mitigated Benchmarking Pipeline

3.2 Dataset

The Coral Reefs Images collection [16] contains 10,382 photographs with image-level labels, Bleached and Healthy, and native train, validation, and test folders of 9,662, 463, and 257 images. The native training folder holds 4,980 Bleached and 4,682 Healthy images, the validation folder 240 and 223, and the test folder 135 and 122. The data come from the Kaggle distribution of the collection. Healthy is the positive class in all metrics, so sensitivity is Healthy recall and specificity is Bleached recall. **Figure 2** shows representative images. Black rotation borders and speckle noise are consistent with an offline augmentation step in the original export, many frames contain channel logos or caption text, and several scenes include divers, fish, or an anemone. Labels describe whole scenes, not isolated colonies.

3.3 Leakage Audit and Scene Grouping

Each image is reduced to a 32 by 32 grayscale thumbnail, from which three 64-bit perceptual hashes [30, 31] are computed: an average hash (aHash), obtained by thresholding an 8 by 8 resampled thumbnail at its mean; a difference hash (dHash), obtained from horizontal gradient signs of a 9 by 8 thumbnail; and a DCT-based hash (pHash), obtained by thresholding the 8 by 8 low-frequency block of the discrete cosine transform at its median without the DC term. Hashes are also computed for the horizontal flip, vertical flip, and 180-degree rotation of each image, because the export contains flipped copies. As shown in **Equation 1**, the distance between images x and y is the minimum over the geometric variants of the second image

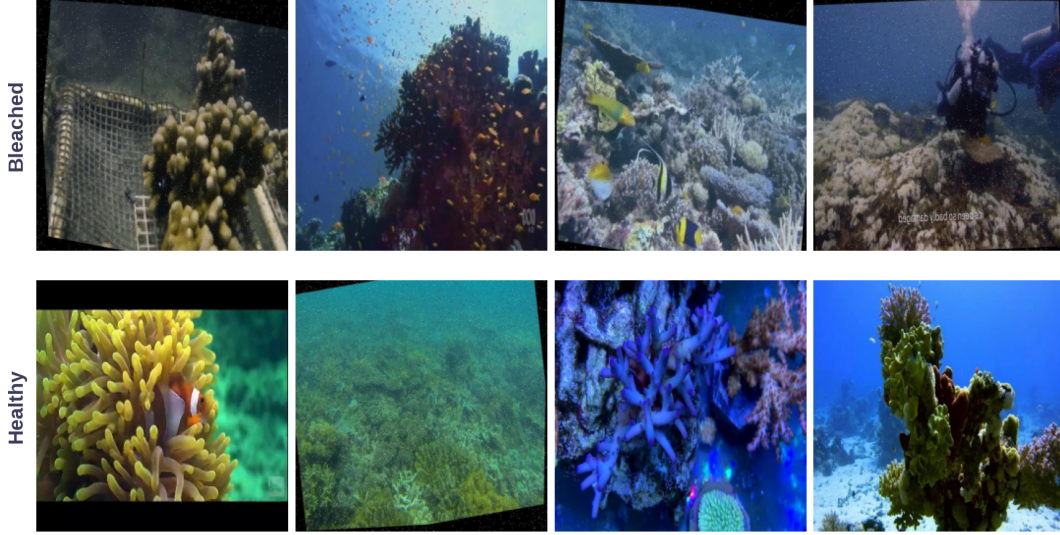


Figure 2: Representative Bleached and Healthy Images

of the mean Hamming distance over the three hashes:

$$d(x, y) = \min_{v \in \mathcal{V}} \frac{1}{3} \sum_{h \in \{a, d, p\}} d_H(h(x), h(v(y))), \quad (1)$$

where $h(x)$ is the 64-bit hash of image x , d_H is the Hamming distance in bits, and $\mathcal{V}$ contains the identity, horizontal flip, vertical flip, and 180-degree rotation. The audit considers all cross-partition image pairs, namely train to validation, train to test, and validation to test. For comparison, a single aHash without flip variants is evaluated at $\tau = 6$, and the threshold is swept over $\tau \in \{0, 2, \dots, 12\}$. Three thresholds play distinct roles: the hash threshold τ , which flags candidates and builds scene groups; the cosine threshold ρ of the embedding edges; and the purge threshold of 0.95 applied to evaluation images. The hash threshold, the embedding threshold, and the purge threshold are operational choices that were not validated against manually annotated duplicate pairs, so the audit is an operational screen and not an exhaustive duplicate census. The value $\tau = 6$ was selected for scene grouping because it preserved a non-degenerate group structure while capturing visually near-identical frames.

Scene groups are the connected components of a graph on all images with three edge types. Hash edges link images with $d \leq \tau$ and $\tau = 6$. Source-frame edges link images with an equal Roboflow source-frame key, which is the file stem before the `.rf.` export hash. Embedding edges link each image to its $k = 20$ nearest neighbors whose similarity reaches a threshold ρ . Embeddings are the L2-normalized pooled features of an ImageNet-pretrained EfficientNet-B0, computed for each image and for its horizontal flip. As shown in **Equation 2**, the embedding similarity between images x and y is the larger of the cosine similarities to the image and to its horizontal flip:

$$s_c(x, y) = \max \{f(x)^\top f(y), f(x)^\top f(\eta(y))\}, \quad (2)$$

where $f(\cdot)$ is the L2-normalized embedding of the network used in the respective stage and η is the horizontal flip. The threshold ρ is chosen automatically as the lowest value in $\{0.90, 0.92, 0.94, 0.95, 0.96, 0.97, 0.98\}$ for which the largest group is at most 25% of the data, which prevents chaining into a single giant component.

3.4 Partitioning and Purge

Groups are assigned to partitions with a shuffled stratified group k -fold split with $k = 7$ and seed 2024 [43]. Fold 0 serves as the test partition, fold 1 as the validation partition, and the remaining five folds as the training partition, which gives nominal proportions of 5/7, 1/7, and 1/7 with balanced classes. A flip-aware purge then moves validation and test images whose ResNet-50 embedding similarity s_c , defined in **Equation 2**, is at least 0.95 to any training image into the training partition, and test images similar to validation images into the validation partition. The purge repeats until no pair remains, with at most 25 passes, as summarized in **Algorithm 1**. The purge uses image content only, and labels and predictions play no role.

3.5 Partitioning Assumptions and Independent Audit

Because the purge relocates individual images rather than whole groups, some groups end up in more than one partition. The final partitions are near-duplicate-aware but not strictly group-disjoint, and the reported results reflect a leakage-mitigated, not scene-disjoint, protocol. Moving images instead of discarding them keeps every image in use, at the cost of reduced independence of the evaluation partitions and a shift in class prior. The threshold of 0.95 is a fixed operating point that was not validated against annotated pairs, and the independent audit below shows that it does not remove all neighbors.

Algorithm 1 Scene-Grouped Partitioning with Similarity Purge

Input: image set $\mathcal{I}$; hash threshold τ ; cosine grid $\mathcal{P}$; neighbor count k ; fold count K ; purge threshold ρ_a ; seed

Output: partitions $\mathcal{T}, \mathcal{V}, \mathcal{S}$ and scene labels $g(\cdot)$

```

1:  $E_h \leftarrow \{(x, y) : d(x, y) \leq \tau\}$  ▷ hash edges, Equation 1
2:  $E_s \leftarrow \{(x, y) : x \text{ and } y \text{ share a source-frame key}\}$ 
3: for all  $\rho \in \mathcal{P}$  in ascending order do
4:    $E_e(\rho) \leftarrow \{(x, y) : y \in \text{kNN}_k(x), s_c(x, y) \geq \rho\}$  ▷ embedding edges, Equation 2
5:    $g \leftarrow$  connected components of  $(\mathcal{I}, E_h \cup E_s \cup E_e(\rho))$ 
6:   if the largest component contains at most  $0.25 |\mathcal{I}|$  images then
7:     break
8:   end if
9: end for
10:  $(\mathcal{T}, \mathcal{V}, \mathcal{S}) \leftarrow$  stratified group  $K$ -fold split of  $g$  ▷ fold 0 test, fold 1 validation, remainder training
11: repeat
12:    $M_1 \leftarrow \{x \in \mathcal{V} \cup \mathcal{S} : \max_{t \in \mathcal{T}} s_c(x, t) \geq \rho_a\}$ ; move  $M_1$  to  $\mathcal{T}$ 
13:    $M_2 \leftarrow \{x \in \mathcal{S} : \max_{v \in \mathcal{V}} s_c(x, v) \geq \rho_a\}$ ; move  $M_2$  to  $\mathcal{V}$ 
14: until  $M_1 \cup M_2 = \emptyset$  or 25 passes are completed
15: audit  $(\mathcal{T}, \mathcal{V}, \mathcal{S})$  with an independent network
16: return  $\mathcal{T}, \mathcal{V}, \mathcal{S}, g$ 

```

The partitioned data are audited with two networks. ResNet-50 is the network used for the purge, so the absence of neighbors above its threshold is guaranteed by construction and is not an independent audit. ResNet-50 constructs the purge and therefore cannot check it independently, whereas ConvNeXt-T was not used in any earlier step and serves as the independent audit. For each evaluation image, the nearest training neighbor is found by plain cosine similarity without flips, and the fractions above 0.90, 0.95, and 0.98 are reported. A shortcut baseline trains a histogram gradient-boosting classifier [43] (depth 3, 150 iterations, learning rate 0.08) on handcrafted features, without access to pixel values, on two feature sets under five-fold group cross-validation: file metadata (original height, width, aspect ratio, black border fraction) and global color and texture statistics (luminance, contrast, dark channel [44], saturation, noise level, Laplacian variance, mean RGB). Original width and aspect ratio are constant over the collection, so the metadata set effectively reduces to original height and black border fraction.

3.6 Architectures

Seven architectures are compared in **Table 2**. All are instantiated through the timm library [45] with ImageNet pretrained weights [26] and a two-class head: ResNet-50 [46]; EfficientNet-B0 [47], which uses mobile inverted bottleneck blocks [48] with squeeze-and-excitation [49]; ViT-B/16 [19], which is built on self-attention [50]; Swin-T [51]; ConvNeXt-T [24]; MaxViT-T [25]; and a fusion network. In the fusion network, the pooled features of EfficientNet-B0 and Swin-T are layer-normalized and projected to 512 dimensions, giving f_a and f_b . As shown in **Equation 3**, the two projections are merged by a learned sigmoid gate computed from the concatenation $[f_a; f_b]$:

$$f = g \odot f_a + (1 - g) \odot f_b, \quad g = \sigma(W_g [f_a; f_b] + b_g) \in (0, 1)^{512}, \quad (3)$$

where σ is the logistic function, W_g and b_g are learned parameters, and $\odot$ denotes the elementwise product. The fused vector f is passed to a dropout layer and a linear head. Head weights are initialized so that initial logits are near zero for every family. Inputs are 224 by 224 pixels and are normalized with the statistics of the corresponding pretrained checkpoint.

3.7 Training Protocol

Training uses AdamW [52] with decoupled weight decay, a one-epoch linear warmup followed by cosine decay without restarts [53], a batch size of 64, gradient norm clipping at 1.0 [54], and mixed precision with fp16 and loss scaling [55].

Table 2: Architectures and ImageNet Checkpoints

Model	timm identifier	Family
ResNet-50	resnet50	Convolutional
EfficientNet-B0	efficientnet_b0	Convolutional
ConvNeXt-T	convnext_tiny	Convolutional
ViT-B/16	vit_base_patch16_224	Transformer
Swin-T	swin_tiny_patch4_window7_224	Transformer
MaxViT-T	maxvit_tiny_tf_224	Hybrid
Fusion	EfficientNet-B0 + Swin-T, gated	Hybrid

Training runs for at most 12 epochs with early stopping on validation negative log-likelihood (NLL) and a patience of four epochs. The model weights are tracked by an exponential moving average. As shown in **Equation 4** [56, 57],

$$\bar{\theta}_t = \delta_t \bar{\theta}_{t-1} + (1 - \delta_t) \theta_t, \quad \delta_t = \min\left(\delta, \frac{1+t}{10+t}\right), \quad (4)$$

where θ_t are the weights after update t and $\delta = 0.998$. The same rule is applied to floating-point buffers, which include the batch normalization statistics [58], and the averaged weights are evaluated. The warm-up term reaches 0.998 only after about 4,500 updates, whereas a 12-epoch run comprises at most 1,428 updates, so the warm-up term governs the whole run. At the end of a run the decay is about 0.994, which corresponds to an averaging horizon of roughly 160 updates, or about 1.3 epochs. Fine-tuning optionally begins with a linear-probe epoch in which the backbone and its batch normalization statistics are frozen [29]. ViT-B/16, Swin-T, ConvNeXt-T, MaxViT-T, and the fusion network use layerwise learning rate decay. If a batch exceeds GPU memory, it is split into micro-batches with gradient accumulation, which keeps the effective batch size and the gradient unchanged. Batch normalization statistics are still computed within each micro-batch, which differs from full-batch statistics for the batch-normalized networks (ResNet-50 and EfficientNet-B0). Inference averages the logits of the image and its horizontal flip.

The loss uses inverse-frequency class weights, label smoothing [59, 60] with factor ϵ , and soft targets t_{ic} produced by MixUp [61] or CutMix [62]. Let n_c be the number of training images of class c and $C = 2$ the number of classes. As shown in **Equation 5**, the class weights are

$$w_c = \frac{C/n_c}{\sum_{c'=1}^C 1/n_{c'}}, \quad (5)$$

which have a mean of one over the classes. For a batch of N images with targets t_{ic} (one-hot for unmixed images) and softmax probabilities p_{ic} , the smoothed targets are computed as shown in **Equation 6** [59]:

$$q_{ic} = (1 - \epsilon) t_{ic} + \frac{\epsilon}{C}, \quad (6)$$

and the corresponding loss is defined as shown in **Equation 7**:

$$\mathcal{L} = - \frac{\sum_{i=1}^N \sum_{c=1}^C w_c q_{ic} \log p_{ic}}{\sum_{i=1}^N \sum_{c=1}^C w_c t_{ic}}. \quad (7)$$

For hard labels, **Equation 7** is identical to the weighted cross-entropy with label smoothing of PyTorch [63], which a unit test verified. When gradients are accumulated, the denominator is the weight mass of the whole batch.

Hyperparameters are tuned for each model with Optuna [64], using a multivariate tree-structured Parzen estimator [65] and a Hyperband pruner [66]. Each study is allotted 10 trials on a fixed random 30% subset of the training images, with three epochs per trial and a fixed validation subset of at most 1,024 images that is identical for all trials of a model. Both subsets are drawn at image level, so the subsampling cannot create additional overlap with the validation data beyond the residual overlap described above. The objective is validation NLL. The search space comprises the learning rate on a logarithmic scale (2e-4 to 2e-3 for ResNet-50 and EfficientNet-B0, and 5e-5 to 6e-4 for the other five models), weight decay (1e-3 to 0.2, logarithmic), classifier dropout (0 to 0.5), stochastic depth [67] (0 to 0.3), label smoothing (0 to 0.15), MixUp $\alpha \in \{0, 0.2, 0.4\}$, CutMix probability (0 to 0.5), layer decay (0.6 to 0.95, for the five models with layerwise decay), and zero or one linear-probe epoch. The budget is equal only in nominal terms: every model was allotted 10 trials, but only 5 to 9 trials completed per model because the others were pruned or failed (Section 4.2), and wall-clock time was not equalized. The benchmark therefore compares architectures under a comparable, not identical, search effort. The test partition is not touched during tuning, early stopping, or ensembling. The sampler seed is derived from the split seed and the model name, and final training uses seeds 0, 1, and 2.

3.8 Augmentation Arms and Corruption Sweep

All augmentations are performed on the GPU using a cache of uint8 images resized to 256 pixels, as illustrated in **Figure 3**. Both arms apply a random resized crop (area 0.6 to 1.0, aspect ratio 0.9 to 1.1), rotation by up to 20 degrees with probability 0.5, horizontal flip (0.5), vertical flip (0.2), brightness, contrast, and saturation jitter of 25% (probability 0.7), an underwater haze and color cast model (probability 0.3), and random erasing of a square covering 25% of the side length (probability 0.3). Haze blends the image with a blue-green veil and attenuates the red channel by up to 15% [42, 44]. The basic arm adds Gaussian noise with probability 0.15 (maximum standard deviation 0.04) and blur with probability 0.15 (σ from 0.4 to 0.8 pixels). The robust arm adds Gaussian, signal-dependent shot, or multiplicative speckle noise with probability 0.45 (maximum 0.10), blur with probability 0.40 (σ from 0.3 to 2.5), corner dropout (a square of side 10% to 25% of the image side at one of the four corners, probability 0.35), and caption band dropout (a full-width band of height 10% to 22%, placed at the bottom with probability 0.7 and at the top otherwise, probability 0.25). Dropped regions are filled with the image mean. The corner and band operations are designed to break logo and caption shortcuts [15]. Contrast enhancement is not applied: an ablation of contrast-limited adaptive histogram equalization (CLAHE) [68, 69] in an earlier version of this pipeline, which used a less strict split, showed no benefit, with every paired difference at or below zero. Illumination and turbidity variation is instead represented by the haze, brightness, and color augmentation and by the corruption sweep.

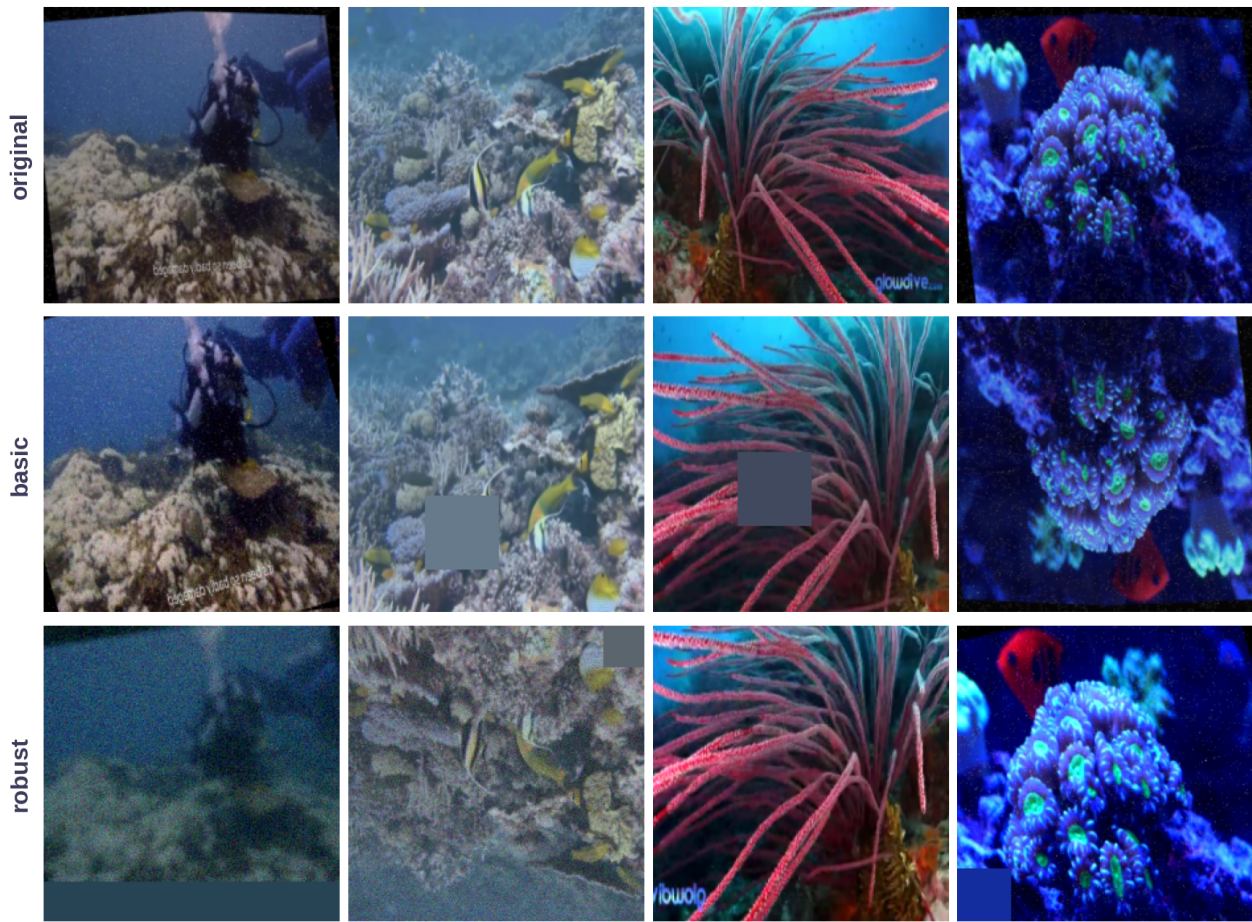


Figure 3: Original, Basic, and Robust Augmentation of Four Training Images

The basic arm is trained for the four baseline architectures (ResNet-50, EfficientNet-B0, ViT-B/16, Swin-T) and the robust arm for all seven, giving 11 configurations and 33 training runs. Only these four architectures have both arms, so the augmentation comparison is restricted to them and is not a balanced ablation over all seven. At test time, 17 conditions, namely the clean images and 16 corruptions, are applied to a fixed subset of at most 600 test images using deterministic seeds, so that every model receives identical corrupted images. **Figure 4** shows representative conditions. The corruptions are Gaussian noise at standard deviations 0.02, 0.05, 0.10, and 0.15; speckle noise at 0.10; shot noise at 0.10; JPEG compression at quality 30; Gaussian blur at σ of 0.5, 1, 2, and 3 pixels; darkening and brightening by 40%; haze at 0.35 and 0.60; and red loss at 0.40. The design follows common-corruption benchmarking [41] with underwater-specific additions. Noise and blur are perturbation families on which the robust arm was explicitly trained, whereas JPEG compression and red loss were not used in training, and brightness changes and haze occur in both arms. Corruption results are therefore

interpreted separately for these two groups, and all contrasts between the arms are descriptive and untested.

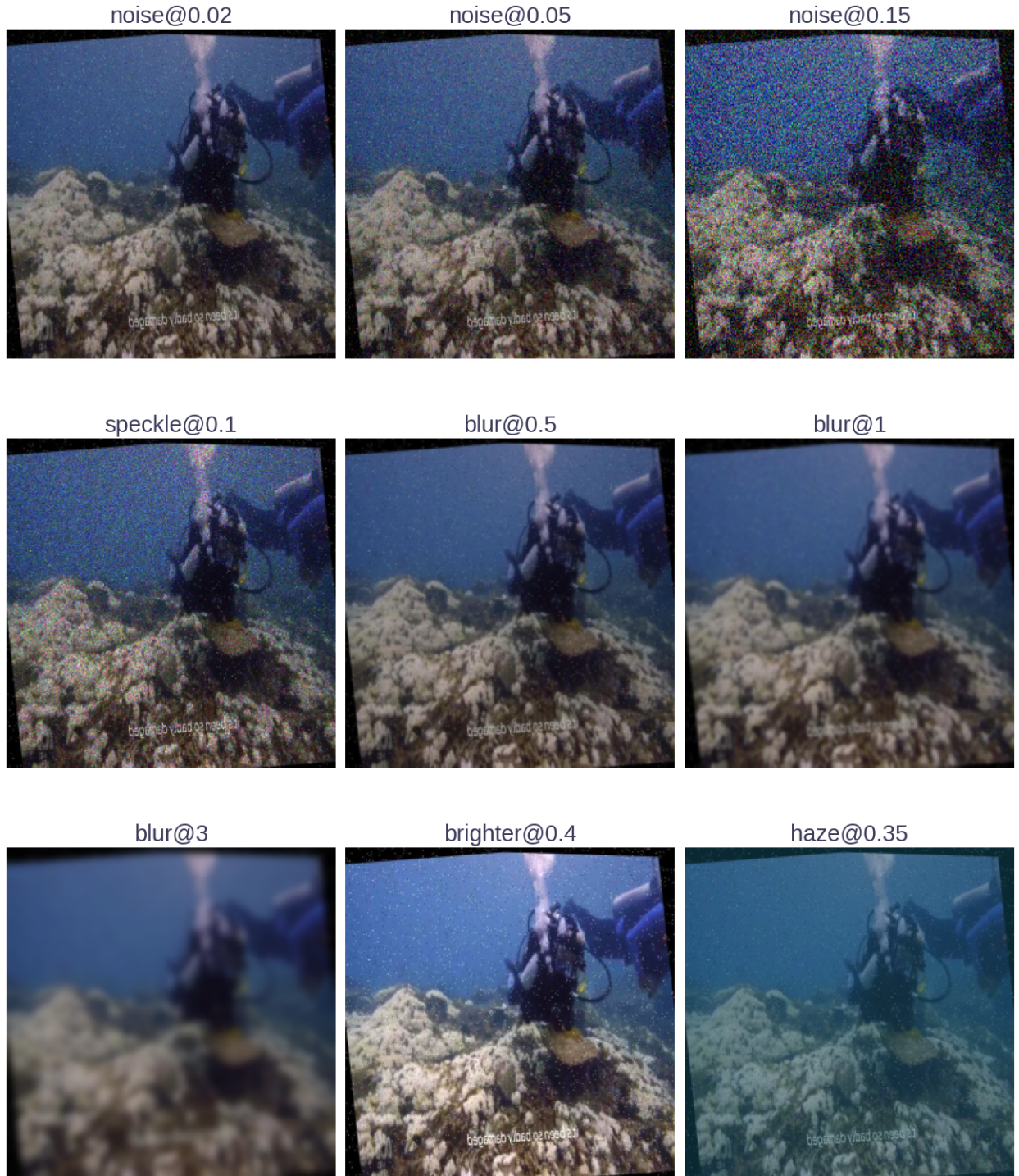


Figure 4: Examples of Test-Time Corruption Conditions Applied to One Test Image

3.9 Evaluation Metrics and Statistics

Reported metrics are accuracy, Healthy F1, ROC-AUC [70], average precision [71], sensitivity, specificity, balanced accuracy (the mean of sensitivity and specificity) [72], log-loss, and the expected calibration error (ECE) [73]. Calibration is assessed on the Healthy probability p_i with $M = 10$ equal-width bins B_m , as shown in **Equation 8** [73]:

$$\text{ECE} = \sum_{m=1}^M \frac{|B_m|}{N} \left| \frac{1}{|B_m|} \sum_{i \in B_m} y_i - \frac{1}{|B_m|} \sum_{i \in B_m} p_i \right|, \quad (8)$$

where $y_i \in \{0, 1\}$ is the Healthy label and N is the number of evaluated images. Ensembles are the average of seed-wise probabilities for one model, a soft vote across all seven models, and a logistic-regression stack fitted on validation logits [74]. Results are reported as mean and standard deviation over the three seeds and, for seed-averaged ensembles, as a single set of test predictions.

Because test images from one scene are highly correlated, uncertainty is estimated by a cluster bootstrap [75, 34, 35]: scene groups are resampled with replacement, all images of the selected groups are included, and the statistic is recomputed over $B = 2000$ replicates, from which percentile intervals are taken. Paired differences between two classifiers use the same resampled groups, and the bootstrap p-value is twice the smaller tail fraction of the paired difference. Paired accuracy is also tested with a cluster sign-flip test, which is a permutation test [76]. Let c_i^A and c_i^B indicate correct predictions of classifiers A and B for image i , and let $D_g = \sum_{i \in g} (c_i^A - c_i^B)$ for scene group g . The observed statistic is $T_{\text{obs}} = |\sum_g D_g|$. If the two classifiers are exchangeable within each scene group, the sign of every D_g is equally likely to be positive or negative, so the null replicates are generated as shown in **Equation 9**:

$$T^{(b)} = \left| \sum_g s_g^{(b)} D_g \right|, \quad s_g^{(b)} \in \{-1, +1\} \text{ with equal probability,} \quad (9)$$

for $b = 1, \dots, B'$ with $B' = 20,000$, and the p-value is computed as shown in **Equation 10**, which includes the correction that prevents zero p-values for randomly drawn permutations [77]:

$$p = \frac{1 + \#\{b : T^{(b)} \geq T_{\text{obs}}\}}{B' + 1}. \quad (10)$$

Because both the observed and the null statistics are absolute values, the test is two-sided. Groups with $D_g = 0$ do not contribute to **Equation 9** and **Equation 10**. The exact binomial form of the McNemar test [32] is computed for comparison but is not used for inference, since it treats images as independent. For the 21 architecture pairs, significance is judged from the cluster bootstrap p-values for accuracy and for ROC-AUC, which are corrected jointly (42 tests) with the Holm procedure [78]; the sign-flip and McNemar p-values for accuracy are reported alongside. Effects of augmentation on clean test images are estimated per model from three paired seeds with t intervals, and the McNemar and sign-flip p-values are Holm corrected across the four models. The corruption-sweep contrasts between the two arms are reported as mean differences only. No interval or significance test was computed for them, and the contrasts are interpreted as descriptive.

3.10 Explainability Methods

Grad-CAM [36] is computed on the signed logit margin $s_c = z_c - z_{\bar{c}}$ between the predicted class c and the other class $\bar{c}$, which avoids blank maps that arise from raw logits in a two-class softmax. Let A_{uv}^k be the activation of channel k of the target layer at location (u, v) , with spatial size $h \times w$. As shown in **Equation 11**, the channel weights and the map are

$$\alpha_k = \frac{1}{hw} \sum_{u,v} \frac{\partial s_c}{\partial A_{uv}^k}, \quad L_{uv} = \text{ReLU}\left(\sum_k \alpha_k A_{uv}^k\right), \quad (11)$$

and the map L is upsampled bilinearly and scaled to $[0, 1]$. Target layers are the last stage of convolutional and hybrid networks, the token normalization of the last block reshaped to a grid for ViT-B/16 and Swin-T [38], and both branch layers for the fusion network, whose maps are averaged. All models are treated identically, and attention rollout [37] is not used.

A fixed set of 48 randomly drawn, class-balanced test images from 44 distinct scene groups (24 Healthy) is explained for every model and every seed. Stored checkpoints reproduce the evaluated logits on the first 64 test images (mean maximum absolute logit difference of at most 0.003 and full label agreement). Five quantities are computed. Deletion and insertion curves [40] use 12 steps and replace pixels by a blurred copy (window size 21). The area under the predicted-class probability curve is compared with that of smooth random maps (mean of three draws), where lower is better for deletion and higher for insertion. Agreement with model-agnostic occlusion sensitivity [79] is the Spearman correlation between the map and the mean probability drop caused by 48-pixel patches with stride 24, evaluated on 16 images. Parameter randomization [39] correlates maps of the trained model with maps after re-initializing the head, the upper 50% of layers, and all layers, on eight images. Seed consistency is the mean Spearman correlation between maps of independently trained seeds. Let $S(p) \in [0, 1]$ be the saliency map at pixel p . As shown in **Equation 12**, the saliency mass ratio for a pixel region Ω is

$$R_\Omega = \frac{\sum_{p \in \Omega} S(p) / \sum_p S(p)}{|\Omega| / |\text{image}|}, \quad (12)$$

where $|\Omega|$ is the number of pixels in the region, so that $R_\Omega = 1$ indicates no preference. Regions are the four corner squares (22% of the side), the top and bottom bands (20% of the height), and black rotation borders (images with more than 2% black pixels). Corner and band occlusion tests replace the region by the image mean and record the fraction of flipped predictions and the accuracy change on all test images. Embeddings are the penultimate features of seed 0 for all test images; the silhouette coefficient [80] of the two classes is computed in the full feature space, and a two-component PCA is used only for display.

3.11 Efficiency Profiling and Implementation

Parameters and floating-point operations (two operations per multiply-accumulate) are computed for a 224 by 224 input. Latency is the median and 95th percentile of batch-size-one forward passes, throughput uses batches of 32, and peak memory is recorded, all under mixed precision on the same GPU. Preprocessing, training, and inference times are measured separately, together with the wall-clock time of each stage.

Experiments ran on a Kaggle notebook with a Tesla T4 GPU (14.56 GB, compute capability 7.5), four CPU cores, and 31.3 GB RAM, using Python 3.13.15, PyTorch 2.11.0 [63] with CUDA 12.8 and cuDNN 9.19, timm 1.0.29, NumPy 2.1.3, pandas 2.3.3, OpenCV 4.14.0, scikit-learn 1.6.1, SciPy 1.16.3, and Optuna 5.0.0. A suite of 49 unit tests covering the loss, augmentation, grouping, bootstrap, sign-flip test, and explanation code passed before the run. The full run took 320.1 minutes.

4. RESULTS

4.1 Leakage Audit and Partitions

On the native splits, 418 of 720 validation and test images (58.1%) have a same-class near-duplicate in the training partition under a single average hash with $\tau = 6$, and 40 of the 257 native test images have a near-duplicate in the validation partition. Under the multi-hash criterion at $\tau = 6$ there are 165 cross-class near-duplicate pairs, each of which is either a label conflict or a visually ambiguous pair of images. The contamination rate rises monotonically with the threshold, as shown in **Figure 5**. The single hash flags about twice as many images as the stricter multi-hash criterion at moderate thresholds, from about 58% against 29% at $\tau = 6$ to about 90% against 60% at $\tau = 12$. The difference arises because the multi-hash distance averages three hashes, so a pair that is close under aHash but distant under dHash or pHash is not flagged.

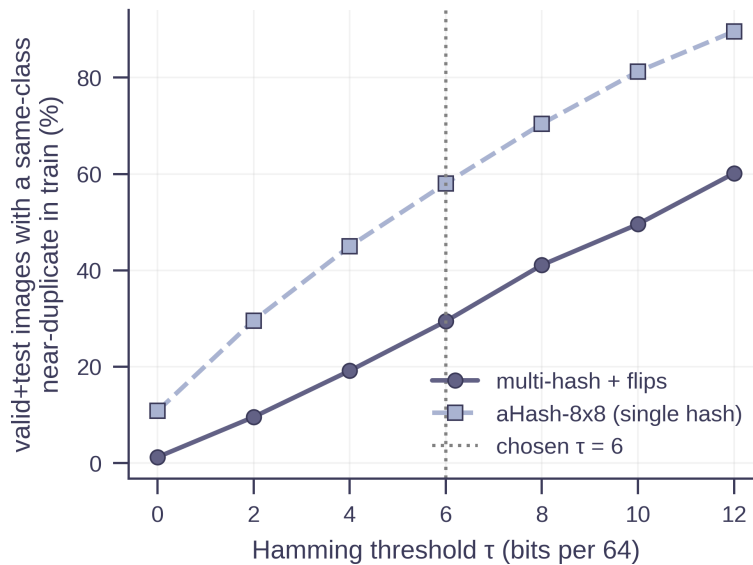


Figure 5: Same-Class Near-Duplicate Rate Against Hamming Threshold

Group structure is sensitive to the thresholds, as shown in **Table 3**. With hash edges only, the largest group stays below 17% of the data up to $\tau = 10$ and merges into one component of 51% at $\tau = 12$, so $\tau = 6$ is retained. Adding embedding edges reduces the group count markedly, and the lowest cosine threshold with a largest group of at most 25% is 0.95, which yields 669 scene groups, the largest with 2,490 images (24.0%), and 146 singletons. The largest group alone holds almost a quarter of the collection.

The grouped split initially yielded 7,267 training, 1,149 validation, and 1,966 test images. After the purge moved 99 validation images and 239 test images to training and 17 test images to validation, the final partitions contain 7,605 training, 1,067 validation, and 1,710 test images, with 515, 103, and 112 scene groups and Healthy fractions of 0.498, 0.494, and 0.415. The group counts sum to 730, which exceeds 669 because the purge moved images, so some groups appear in more than one partition. The test partition has fewer Healthy images (709) than Bleached images (1,001), a prior shift relative to validation that matters for the stacked ensemble below.

The post-split audit is summarized in **Table 4**, where Median is the median cosine similarity between an evaluation image

Table 3: Scene-Group Structure at Different Similarity Thresholds

Edge setting	Largest group		Share (%)	Singletons
	Groups	Images		
<i>Hash-only grouping</i>				
Hash $\tau = 4$	1,145	530	5.1	255
Hash $\tau = 6$	956	1,322	12.7	210
Hash $\tau = 8$	746	1,651	15.9	158
Hash $\tau = 10$	597	1,701	16.4	129
Hash $\tau = 12$	450	5,293	51.0	98
<i>Hash and embedding grouping</i>				
$\tau = 6, \rho = 0.90$	334	4,971	47.9	70
$\tau = 6, \rho = 0.94$	583	2,979	28.7	124
$\tau = 6, \rho = 0.95$	669	2,490	24.0	146
$\tau = 6, \rho = 0.96$	772	1,496	14.4	169
$\tau = 6, \rho = 0.98$	922	1,396	13.5	204

Notes: τ denotes the multi-hash distance threshold, and ρ denotes the embedding cosine-similarity threshold. The upper block reports hash-only grouping; the lower block reports grouping with embedding edges at the indicated thresholds. Largest-group share is the percentage of all 10,382 images contained in the largest connected component. Singletons are groups containing one image. The selected thresholds were $\tau = 6$ and $\rho = 0.95$.

and its nearest training image, and the last three columns give the fraction of evaluation images whose nearest training neighbor reaches the stated similarity. On the native splits, the median is 0.95 to 0.96, and at least 52% of images exceed 0.95. After grouped splitting, the median falls to 0.88 and 6.2% of test images still exceed 0.95 for ResNet-50. After the purge, the ResNet-50 audit shows no image above 0.95 by construction. The independent ConvNeXt-T audit, however, still finds 8.6% of validation and 10.9% of test images above 0.95, but only 0.3% and 0.4% above 0.98.

Table 4: Nearest Training Neighbor Similarity of Evaluation Images

Network	Setting	Partition	Median similarity	Fraction at similarity threshold		
				≥ 0.90	≥ 0.95	≥ 0.98
ResNet-50	Native	Validation	0.958	0.853	0.559	0.121
ResNet-50	Native	Test	0.952	0.844	0.521	0.125
ResNet-50	Grouped	Validation	0.886	0.389	0.060	0.001
ResNet-50	Grouped	Test	0.879	0.383	0.062	0.000
ResNet-50	Purged	Validation	0.881	0.354	0.000	0.000
ResNet-50	Purged	Test	0.865	0.299	0.000	0.000
ConvNeXt-T	Native	Validation	0.958	0.881	0.609	0.257
ConvNeXt-T	Native	Test	0.960	0.856	0.623	0.237
ConvNeXt-T	Purged	Validation	0.898	0.485	0.086	0.003
ConvNeXt-T	Purged	Test	0.879	0.402	0.109	0.004

A handcrafted-feature baseline reaches ROC-AUC 0.553 (accuracy 0.547) from file metadata, which is close to chance, but 0.925 (accuracy 0.859) from global color and texture statistics under group cross-validation. The high performance of this handcrafted-feature baseline indicates that global color and texture statistics contain substantial label information. This agrees with the biology of bleaching, which changes tissue color, but it does not establish that trained networks rely primarily on these cues.

4.2 Tuning and Training Dynamics

Each study completed 5 to 9 of its 10 allotted trials, and the remaining trials were pruned or failed. Tuning took 53.7 minutes in total, as shown in **Table 5**, where the negative log-likelihood is the best validation value on the 30% subset, the time is the study duration, and a layer decay of 1.00 indicates that layerwise decay was not used. EfficientNet-B0, ConvNeXt-T, and MaxViT-T completed only five trials. The best negative log-likelihood ranged from 0.0735 for the fusion network to 0.1300 for MaxViT-T. Tuned configurations differ substantially: the learning rate ranged from $1.0e-4$ for ViT-B/16 to $7.5e-4$

for ResNet-50, MixUp was disabled for ResNet-50 and used for the others, the CutMix probability ranged from 0.007 to 0.42, and only ResNet-50 and EfficientNet-B0 used a linear-probe epoch.

Table 5: Hyperparameter Search Summary and Selected Hyperparameters by Architecture

Model	Search summary			Selected hyperparameters							
	Completed trials	Time (s)	Best val. NLL	LR	WD	Dropout	Stoch. depth	LS	MixUp α	CutMix Pr	Layer decay
ResNet-50	9	261	0.1145	7.5×10^{-4}	0.0016	0.31	0.20	0.003	0.0	0.086	1.00
EfficientNet-B0	5	245	0.1279	4.4×10^{-4}	0.0010	0.23	0.11	0.012	0.2	0.208	1.00
ViT-B/16	7	616	0.0932	1.0×10^{-4}	0.0495	0.05	0.11	0.010	0.4	0.140	0.78
Swin-T	9	449	0.0983	1.1×10^{-4}	0.0088	0.05	0.06	0.023	0.2	0.052	0.93
ConvNeXt-T	5	367	0.0749	2.8×10^{-4}	0.0144	0.21	0.005	0.010	0.2	0.107	0.84
MaxViT-T	5	705	0.1300	3.1×10^{-4}	0.0020	0.09	0.07	0.056	0.2	0.416	0.76
Fusion	7	582	0.0735	2.8×10^{-4}	0.0012	0.42	0.23	0.024	0.2	0.007	0.92

Notes: Time denotes the total hyperparameter-search duration for each architecture. NLL, negative log-likelihood; LR, learning rate; WD, weight decay; LS, label-smoothing factor. Stoch. depth denotes the stochastic-depth probability; CutMix Pr denotes the CutMix application probability. Each architecture was allotted 10 trials; the completed-trial count excludes pruned and failed trials. Best validation NLL is the tuning objective obtained when training on a fixed 30% subset of the training images. A layer-decay factor of 1.00 indicates that layerwise learning-rate decay was not used.

Mean training curves in **Figure 6** show that the mean validation NLL is minimal at the third epoch for ResNet-50, ViT-B/16, MaxViT-T, and the fusion network, and stays within a narrow band for the remaining models. Early stopping ended individual runs after 5 to 11 epochs. ResNet-50 reaches the lowest training loss (about 0.08 by epoch 7), but its validation loss rises after epoch 3, while models with stronger regularization keep a training loss of about 0.17 to 0.28. Training losses are not directly comparable across models, since MixUp, CutMix, and label smoothing differ. From the second epoch onward, the mean validation ROC-AUC lies between about 0.988 and 0.999 for all models, and the lower first-epoch values of ResNet-50 and EfficientNet-B0 belong to the linear-probe epoch. Validation ROC-AUC is saturated and cannot discriminate between architectures, which supports the use of NLL for selection.

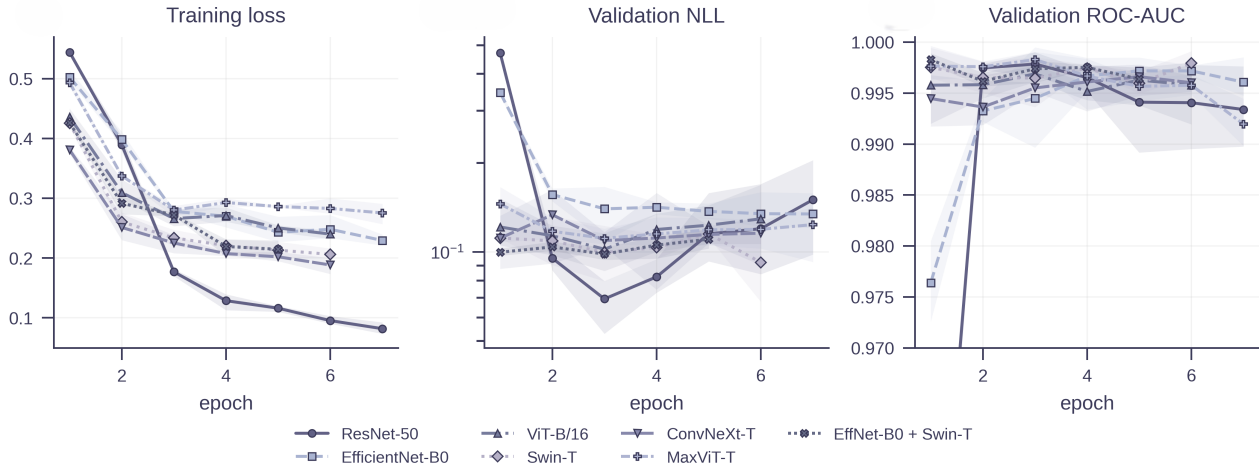


Figure 6: Mean Training Loss, Validation NLL, and Validation ROC-AUC per Epoch

4.3 Classification Performance

Table 6 reports test performance over three seeds. Across all configurations, mean accuracy lies between 93.0% and 95.6%, Healthy F1 between 0.915 and 0.946, and ROC-AUC between 0.969 and 0.984. In the robust arm, Swin-T (0.9520) and MaxViT-T (0.9491) have the highest mean accuracy, and ResNet-50 (0.9300) the lowest. The seed standard deviation in the robust arm ranges from 0.25 percentage points (fusion) to 1.05 (ViT-B/16), and it reaches 2.3 percentage points for ResNet-50 in the basic arm, which is comparable with the whole spread between models. Seed variation is therefore comparable to the between-architecture variation. The final partitions are leakage-mitigated rather than scene-disjoint, so these values are optimistic estimates.

Table 6: Test Performance, Mean and Standard Deviation over Three Seeds

Model	Arm	Accuracy	Healthy F1 score	AUC
ResNet-50	basic	0.9296 $\pm$ 0.0231	0.9170 $\pm$ 0.0248	0.9777 $\pm$ 0.0017
ResNet-50	robust	0.9300 $\pm$ 0.0075	0.9154 $\pm$ 0.0096	0.9691 $\pm$ 0.0114
EfficientNet-B0	basic	0.9505 $\pm$ 0.0056	0.9386 $\pm$ 0.0071	0.9773 $\pm$ 0.0059
EfficientNet-B0	robust	0.9386 $\pm$ 0.0041	0.9236 $\pm$ 0.0051	0.9709 $\pm$ 0.0039
ViT-B/16	basic	0.9558 $\pm$ 0.0032	0.9458 $\pm$ 0.0043	0.9828 $\pm$ 0.0062
ViT-B/16	robust	0.9452 $\pm$ 0.0105	0.9328 $\pm$ 0.0130	0.9835 $\pm$ 0.0078
Swin-T	basic	0.9522 $\pm$ 0.0078	0.9424 $\pm$ 0.0093	0.9835 $\pm$ 0.0043
Swin-T	robust	0.9520 $\pm$ 0.0047	0.9422 $\pm$ 0.0055	0.9756 $\pm$ 0.0040
ConvNeXt-T	robust	0.9392 $\pm$ 0.0083	0.9254 $\pm$ 0.0098	0.9780 $\pm$ 0.0052
MaxViT-T	robust	0.9491 $\pm$ 0.0062	0.9379 $\pm$ 0.0075	0.9816 $\pm$ 0.0068
Fusion	robust	0.9427 $\pm$ 0.0025	0.9294 $\pm$ 0.0039	0.9820 $\pm$ 0.0063

Seed-averaged ensembles of the robust arm are reported in **Table 7** together with scene-cluster accuracy intervals. Accuracy depends on the class prior, and the test partition contains 41.5% Healthy images, so balanced accuracy is reported as well. It ranges from 0.929 (ResNet-50) to 0.956 (Swin-T) and preserves the ordering of accuracy except for small swaps among models that are within sampling uncertainty. Image-level bootstrap intervals, which assume independent images, are only 1.9 to 2.4 percentage points wide (for example 0.9269 to 0.9497 for ConvNeXt-T), whereas the cluster intervals are 8 to 12 percentage points wide (0.8825 to 0.9991 for the same model). With 112 test scene groups, the effective sample size is closer to the number of groups than to the 1,710 images. The soft vote over all models reaches 0.9491 accuracy and 0.9836 ROC-AUC, which did not exceed the best single seed-ensembled models (Swin-T 0.9567, MaxViT-T 0.9532). The logistic-regression stack fitted on validation logits has a sensitivity of 0.9478, which is close to but slightly below the highest value (Swin-T, 0.9492). It also has the lowest specificity (0.9371) and a log-loss of 0.341, about 1.8 to 2.3 times that of the single-model ensembles. The class-prior shift between validation (49.4% Healthy) and test (41.5% Healthy) is a plausible explanation, since a stacker fitted on validation cannot anticipate it, but the single split does not allow a formal test of this explanation.

Table 7: Seed-Averaged Robust-Arm Models and Ensembles on the Test Set

Model	Accuracy			Discrimination			Calibration		Class recall
	Overall	Balanced	95% CI	Healthy F1	ROC-AUC	Log-loss	ECE	Sens. (Healthy)	Spec. (Bleached)
ResNet-50	0.9316	0.9289	0.8820–0.9800	0.9171	0.9728	0.1913	0.0442	0.9126	0.9451
EfficientNet-B0	0.9398	0.9327	0.8875–0.9948	0.9247	0.9726	0.1811	0.0361	0.8914	0.9740
ViT-B/16	0.9474	0.9427	0.8965–0.9978	0.9352	0.9875	0.1602	0.0190	0.9154	0.9700
Swin-T	0.9567	0.9556	0.9169–0.9992	0.9479	0.9766	0.1511	0.0232	0.9492	0.9620
ConvNeXt-T	0.9386	0.9346	0.8825–0.9991	0.9248	0.9774	0.1746	0.0304	0.9111	0.9580
MaxViT-T	0.9532	0.9494	0.9077–0.9988	0.9426	0.9854	0.1571	0.0341	0.9267	0.9720
Fusion	0.9392	0.9328	0.8925–0.9866	0.9243	0.9877	0.1461	0.0400	0.8956	0.9700
Soft vote, all models	0.9491	0.9444	—	0.9373	0.9836	0.1488	0.0284	0.9168	0.9720
Stack, validation-fitted	0.9415	0.9425	—	0.9307	0.9772	0.3413	0.0531	0.9478	0.9371

Notes: CI, scene-cluster bootstrap confidence interval for accuracy; ROC-AUC, area under the receiver operating characteristic curve; ECE, expected calibration error; Sens., sensitivity (Healthy recall); Spec., specificity (Bleached recall). Healthy is the positive class. — indicates that the confidence interval was not computed.

The error profiles below are descriptive and untested. The confusion matrices in **Figure 7** and **Figure 8** show that most errors are Healthy images classified as Bleached for EfficientNet-B0, the fusion network, MaxViT-T, and ConvNeXt-T. EfficientNet-B0 makes 77 such errors against 26 in the opposite direction, the fusion network 74 against 30, MaxViT-T 52 against 28, and ConvNeXt-T 63 against 42. Only Swin-T is balanced, with 36 and 38 errors, and ResNet-50 is nearly balanced with 62 and 55. The Healthy recall of EfficientNet-B0, 0.891, is therefore 5.8 percentage points below that of Swin-T, 0.949, a larger gap than the difference in overall accuracy (1.7 percentage points). These error profiles are descriptive, since they come from one split and were not tested for significance.

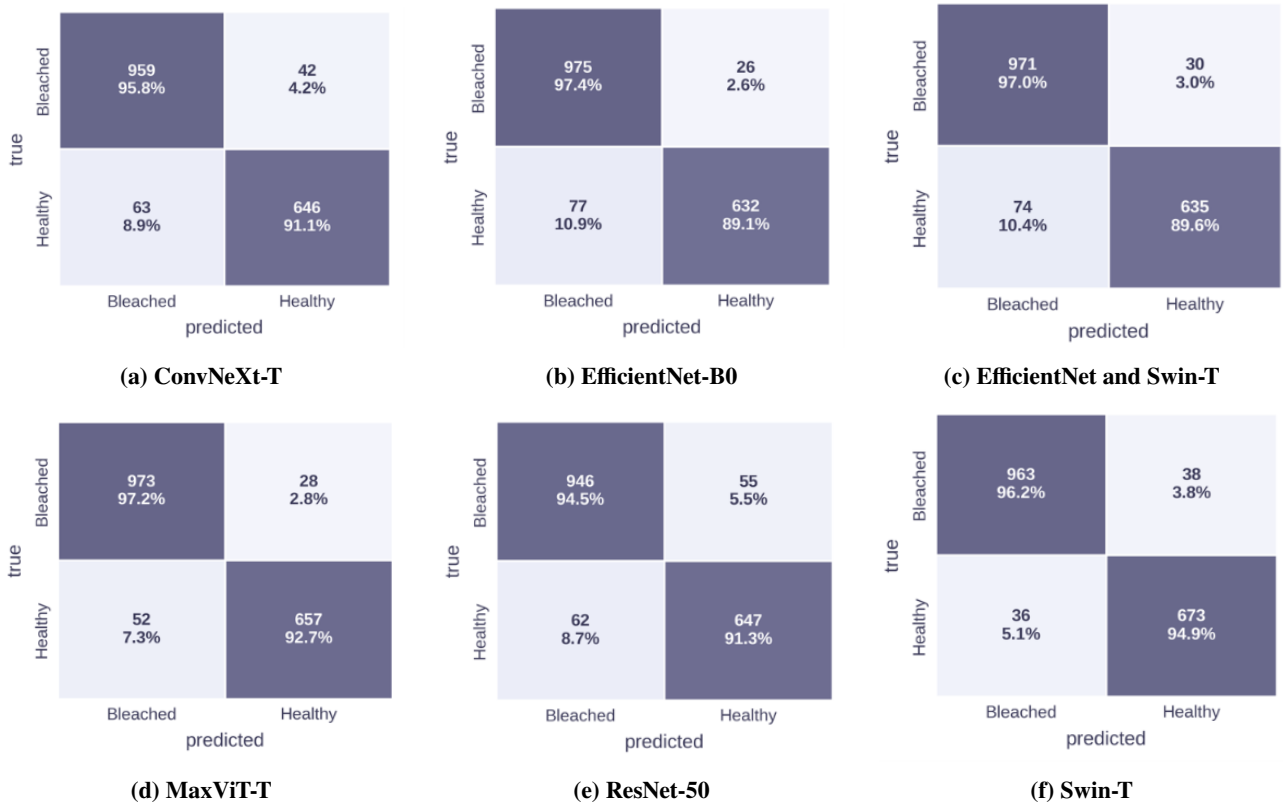


Figure 7: Confusion Matrices of Six Seed-Ensembled Models, Robust Arm

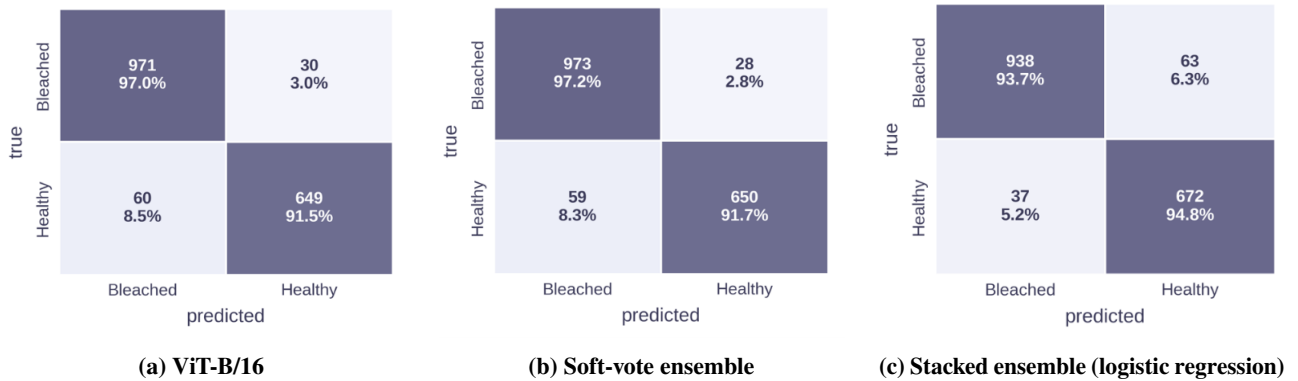
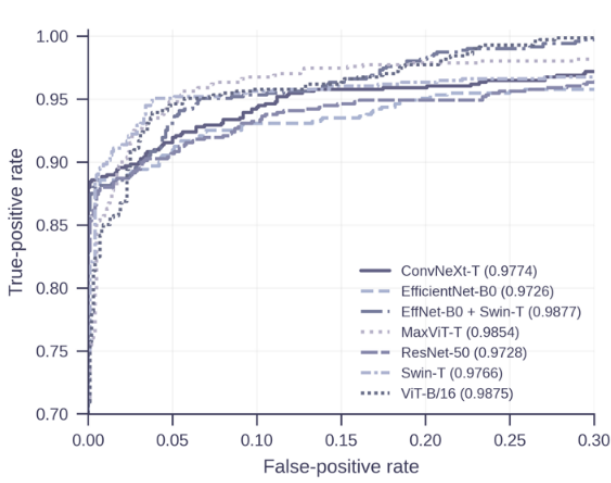
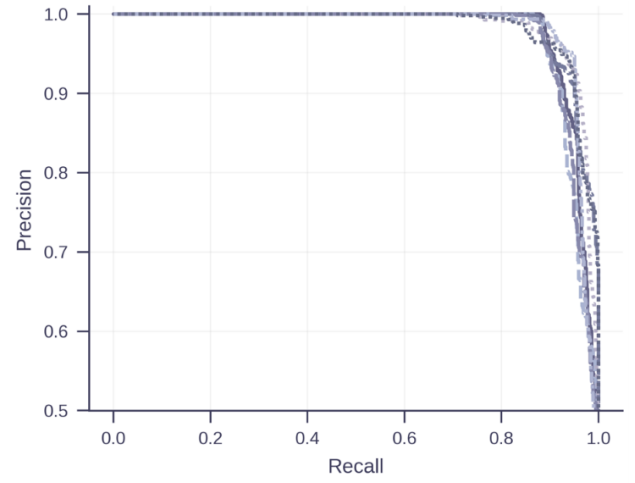


Figure 8: Confusion Matrices of ViT-B/16, Soft Vote, and Stacked Ensemble, Robust Arm

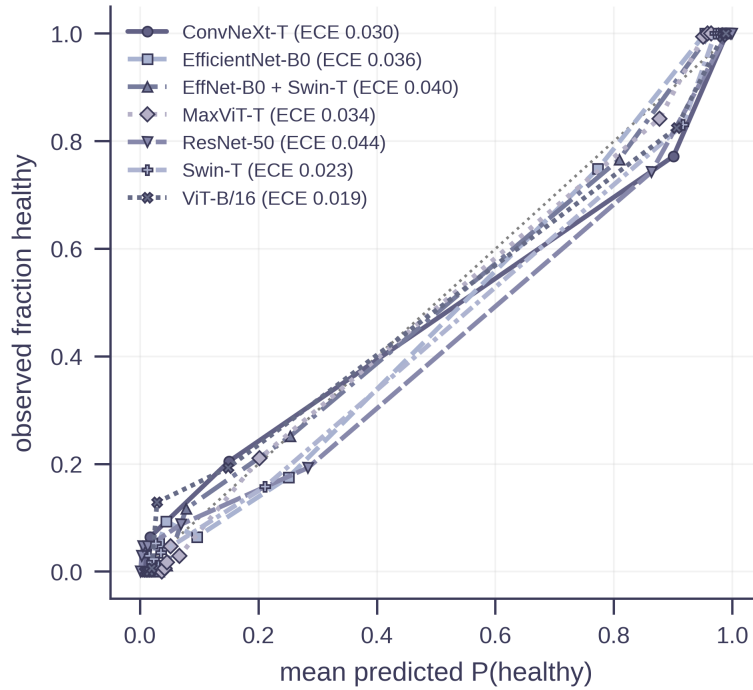
The zoomed ROC curves in **Figure 9a** rise to a true-positive rate of about 0.88 at a false-positive rate near zero and then flatten at 0.95 to 0.97 for most models, whereas ViT-B/16 and the fusion network continue to about 1.0 at a false-positive rate of 0.3. This tail explains the higher ROC-AUC of these two models, 0.9875 and 0.9877, despite unremarkable accuracy. The precision-recall curves in **Figure 9b** stay near 1.0 up to a recall of about 0.88 and then fall steeply. About 12% of Healthy images score below the highest-scoring Bleached images. The reliability diagram in **Figure 10**, with ten equal-count bins on uncalibrated probabilities, follows the diagonal within a few points for most models. ResNet-50 and EfficientNet-B0 lie below the diagonal between predicted Healthy probabilities of 0.3 and 0.9, which means that the Healthy probability is overestimated. The ECE of the seed-enssembled models in **Table 7**, computed with equal-width bins, ranges from 0.019 for ViT-B/16 to 0.044 for ResNet-50.



(a) Zoomed ROC Curves, Robust Arm



(b) Precision-Recall Curves, Robust Arm

Figure 9: ROC and Precision-Recall Results**Figure 10: Reliability Diagram of Seed-Ensembled Models**

4.4 Statistical Comparison of Architectures

Of the 21 architecture pairs, only one remained significant after joint Holm correction. The other 20 differences are not distinguishable from sampling variation under the scene-aware procedure. This does not show that the architectures perform equivalently. The wide cluster intervals indicate substantial uncertainty and limited evidence for distinguishing small differences, so the data do not support a reliable ranking of the seven architectures. The importance of scene-aware inference is illustrated by the augmentation contrasts in **Table 8**. For EfficientNet-B0 and ViT-B/16, the McNemar test gives Holm-adjusted p-values below 0.0001, whereas the two-sided cluster sign-flip test gives 0.36 and 0.37, and for ResNet-50 the two tests give 0.013 and 0.73. Treating 1,710 correlated images as independent overstates the evidence. The sign-flip test assumes that scene-group differences are exchangeable under the null, and the joint Holm correction over both metrics is conservative.

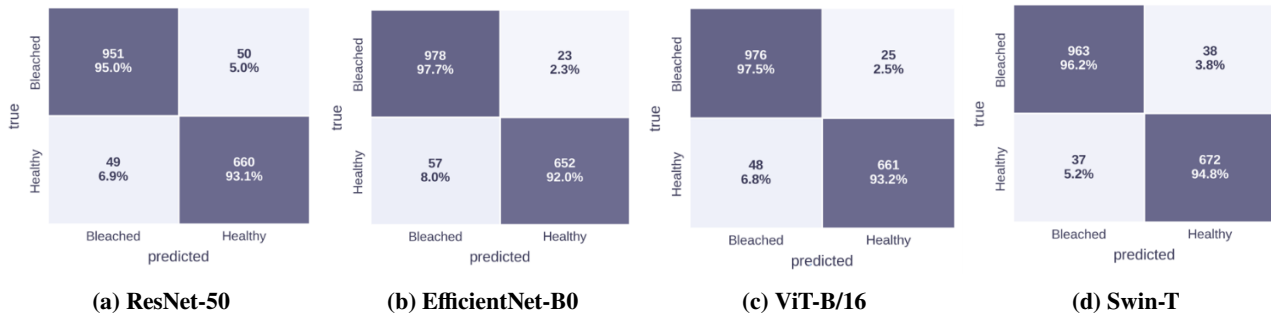
Table 8: Effect of the Robust Augmentation Package on Clean Test Images: Robust Minus Basic

Model	Accuracy change (95% CI)	ROC-AUC change	Healthy F1 change	McNemar adjusted p	Cluster sign-flip adjusted p
EfficientNet-B0	-0.0119 (-0.0178, -0.0060)	-0.0065	-0.0150	< 0.0001	0.362
ResNet-50	+0.0004 (-0.0750, +0.0758)	-0.0085	-0.0017	0.013	0.726
Swin-T	-0.0002 (-0.0195, +0.0192)	-0.0079	-0.0002	1.000	1.000
ViT-B/16	-0.0105 (-0.0440, +0.0230)	+0.0007	-0.0131	< 0.0001	0.374

Notes: Changes are calculated as robust minus basic and are reported on the original metric scale, not in percentage points. CI denotes the 95% t confidence interval based on three paired training-seed differences; it does not quantify scene-sampling uncertainty. ROC-AUC denotes the area under the receiver operating characteristic curve. For each test, p -values are Holm-adjusted across the four architecture comparisons. McNemar results are included for comparison only because the test does not account for scene clustering; the cluster sign-flip test uses scene groups.

4.5 Effect of the Robust Augmentation Package

The robust arm bundles noise, blur, corner dropout, and caption-band dropout, so differences are attributed to the package and not to individual components. On the full clean test set, the robust arm changes mean accuracy by -0.56 percentage points and ROC-AUC by -0.006, averaged over the four architectures that have both arms (on the clean subset of at most 600 images used in the corruption sweep, the changes are -0.54 percentage points and -0.003). Per model, the accuracy change is -1.19 percentage points for EfficientNet-B0, -1.05 for ViT-B/16, and +0.04 for ResNet-50 and -0.02 for Swin-T, as reported in **Table 8**. Seed-level intervals exclude zero only for EfficientNet-B0, and none of the contrasts is significant under the cluster sign-flip test after Holm correction. The basic-arm confusion matrices shown in **Figure 11** exhibit the same error structure, for example 57 Healthy errors for EfficientNet-B0 compared with 77 in the robust arm.

**Figure 11: Confusion Matrices of Four Seed-Ensembled Models, Basic Arm**

Under corruption, the largest differences occur for strong degradations, as shown in **Table 9**. Averaged over the four models trained with both arms, the robust arm shows descriptive accuracy gains of 4.5 percentage points (and 0.020 AUC) at noise 0.15, 4.1 at blur $\sigma = 3$, 3.0 under shot noise, and 1.3 under speckle noise. These are perturbation families on which the robust arm was explicitly trained, so a gain is expected. For conditions not used in training, the differences are near zero or slightly negative (JPEG -0.06 percentage points, red loss -0.18), which indicates little transfer beyond the trained families. The robust arm also loses 1.1 to 1.2 percentage points under haze, 0.8 under brightening, and 0.5 to 0.7 under mild blur. These contrasts are mean differences over four models on at most 600 images per condition. No confidence intervals or significance tests were computed for them, and differences below one percentage point are not interpreted. The contrasts are descriptive and should not be read as established effects, in contrast to the clean-set contrasts of **Table 8**, which were tested under the scene-aware procedure.

4.6 Representation Geometry (Descriptive)

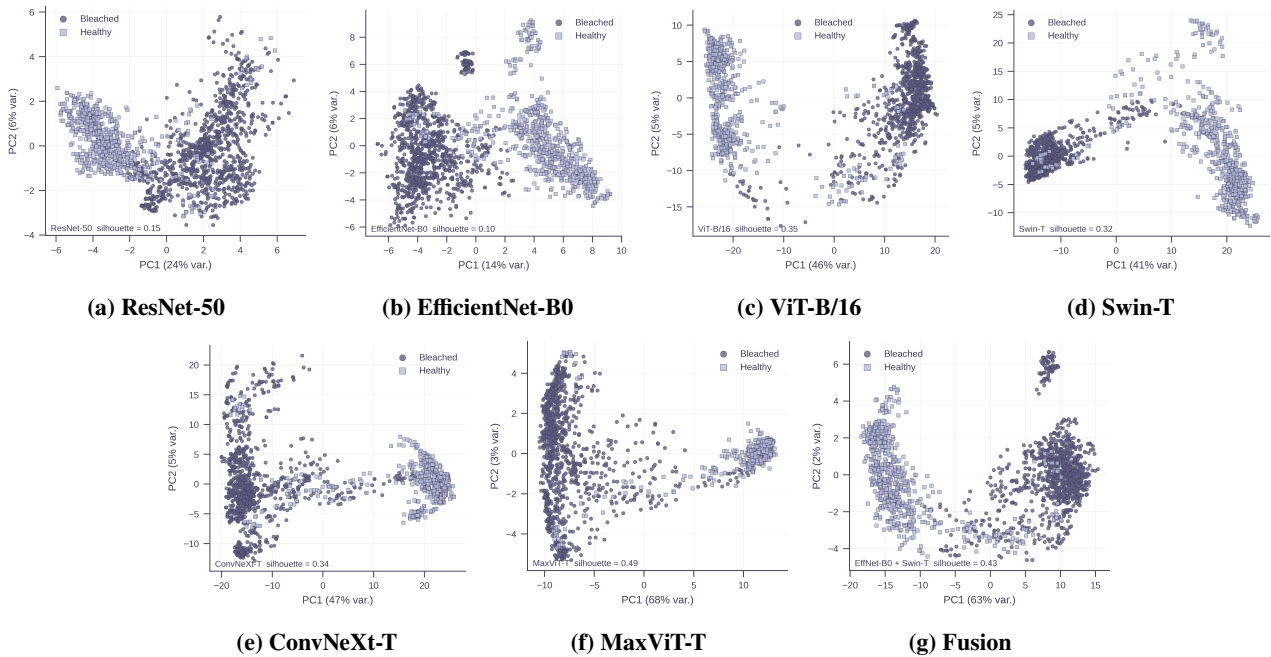
The following observations are descriptive and untested. PCA projections of penultimate features for all 1,710 test images in **Figure 12** differ across models. MaxViT-T, with a silhouette of 0.49, and the fusion network, with 0.43, produce a compact Healthy cluster and a broader Bleached distribution, whereas ResNet-50, with 0.15, and EfficientNet-B0, with 0.10, show strong overlap in the first two components. ConvNeXt-T, ViT-B/16, and Swin-T are intermediate, with silhouette values of 0.34, 0.35, and 0.32. The first component explains 68% of the variance for MaxViT-T and 63% for the fusion network but only 14% for EfficientNet-B0 and 24% for ResNet-50. EfficientNet-B0 and the fusion network also show small isolated clusters, for example a Bleached group at the top of both plots. The silhouette is a coarse measure and does not

Table 9: Corruption Performance Changes: Robust Minus Basic, Averaged over Four Architectures (Descriptive, Untested)

Condition	Accuracy change (pp)	ROC-AUC change	Condition	Accuracy change (pp)	ROC-AUC change
Clean	−0.54	−0.0034	Blur 0.5	−0.50	−0.0022
Gaussian noise 0.02	+0.01	−0.0026	Blur 1	−0.68	−0.0015
Gaussian noise 0.05	+0.24	+0.0004	Blur 2	−0.04	+0.0002
Gaussian noise 0.10	+0.90	+0.0051	Blur 3	+4.11	+0.0058
Gaussian noise 0.15	+4.47	+0.0201	Darker 0.4	−0.39	−0.0007
Speckle noise 0.10	+1.25	+0.0071	Brighter 0.4	−0.83	−0.0005
Shot noise 0.10	+2.99	+0.0134	Haze 0.35	−1.10	−0.0030
JPEG quality 30	−0.06	−0.0021	Haze 0.60	−1.15	−0.0041
Red loss 0.4	−0.18	+0.0013			

Notes: Values are mean differences (robust minus basic) across ResNet-50, EfficientNet-B0, ViT-B/16, and Swin-T. Accuracy changes are reported in percentage points (pp); ROC-AUC changes are reported on the original metric scale. Positive values favor the robust arm. All conditions, including clean images, were evaluated on the same fixed test subset of at most 600 images. Gaussian noise levels denote the noise standard deviation; blur levels denote the Gaussian standard deviation in pixels. No confidence intervals or significance tests were computed; the differences are descriptive.

rank accuracy: EfficientNet-B0, with the lowest silhouette (0.10), reaches 0.9398 accuracy, above ConvNeXt-T (0.9386, silhouette 0.34) and the fusion network (0.9392, silhouette 0.43).

**Figure 12: PCA of Penultimate Features on 1,710 Test Images**

4.7 Explainability

Figure 13 summarizes the quantitative tests, and **Table 10** lists deletion and insertion AUC for the 48 fixed test images. Grad-CAM maps achieved lower deletion AUC and higher insertion AUC than smooth random maps for all seven models, but the margin varies greatly. These results demonstrate relative map utility under this protocol, not biological faithfulness or causal localization. ResNet-50 has the largest margin (deletion 0.844 against 0.911 and insertion 0.960 against 0.912), whereas Swin-T (0.909 against 0.930 and 0.937 against 0.929) and ViT-B/16 (insertion 0.951 against 0.946) are close to random. Deletion AUC values of 0.84 to 0.91 show that the predicted probability stays high when pixels are progressively replaced by a blurred copy. The curves in **Figure 13(a)** and **(b)** show why: the fully blurred image, which keeps only

coarse color and layout, still receives a predicted-class probability of about 0.81 (ResNet-50) to 0.92 (ViT-B/16). Because color statistics alone carry substantial label information according to the shortcut baseline, a slow decay is expected, and a high deletion AUC therefore does not by itself show that Grad-CAM is unfaithful. The deletion curves also recover at the last step, because a fully blurred image is a coherent image, whereas a partly replaced one contains artificial edges. Swin-T produced blank maps for 4.2% of images on average (standard deviation 7.2 across seeds), and all other models produced none. A blank map means that the gradient-weighted activations of the target layer are negative everywhere, so the rectification removes all relevance.

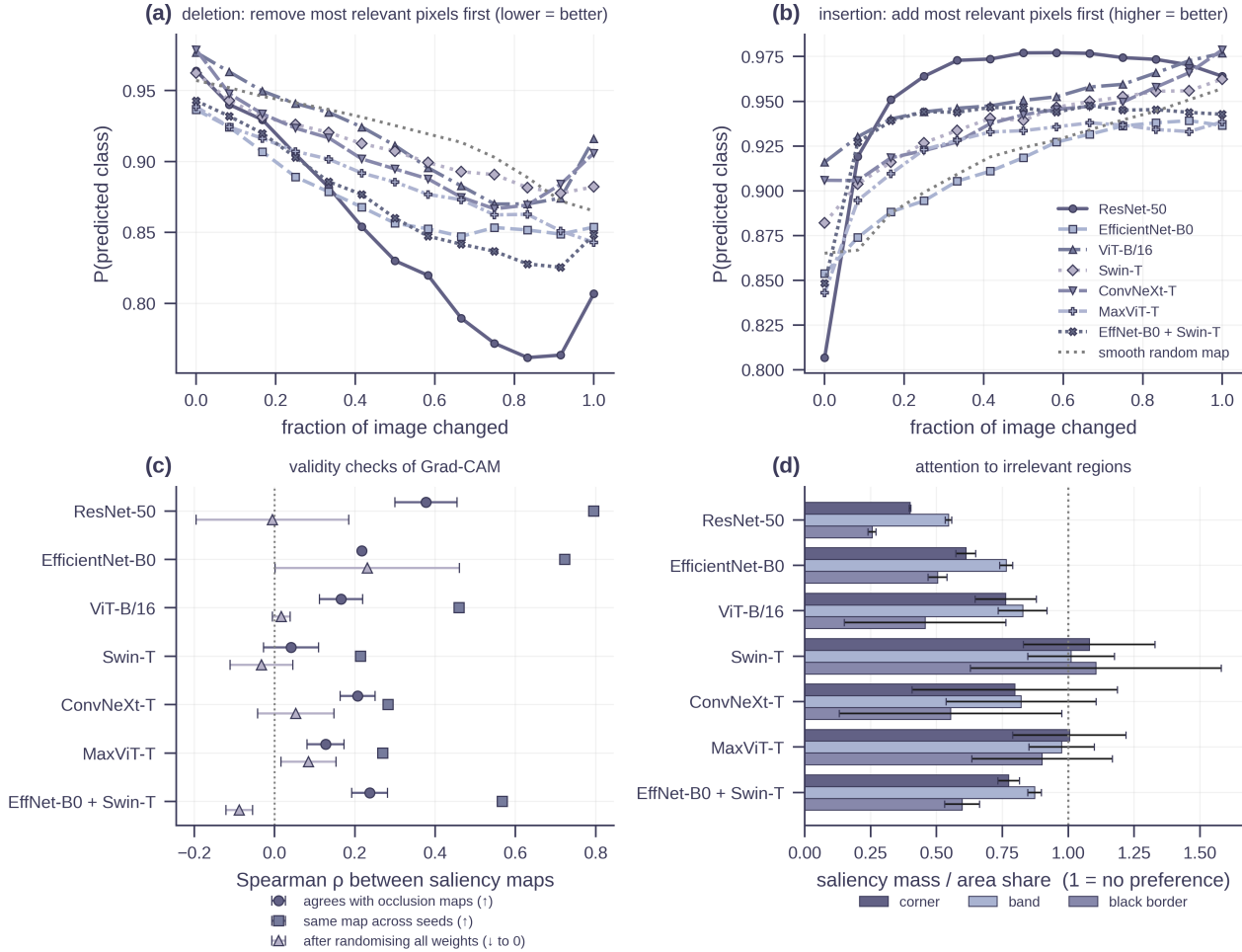


Figure 13: Quantitative Evaluation of Grad-CAM, Mean over Three Seeds on 48 Test Images

Table 10: Deletion and Insertion Area Under the Curve of Grad-CAM, Mean and Standard Deviation over Seeds

Model	Deletion AUC		Insertion AUC	
	Grad-CAM map	Random map	Grad-CAM map	Random map
ResNet-50	0.844 (0.001)	0.911 (0.014)	0.960 (0.007)	0.912 (0.007)
EfficientNet-B0	0.873 (0.009)	0.894 (0.010)	0.913 (0.003)	0.896 (0.007)
ViT-B/16	0.914 (0.021)	0.946 (0.012)	0.951 (0.010)	0.946 (0.011)
Swin-T	0.909 (0.023)	0.930 (0.020)	0.937 (0.014)	0.929 (0.020)
ConvNeXt-T	0.904 (0.025)	0.924 (0.022)	0.939 (0.024)	0.918 (0.024)
MaxViT-T	0.887 (0.014)	0.919 (0.007)	0.924 (0.001)	0.912 (0.008)
Fusion	0.871 (0.029)	0.916 (0.012)	0.939 (0.006)	0.913 (0.013)

Figure 13(c) and (d) and Table 11 report localization and sanity checks. In Table 11, the columns on randomized heads, upper halves, and all layers give Spearman correlations between trained-model maps and maps after parameter randomization; the three mass ratio columns give saliency mass ratios; black borders cover 6.2% of the explained images

on average; and seed consistency is a single value per model and has no standard deviation. Agreement with occlusion sensitivity is low to moderate for all models (Spearman 0.04 to 0.38), highest for ResNet-50 and lowest for Swin-T. This is expected, because gradient-based and perturbation-based maps measure different notions of relevance. All correlations after randomization of the head, the upper half of the layers, or all layers lie between -0.09 and 0.23 , which indicates that the maps depend on the learned weights [39]. The largest values occur for EfficientNet-B0 (0.16 and 0.23) with large standard deviations, so the evidence of weight dependence is weaker for this model. Seed-to-seed consistency is high for ResNet-50 (0.80) and EfficientNet-B0 (0.72) and low for MaxViT-T (0.27), ConvNeXt-T (0.28), and Swin-T (0.21), so explanations of the latter should not be interpreted as stable properties of the architecture.

Table 11: Localization, Sanity Checks, and Region Mass Ratios, Mean and Standard Deviation over Seeds

Model	Localization		Sanity checks			Region mass ratios		
	Occlusion agreement	Seed consistency	Head randomized	Upper half randomized	All layers randomized	Corner mass ratio	Band mass ratio	Border mass ratio
ResNet-50	0.377 (0.077)	0.795	0.020 (0.071)	-0.019 (0.044)	-0.005 (0.190)	0.399 (0.003)	0.546 (0.012)	0.255 (0.015)
EfficientNet-B0	0.217 (0.003)	0.723	-0.004 (0.139)	0.161 (0.312)	0.231 (0.230)	0.612 (0.038)	0.764 (0.025)	0.504 (0.036)
ViT-B/16	0.166 (0.054)	0.460	-0.077 (0.073)	0.028 (0.158)	0.017 (0.022)	0.763 (0.116)	0.827 (0.092)	0.457 (0.307)
Swin-T	0.041 (0.069)	0.214	0.094 (0.040)	0.040 (0.013)	-0.032 (0.078)	1.080 (0.250)	1.011 (0.164)	1.105 (0.476)
ConvNeXt-T	0.207 (0.043)	0.283	0.107 (0.182)	0.073 (0.052)	0.053 (0.095)	0.797 (0.390)	0.821 (0.285)	0.553 (0.422)
MaxViT-T	0.127 (0.046)	0.269	-0.022 (0.045)	0.025 (0.068)	0.085 (0.069)	1.004 (0.215)	0.975 (0.124)	0.901 (0.267)
Fusion	0.237 (0.044)	0.567	0.006 (0.126)	0.080 (0.202)	-0.088 (0.033)	0.774 (0.041)	0.872 (0.026)	0.597 (0.066)

The saliency mass ratios in **Figure 13(d)** and **Table 11** are below one for ResNet-50, EfficientNet-B0, ViT-B/16, ConvNeXt-T, and the fusion network in the corner, band, and border regions. This indicates that these models assign less relevance to peripheral regions than a uniform saliency map would. Swin-T exceeds one in all three regions (1.080, 1.011, and 1.105, the last with a standard deviation of 0.48), and MaxViT-T is close to one (1.004, 0.975, and 0.901). The saliency mass results do not exclude reliance on peripheral cues for these two models, but the occlusion results in **Table 12** show that peripheral regions have limited behavioral impact under the tested perturbations. Replacing any single corner changes at most 0.70% of predictions, all four corners together at most 1.44%, and both caption bands at most 2.55%, with accuracy changes within 0.8 percentage points. The flip fractions of Swin-T (0.0103 for corners) and MaxViT-T (0.0096) lie within the range of the other models (0.0057 to 0.0144), so the tested occlusions do not show a behavioral dependence of these two models on corner or caption regions. These results show reduced sensitivity to corners and caption bands in the robust models. This finding does not demonstrate the absence of a logo or caption shortcut, because these models were trained with corner and caption-band dropout so low sensitivity is the expected outcome. The occlusion test was run on the robust arm only, so it cannot show how much of the insensitivity is due to the augmentation.

Table 12: Fraction of Predictions Flipped by Occlusion, Robust Arm

Model	Predictions flipped		Accuracy change	
	All corners	Both bands	Corners	Bands
ResNet-50	0.0115	0.0253	-0.0006	-0.0012
EfficientNet-B0	0.0076	0.0218	-0.0010	-0.0047
ViT-B/16	0.0090	0.0138	0.0039	0.0045
Swin-T	0.0103	0.0160	0.0037	0.0016
ConvNeXt-T	0.0057	0.0119	0.0029	0.0076
MaxViT-T	0.0096	0.0172	0.0010	0.0051
Fusion	0.0144	0.0255	-0.0004	-0.0041

The following reading of representative maps is qualitative and exploratory. Maps for four test images are shown in **Figure 14**, using the first seed of each model. All seven models classify the first three images correctly, with Bleached probabilities of at least 0.95 for the first and third image and Healthy probabilities of at least 0.96 for the second. For the fourth image, a cyan branching colony, six models predict Healthy with probabilities of 0.83 (EfficientNet-B0) to 0.98 (ViT-B/16 and ConvNeXt-T), whereas ResNet-50 predicts Bleached with a Healthy probability of 0.39. The maps differ strongly in shape even though most predictions agree. ResNet-50 and ConvNeXt-T give compact regions that lie on the coral in the first and third images. ConvNeXt-T places relevance on open water in the second image and above the colony in the fourth, and ResNet-50 highlights the lower part of the colony in the fourth image, where it errs. EfficientNet-B0 produces broad regions that cover much of each image, including background. ViT-B/16 gives fragmented maps with many small regions scattered over water, borders, and coral. The fusion network shows several separate regions, some at image

edges. MaxViT-T concentrates relevance near image corners and edges, with almost nothing on the coral in the third image. Swin-T returns blank maps for the first and third images in this seed and a diffuse map for the second. These patterns agree with the quantitative results: the maps of ResNet-50 are the most aligned with occlusion sensitivity, and those of Swin-T and MaxViT-T are the least aligned and the most peripheral. The single misclassification in **Figure 14** is not sufficient to characterize error behavior, which the confusion matrices describe instead.

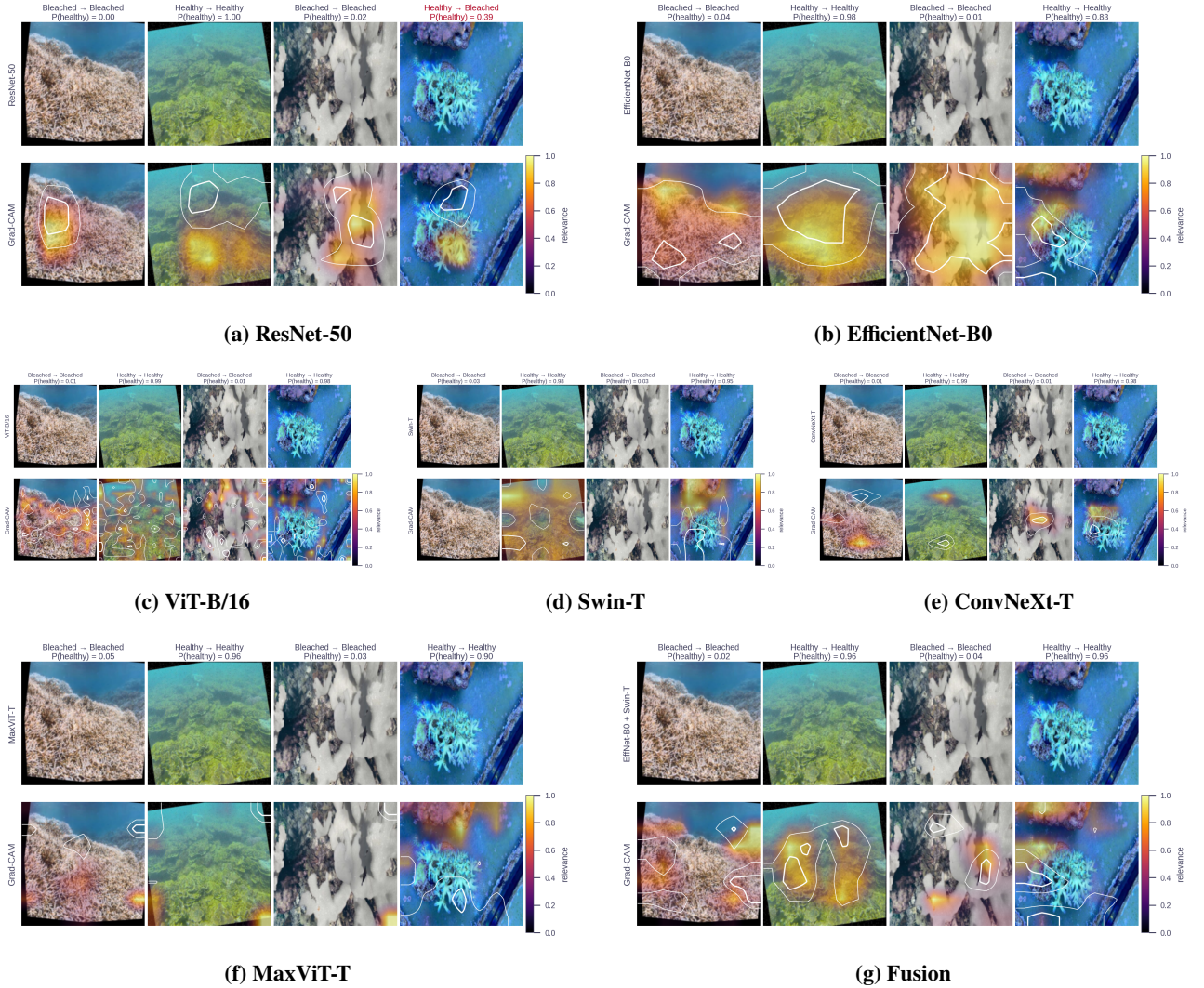


Figure 14: Grad-CAM Maps of Seven Models on Four Identical Test Images, First Seed of Each Model

4.8 Computational Cost

Table 13 reports the profile on the Tesla T4. EfficientNet-B0 has the fewest parameters (4.01 M) and operations (0.77 GFLOPs) and the highest throughput (1,727 images per second at batch size 32), but its batch-size-one latency (9.15 ms) is higher than that of ResNet-50 (6.93 ms) and of ViT-B/16 (6.27 ms), which has 35.1 GFLOPs. ViT-B/16 with 35.1 GFLOPs is faster at batch size 1 than EfficientNet-B0 with 0.77 GFLOPs, so FLOPs are a poor predictor of latency. MaxViT-T is the slowest at batch size 1 (30.5 ms), has the lowest throughput (318 images per second), and needs 1,635 MB of peak inference memory, nearly four times that of ConvNeXt-T (435 MB). The fusion network needs 23.9 ms per image, close to the sum of the latencies of its two branches (9.15 ms and 13.15 ms).

Table 13: Efficiency Profile with Mixed Precision on a Tesla T4

Model	Model complexity			Inference			Training	
	Parameters (millions)	Size, 32-bit (megabytes)	Floating-point operations (billions)	Median latency, batch size 1 (milliseconds)	95th percentile latency (milliseconds)	Throughput, batch size 32 (images/s)	Peak memory (megabytes)	Training time per epoch (seconds)
ResNet-50	23.51	89.7	8.17	6.93	7.45	911	584	25.0
EfficientNet-B0	4.01	15.3	0.77	9.15	9.95	1727	531	17.5
ViT-B/16	85.80	327.3	35.13	6.27	6.78	394	657	66.5
Swin-T	27.52	105.0	8.98	13.15	15.42	466	556	46.6
ConvNeXt-T	27.82	106.1	8.91	7.41	8.21	656	435	37.6
MaxViT-T	30.40	116.0	11.10	30.46	32.19	318	1635	71.0
Fusion	33.11	126.3	9.75	23.90	25.71	357	588	62.3

Preprocessing is separated from model time. Decoding and resizing to the 256-pixel cache was measured at 6.1 ms per image in a separate timing test, and the one-off cache construction took 46.1 s for 10,382 images (4.4 ms per image). GPU augmentation takes 18 to 24 ms per batch of 64 (0.29 to 0.37 ms per image) for the two arms, so it is small beside the 17 to 71 s per epoch of model training. The full run took 320.1 minutes, of which the listed stages account for 306.1 minutes. The stage wall-clock was 0.8 minutes for preprocessing, 0.1 for the hash audit, 0.7 for embedding grouping, 53.7 for tuning, 197.8 for training, 4.4 for test inference with flip averaging, 11.5 for the corruption sweep, 36.0 for explanation and embeddings, and 1.1 for efficiency profiling. Test inference with flip averaging takes 2.5 ms per image for ResNet-50 and 9.7 ms for MaxViT-T.

5. DISCUSSION

5.1 Evaluation Validity

More than half of the native evaluation images have a near-duplicate in the training partition, so accuracy on those splits does not provide a valid estimate of generalization to unseen scenes. Under scene-grouped partitioning, accuracy is 93% to 96%, and the models differ more in sensitivity than in accuracy. The embedding-based purge is also imperfect: the independent ConvNeXt-T audit still finds 10.9% of test images with a neighbor above 0.95, and some groups straddle partitions after the purge. Stricter purge thresholds, such as 0.90, would remove more of these pairs but require complete retraining and shrink the evaluation partition. The reported accuracy should therefore be read as an optimistic estimate of performance on new reefs.

5.2 Architecture Comparison

Differences between architectures are small relative to scene-level uncertainty. Cluster intervals span 8 to 12 percentage points, only one of 21 pairs is significant, and seed variation reaches the size of the between-model spread. Only one of 21 pairs differed significantly, and the remaining differences are not distinguishable from sampling variation, which is not proof of equivalence. Observations about individual models are descriptive and are not supported by significance tests. EfficientNet-B0 has 4.0 M parameters and the highest throughput, an efficiency advantage that was measured, but parity in accuracy with larger models is not established. Swin-T has the highest Healthy recall (0.949) and EfficientNet-B0 the highest Bleached recall (0.974), but the differences between these error profiles were not tested. The ranking by ROC-AUC differs from the ranking by accuracy, and the choice of operating point is application dependent. Where a missed bleaching event costs more than a false alarm, the decision threshold should favor Bleached recall, and it should be set on data matching the deployment prior. The test partition holds 41.5% Healthy images against 49.4% in validation, a class-prior shift that affects accuracy, F1, and calibration and penalizes the stack fitted on validation logits, so thresholds and calibration should be fitted on deployment-matched data.

5.3 Augmentation and Tuning

Noise and blur augmentation raised descriptive mean accuracy by 3.0 to 4.5 percentage points under strong noise, heavy blur, and shot noise, which are perturbation families the robust arm was trained on, lowered clean accuracy by 0.56 percentage points on average (not significant in any single model after scene-aware correction), and slightly reduced accuracy under haze, brightening, and mild blur. No gain appeared for JPEG compression or red loss, which were not part of training. Only four architectures have both arms, so the comparison is not a balanced ablation over all seven. The corruption contrasts were not tested for significance and are descriptive. The augmentation profile should match the expected degradation, and the 17 synthetic conditions, evaluated on at most 600 images each, only approximate real underwater deployment. Because tuning

used a 30% subset for three epochs and only 5 to 9 of the 10 allotted trials completed per model, the selected configurations were not confirmed with longer or fully completed searches.

5.4 Explanations and Shortcuts

The quantitative tests indicate that Grad-CAM maps carry model-specific information, because the maps outperform random maps and depend on trained weights, yet agreement with occlusion is low and varies between seeds for the transformer-based models. Grad-CAM maps should be treated as exploratory visualizations, not as reliable evidence of the image regions that drive a classification. At the same time, color and texture statistics alone achieve a ROC-AUC of 0.925, so global color carries a large share of the label information, and performance under unusual illumination or for species that bleach to non-white colors was not evaluated. This result also means that the reported image-level accuracy should not be read as evidence that the networks rely mainly on colony-level morphological indicators of bleaching.

5.5 Practical Implications for Coral Reef Benchmarking

The findings support three practices for coral imagery collected from video, transects, or repeated site visits. First, split manifests, source-frame identifiers, and duplicate-audit thresholds should be published together with trained models. Second, scene-cluster uncertainty should be reported whenever images originate from correlated sources. Third, architecture comparisons based on a single grouped split should be treated as exploratory unless supported by repeated scene-level resampling or multiple independent sites.

5.6 Limitations of the Evaluation Protocol

The results describe within-collection performance on a single Kaggle collection with whole-scene labels and frames that carry channel logos and caption text, and they come from one grouped split (seed 2024) with 112 test scene groups, which sets the width of the cluster intervals. The partitions are leakage-mitigated and not scene-disjoint, the hash and purge thresholds were not validated against annotated duplicate pairs, no purge-threshold sensitivity analysis was run, and the 165 cross-class near-duplicate pairs were not reviewed manually. The shortcut analysis has no control arm trained without corner and caption dropout and no external dataset free of logos, and the handcrafted-feature baseline was not extended to an ablation that removes color. The corruption contrasts carry no intervals or tests, only four architectures have both augmentation arms, and corruptions held out from training were not evaluated for models trained without the matching family. Tuning effort was comparable and not identical across models, error analysis is limited to confusion matrices and one qualitative example, and the latencies were measured on one GPU type and do not transfer to edge devices. The same encoder families also construct the benchmark and are evaluated as classifiers, so the benchmark composition may depend on the construction encoders, and alternative encoders were not tested.

5.7 Future Work

Future studies should repeat the full pipeline over several grouped split seeds or grouped cross-validation and report the spread of point estimates, rerun the audit with stricter purge thresholds such as 0.90 and report the sensitivity of the ranking, and review the cross-class near-duplicate pairs against an annotated duplicate benchmark. A control arm trained without corner and caption dropout, together with external site-level data free of logos and captions, would test the shortcut question directly. Robustness gains should be tested with paired cluster bootstrap intervals, separately for corruptions seen and unseen in training, and tuning should use an equal number of completed trials or equal GPU time per model. Finally, models should be calibrated with decision thresholds set on data that match the deployment class prior.

6. CONCLUSION

A multi-hash, flip-aware audit showed that 58.1% of the native validation and test images in the Coral Reefs Images collection have a same-class near-duplicate candidate in the training partition under a single average hash, so accuracy on the native splits is not a valid estimate of generalization. Scene-grouped partitioning with a similarity purge produced 1,710 test images from 112 scene groups, and seven architectures trained with three seeds on this split reached 93.0% to 95.6% accuracy and 0.969 to 0.984 ROC-AUC. Scene-cluster intervals were 8 to 12 percentage points wide and only one of 21 architecture pairs was distinguishable, so the data do not support a ranking of the architectures. Noise and blur augmentation raised descriptive accuracy by up to 4.5 percentage points under the degradations it was trained on, at a clean cost of 0.56 percentage points on average. Occlusion tests showed reduced sensitivity to corners and captions in the

robust models, which does not establish the absence of logo or caption shortcuts, and explanation quality differed between models. Residual neighbors found by an independent network show that the reported accuracies are optimistic. For future reef-monitoring studies, the results support auditing partitions for near-duplicates and scene overlap, reporting cluster-aware uncertainty, and avoiding model rankings when intervals overlap substantially. Future work should use repeated grouped splits, stricter purge thresholds with full retraining, a shortcut control arm, external site-level validation on imagery collected without channel logos or captions, and calibration on data that match the deployment class prior.

AUTHOR CONTRIBUTION STATEMENT

All authors contributed equally to the study conception and design. Material preparation, data collection, and analysis were performed by the authors. The first draft of the manuscript was written by the authors, and all authors commented on previous versions of the manuscript. All authors read and approved the final manuscript.

ETHICS APPROVAL AND CONSENT TO PARTICIPATE

Ethics declaration: not applicable. This study did not involve human participants or animals. Therefore, ethical approval and consent to participate are not applicable.

CONSENT FOR PUBLICATION

Not Applicable.

DATA AVAILABILITY

This study uses the publicly available Coral Reefs Images dataset distributed on Kaggle <https://www.kaggle.com/datasets/asfarhossainsitab/coral-reefs-images>.

ACKNOWLEDGMENTS

The authors gratefully acknowledge the use of OpenAI's ChatGPT for assistance with manuscript refinement.

FUNDING

No Funding.

DISCLOSURE STATEMENT

The authors declare no conflict of interest.

REFERENCES

- [1] O. Hoegh-Guldberg, P. J. Mumby, A. J. Hooten, R. S. Steneck, P. Greenfield, E. Gomez, C. D. Harvell, P. F. Sale, A. J. Edwards, K. Caldeira, N. Knowlton, C. M. Eakin, R. Iglesias-Prieto, N. Muthiga, R. H. Bradbury, A. Dubi, and M. E. Hatziolos, "Coral reefs under rapid climate change and ocean acidification," *Science*, vol. 318, no. 5857, pp. 1737–1742, 2007.
- [2] T. P. Hughes, J. T. Kerry, M. Álvarez-Noriega, J. G. Álvarez-Romero, K. D. Anderson, A. H. Baird, R. C. Babcock, M. Beger, D. R. Bellwood, R. Berkelmans, *et al.*, "Global warming and recurrent mass bleaching of corals," *Nature*, vol. 543, no. 7645, pp. 373–377, 2017.

- [3] T. P. Hughes, K. D. Anderson, S. R. Connolly, S. F. Heron, J. T. Kerry, J. M. Lough, A. H. Baird, J. K. Baum, M. L. Berumen, T. C. Bridge, D. C. Claar, C. M. Eakin, J. P. Gilmour, N. A. J. Graham, H. Harrison, J.-P. A. Hobbs, A. S. Hoey, M. Hoogenboom, R. J. Lowe, M. T. McCulloch, J. M. Pandolfi, M. Pratchett, V. Schoepf, G. Torda, and S. K. Wilson, "Spatial and temporal patterns of mass bleaching of corals in the anthropocene," *Science*, vol. 359, no. 6371, pp. 80–83, 2018.
- [4] O. Beijbom, P. J. Edmunds, D. I. Kline, B. G. Mitchell, and D. Kriegman, "Automated annotation of coral reef survey images," in *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1170–1177, 2012.
- [5] O. Beijbom, P. J. Edmunds, C. Roelfsema, J. Smith, D. I. Kline, B. P. Neal, M. J. Dunlap, V. Moriarty, T.-Y. Fan, C.-J. Tan, S. Chan, T. Treibitz, A. Gamst, B. G. Mitchell, and D. Kriegman, "Towards automated annotation of benthic survey images: Variability of human experts and operational modes of automation," *PLoS ONE*, vol. 10, no. 7, p. e0130312, 2015.
- [6] A. Gómez-Ríos, S. Tabik, J. Luengo, A. Shihavuddin, B. Krawczyk, and F. Herrera, "Towards highly accurate coral texture images classification using deep convolutional neural networks and data augmentation," *Expert Systems with Applications*, vol. 118, pp. 315–328, 2019.
- [7] S. Wang, N.-L. Chen, Y.-D. Song, T.-T. Wang, J. Wen, T.-Q. Guo, H.-J. Zhang, L. Mo, H.-R. Ma, and L. Xiang, "MI-net: A multi-local perception network for healthy and bleached coral image classification," *Journal of Marine Science and Engineering*, vol. 12, no. 8, 2024.
- [8] N. K. Gautam, M. Mishra, and U. C. Pati, "Bleaching detection in coral reef using deep convolutional autoencoder based model," in *2024 4th International Conference on Artificial Intelligence and Signal Processing (AISP)*, pp. 01–05, 2024.
- [9] A. A. Ghaffar and G. S. Choi, "A two-stream deep learning framework for robust coral reef health classification: Insights and interpretability," *IEEE Access*, vol. 13, pp. 78490–78512, 2025.
- [10] S. M. Bhanu and K. S. Saravanan, "Coral reef image classification using deep cnns, transformer features, and yolov8," *Marine Systems and Ocean Technology*, vol. 21, p. 6, 2026.
- [11] S. Kaufman, S. Rosset, C. Perlich, and O. Stitelman, "Leakage in data mining: Formulation, detection, and avoidance," *ACM Transactions on Knowledge Discovery from Data (TKDD)*, vol. 6, no. 4, pp. 1–21, 2012.
- [12] S. Kapoor and A. Narayanan, "Leakage and the reproducibility crisis in machine-learning-based science," *Patterns*, vol. 4, no. 9, 2023.
- [13] I. E. Tampu, A. Eklund, and N. Haj-Hosseini, "Inflation of test accuracy due to data leakage in deep learning-based classification of oct images," *Scientific Data*, vol. 9, no. 1, p. 580, 2022.
- [14] B. Barz and J. Denzler, "Do we train on test data? purging cifar of near-duplicates," *Journal of Imaging*, vol. 6, no. 6, 2020.
- [15] R. Geirhos, J.-H. Jacobsen, C. Michaelis, R. Zemel, W. Brendel, M. Bethge, and F. A. Wichmann, "Shortcut learning in deep neural networks," *Nature Machine Intelligence*, vol. 2, no. 11, pp. 665–673, 2020.
- [16] Asfar Hossain Sitab, "Coral reefs images." Kaggle dataset.
- [17] X. Shao, H. Chen, K. Magson, J. Wang, J. Song, J. Chen, and J. Sasaki, "Deep learning for multilabel classification of coral reef conditions in the indo-pacific using underwater photo transect method," *Aquatic Conservation: Marine and Freshwater Ecosystems*, vol. 34, no. 9, p. e4241, 2024. e4241 AQC-24-0036.R1.
- [18] X. Zhao, L. Wang, Y. Zhang, *et al.*, "A review of convolutional neural networks in computer vision," *Artificial Intelligence Review*, vol. 57, p. 99, 2024.
- [19] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16x16 words: Transformers for image recognition at scale," in *Proceedings of the International Conference on Learning Representations (ICLR)*, 2021.
- [20] K. Han, Y. Wang, H. Chen, X. Chen, J. Guo, Z. Liu, Y. Tang, A. Xiao, C. Xu, Y. Xu, Z. Yang, Y. Zhang, and D. Tao, "A survey on vision transformer," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 1, pp. 87–110, 2023.
- [21] S. Khan, M. Naseer, M. Hayat, S. W. Zamir, F. S. Khan, and M. Shah, "Transformers in vision: A survey," *ACM computing surveys (CSUR)*, vol. 54, no. 10s, pp. 1–41, 2022.

- [22] M. Raghu, T. Unterthiner, S. Kornblith, C. Zhang, and A. Dosovitskiy, “Do vision transformers see like convolutional neural networks?,” *Advances in neural information processing systems*, vol. 34, pp. 12116–12128, 2021.
- [23] L. Nanni, A. Lumini, L. Barcellona, and S. Ghidoni, “Convolutional neural networks and vision transformers for plankton classification,” *Ecological Informatics*, vol. 90, p. 103272, 2025.
- [24] Z. Liu, H. Mao, C.-Y. Wu, C. Feichtenhofer, T. Darrell, and S. Xie, “A convnet for the 2020s,” in *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11966–11976, 2022.
- [25] Z. Tu, H. Talebi, H. Zhang, F. Yang, P. Milanfar, A. Bovik, and Y. Li, “Maxvit: Multi-axis vision transformer,” in *Computer Vision – ECCV 2022* (S. Avidan, G. Brostow, M. Cissé, G. M. Farinella, and T. Hassner, eds.), (Cham), pp. 459–479, Springer Nature Switzerland, 2022.
- [26] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, et al., “Imagenet large scale visual recognition challenge,” *International journal of computer vision*, vol. 115, no. 3, pp. 211–252, 2015.
- [27] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, “How transferable are features in deep neural networks?,” in *Advances in Neural Information Processing Systems* (Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Weinberger, eds.), vol. 27, pp. 3320–3328, Curran Associates, Inc., 2014.
- [28] S. Kornblith, J. Shlens, and Q. V. Le, “Do better imagenet models transfer better?,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 2661–2671, 2019.
- [29] A. Kumar, A. Raghunathan, R. M. Jones, T. Ma, and P. Liang, “Fine-tuning can distort pretrained features and underperform out-of-distribution,” in *International Conference on Learning Representations*, 2022.
- [30] V. Monga and B. Evans, “Perceptual image hashing via feature points: Performance evaluation and tradeoffs,” *IEEE Transactions on Image Processing*, vol. 15, no. 11, pp. 3452–3465, 2006.
- [31] C. Zauner, “Implementation and benchmarking of perceptual image hash functions,” master’s thesis, Upper Austria University of Applied Sciences, Hagenberg Campus, 2010.
- [32] Q. McNemar, “Note on the sampling error of the difference between correlated proportions or percentages,” *Psychometrika*, vol. 12, no. 2, p. 153–157, 1947.
- [33] T. G. Dietterich, “Approximate statistical tests for comparing supervised classification learning algorithms,” *Neural Computation*, vol. 10, pp. 1895–1923, 10 1998.
- [34] A. C. Cameron, J. B. Gelbach, and D. L. Miller, “Bootstrap-based improvements for inference with clustered errors,” *The Review of Economics and Statistics*, vol. 90, pp. 414–427, 08 2008.
- [35] C. A. Field and A. H. Welsh, “Bootstrapping clustered data,” *Journal of the Royal Statistical Society Series B: Statistical Methodology*, vol. 69, no. 3, pp. 369–390, 2007.
- [36] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, “Grad-cam: Visual explanations from deep networks via gradient-based localization,” in *Proceedings of the IEEE international conference on computer vision*, pp. 618–626, 2017.
- [37] S. Abnar and W. Zuidema, “Quantifying attention flow in transformers,” in *Proceedings of the 58th annual meeting of the association for computational linguistics*, pp. 4190–4197, 2020.
- [38] H. Chefer, S. Gur, and L. Wolf, “Transformer interpretability beyond attention visualization,” in *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 782–791, 2021.
- [39] J. Adebayo, J. Gilmer, M. Muelly, I. Goodfellow, M. Hardt, and B. Kim, “Sanity checks for saliency maps,” in *Advances in Neural Information Processing Systems* (S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, eds.), vol. 31, Curran Associates, Inc., 2018.
- [40] V. Petsiuk, A. Das, and K. Saenko, “RISE: Randomized input sampling for explanation of black-box models,” in *Proceedings of the British Machine Vision Conference (BMVC)*, BMVA Press, 2018.
- [41] D. Hendrycks and T. Dietterich, “Benchmarking neural network robustness to common corruptions and perturbations,” in *International Conference on Learning Representations*, 2019.
- [42] D. Akkaynak and T. Treibitz, “A revised underwater image formation model,” in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6723–6732, 2018.

- [43] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and É. Duchesnay, “Scikit-learn: Machine learning in python,” *Journal of Machine Learning Research*, vol. 12, no. 85, pp. 2825–2830, 2011.
- [44] K. He, J. Sun, and X. Tang, “Single image haze removal using dark channel prior,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 12, pp. 2341–2353, 2011.
- [45] R. Wightman, “Pytorch image models.” <https://github.com/rwightman/pytorch-image-models>, 2019.
- [46] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2016.
- [47] M. Tan and Q. V. Le, “EfficientNet: Rethinking model scaling for convolutional neural networks,” in *Proceedings of the 36th International Conference on Machine Learning (ICML)*, pp. 6105–6114, 2019.
- [48] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, “Mobilenetv2: Inverted residuals and linear bottlenecks,” in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4510–4520, 2018.
- [49] J. Hu, L. Shen, and G. Sun, “Squeeze-and-excitation networks,” in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7132–7141, 2018.
- [50] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin, “Attention is all you need,” in *Advances in Neural Information Processing Systems* (I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, eds.), vol. 30, Curran Associates, Inc., 2017.
- [51] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, “Swin Transformer: Hierarchical Vision Transformer using Shifted Windows,” in *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, (Los Alamitos, CA, USA), pp. 9992–10002, IEEE Computer Society, Oct. 2021.
- [52] I. Loshchilov and F. Hutter, “Decoupled weight decay regularization,” in *International Conference on Learning Representations (ICLR)*, 2019.
- [53] I. Loshchilov and F. Hutter, “Sgdr: Stochastic gradient descent with warm restarts,” *arXiv preprint arXiv:1608.03983*, 2016.
- [54] R. Pascanu, T. Mikolov, and Y. Bengio, “On the difficulty of training recurrent neural networks,” in *Proceedings of the 30th International Conference on Machine Learning* (S. Dasgupta and D. McAllester, eds.), vol. 28 of *Proceedings of Machine Learning Research*, (Atlanta, Georgia, USA), pp. 1310–1318, PMLR, June 2013.
- [55] P. Micikevicius, S. Narang, J. Alben, G. Diamos, E. Elsen, D. Garcia, B. Ginsburg, M. Houston, O. Kuchaiev, G. Venkatesh, and H. Wu, “Mixed precision training,” in *International Conference on Learning Representations*, 2018.
- [56] B. T. Polyak and A. B. Juditsky, “Acceleration of stochastic approximation by averaging,” *SIAM journal on control and optimization*, vol. 30, no. 4, pp. 838–855, 1992.
- [57] A. Tarvainen and H. Valpola, “Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results,” in *Advances in Neural Information Processing Systems* (I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, eds.), vol. 30, Curran Associates, Inc., 2017.
- [58] S. Ioffe and C. Szegedy, “Batch normalization: Accelerating deep network training by reducing internal covariate shift,” in *International conference on machine learning*, pp. 448–456, pmlr, 2015.
- [59] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, “Rethinking the inception architecture for computer vision,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2818–2826, June 2016.
- [60] R. Müller, S. Kornblith, and G. E. Hinton, “When does label smoothing help?,” in *Advances in Neural Information Processing Systems* (H. Wallach, H. Larochelle, A. Beygelzimer, F. d’Alché-Buc, E. Fox, and R. Garnett, eds.), vol. 32, Curran Associates, Inc., 2019.
- [61] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, “mixup: Beyond empirical risk minimization,” in *International Conference on Learning Representations*, 2018.
- [62] 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 *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019.

- [63] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, “Pytorch: An imperative style, high-performance deep learning library,” in *Advances in Neural Information Processing Systems* (H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, eds.), vol. 32, Curran Associates, Inc., 2019.
- [64] T. Akiba, S. Sano, T. Yanase, T. Ohta, and M. Koyama, “Optuna: A next-generation hyperparameter optimization framework,” in *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, pp. 2623–2631, 2019.
- [65] J. Bergstra, R. Bardenet, Y. Bengio, and B. Kégl, “Algorithms for hyper-parameter optimization,” in *Advances in Neural Information Processing Systems* (J. Shawe-Taylor, R. Zemel, P. Bartlett, F. Pereira, and K. Weinberger, eds.), vol. 24, Curran Associates, Inc., 2011.
- [66] L. Li, K. Jamieson, G. DeSalvo, A. Rostamizadeh, and A. Talwalkar, “Hyperband: A novel bandit-based approach to hyperparameter optimization,” *Journal of Machine Learning Research*, vol. 18, no. 185, pp. 1–52, 2018.
- [67] G. Huang, Y. Sun, Z. Liu, D. Sedra, and K. Q. Weinberger, “Deep networks with stochastic depth,” in *Computer Vision – ECCV 2016* (B. Leibe, J. Matas, N. Sebe, and M. Welling, eds.), (Cham), pp. 646–661, Springer International Publishing, 2016.
- [68] S. Pizer, E. Amburn, J. Austin, R. Cromartie, A. Geselowitz, T. Greer, B. {Haar Romenij, ter}, J. Zimmerman, and K. Zuiderveld, “Adaptive histogram equalization and its variations,” *Computer Vision, Graphics, and Image Processing*, vol. 39, no. 3, pp. 355–368, 1987.
- [69] K. Zuiderveld, “Contrast limited adaptive histogram equalization,” in *Graphics gems IV*, pp. 474–485, Boston: Academic Press, 1994.
- [70] T. Fawcett, “An introduction to roc analysis,” *Pattern Recognition Letters*, vol. 27, no. 8, pp. 861–874, 2006. ROC Analysis in Pattern Recognition.
- [71] T. Saito and M. Rehmsmeier, “The precision-recall plot is more informative than the roc plot when evaluating binary classifiers on imbalanced datasets,” *PloS one*, vol. 10, no. 3, p. e0118432, 2015.
- [72] M. Sokolova and G. Lapalme, “A systematic analysis of performance measures for classification tasks,” *Information Processing & Management*, vol. 45, no. 4, pp. 427–437, 2009.
- [73] C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger, “On calibration of modern neural networks,” in *International conference on machine learning*, pp. 1321–1330, PMLR, 2017.
- [74] D. H. Wolpert, “Stacked generalization,” *Neural Networks*, vol. 5, no. 2, pp. 241–259, 1992.
- [75] B. Efron, *Bootstrap Methods: Another Look at the Jackknife*, pp. 569–593. New York, NY: Springer New York, 1992.
- [76] P. Good, *Testing Hypotheses*, pp. 33–65. New York, NY: Springer, 2005.
- [77] B. Phipson and G. K. Smyth, “Permutation P-values should never be zero: calculating exact P-values when permutations are randomly drawn,” *Statistical Applications in Genetics and Molecular Biology*, vol. 9, no. 1, p. Article 39, 2010.
- [78] S. Holm, “A simple sequentially rejective multiple test procedure,” *Scandinavian Journal of Statistics*, vol. 6, no. 2, pp. 65–70, 1979.
- [79] M. D. Zeiler and R. Fergus, “Visualizing and understanding convolutional networks,” in *Computer Vision – ECCV 2014* (D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, eds.), (Cham), pp. 818–833, Springer International Publishing, 2014.
- [80] P. J. Rousseeuw, “Silhouettes: A graphical aid to the interpretation and validation of cluster analysis,” *Journal of Computational and Applied Mathematics*, vol. 20, pp. 53–65, 1987.