EURASIAN JOURNAL OF MATHEMATICAL AND COMPUTER APPLICATIONS
ISSN 2306–6172
Volume 13, Issue 4 (2025) 128 – 140

VALF: VALIDATION-ADAPTIVE FOCAL LOSS FOR HISTOPATHOLOGY

Kaveh V., Amirreza J., Hedieh S.[1], Ardavan D., Azamat A.,

Abstract Medical image classification often fails for two reasons. Rare but clinically important categories create class imbalance. Similarity between classes also makes some diagnoses hard to separate without a loss that focuses on fine patterns. We introduce *Validation Adaptive Focal Loss* (VALF), a plug and play objective that augments focal loss with per class weights that are initialized uniformly or from a user provided prior and that are adapted during training based on validation feedback. We keep the weights fixed for the initial part of training, then update them after each epoch using per class validation accuracy. We apply a small multiplicative change and then renormalize the mean weight. The loss is class weight times focal factor times cross entropy. VALF needs no architectural changes, no auxiliary network, and no multi stage training schedule. On LungHist700 at 20× and 40× across five backbones, VALF attains the top macro F1 in 8 of 10 settings and yields consistent gains in accuracy, precision, and recall. The largest macro F1 improvement is about +4.0% over the best baseline at 40×. Improvements are robust across models and magnifications, with only minor shortfalls in two 20× cases. These results indicate that simple validation driven and class aware weighting can balance sensitivity and specificity and can serve as a practical drop in for clinical pipelines.

Keywords: Validation Adaptive Focal Loss (VALF), focal loss, class imbalance, validation driven reweighting, medical image classification, histopathology, lung cancer, LungHist700, data science, artificial intelligence, machine learning, big data, applied AI, medical informatics.

AMS Mathematics Subject Classification: 68T07, 68T45.

DOI: 10.32523/2306-6172-2025-13-4-128-140

1 Introduction

Class imbalance is pervasive in medical image analysis, where structures of interest can vary greatly in size; the *Generalized Dice overlap* was proposed as "a deep learning loss function [that] is particularly suited to unbalanced problems" and was shown to outperform weighted cross entropy under severe imbalance [11]. At the same time, "the focal loss reshapes the cross entropy loss function with a modulating exponent to down weight errors assigned to well classified examples," preventing easy negatives from dominating, yet it "faces difficulty balancing precision and recall due to small regions of interest (ROI) found in medical images" [12]. Building on both cross entropy and Dice based criteria, the *Unified Focal loss* "generalises Dice based and cross entropy based loss functions into a single framework" and clarifies that these losses are special cases, aiming to be "robust to both input and output imbalances" without significantly increasing training time [11]. Beyond vanilla cross entropy, determining class weights is itself challenging: "the ideal setting of class weight parameters is a challenge"; inverse frequency heuristics "may not always reliably produce the most accurate semantic segmentation results," motivating strategies that "identify 'hard-to-classify' examples and assign greater weights to them" [14]. The focal loss was introduced for imbalance in dense detection and "aims to balance the sample wise classification loss by down weighting easy samples" via

[1]Corresponding Author.

a reweighting factor; in its canonical form

$$\mathcal{L}_{\text{focal}} \;=\; (1-h_i)^{\gamma}\,\mathcal{L}_{\text{CE}} \;=\; -(1-h_i)^{\gamma}\,\log(h_i), \tag{1}$$

where h_i is the predicted probability of the true class and $\gamma \geq 0$ controls how much easy examples are down weighted [2, 4]. To further encode dataset frequency, the *Class Balanced Loss* proposes reweighting by the *effective number of samples*, $E_n = (1-\beta^n)/(1-\beta)$, yielding per class factors $(1-\beta)/(1-\beta^{n_j})$ that improve long tailed recognition [1]. Another principled approach is *logit adjustment*, which "adds $\tau \log \pi_y$ to the logits before softmax" to correct for label frequency induced bias during training [8]. Orthogonal to loss shaping, *decoupled training* argues that "imbalanced training affects representation learning and classifier learning in different ways" and thus separates them into balanced classifier learning and representation learning stages [2]. Mixup based remedies have also been explored; for instance, *Balanced MixUp* "performs mixup with a balanced sampling strategy (instead of the standard one)" and surveys that, despite various resampling and objective designs, "model performance on tail classes remains to be improved" [7]. Validation driven and meta learned reweighting offer another line of evidence. *Learning to Reweight Examples* uses "a small but clean validation set" to "meta-learn a robust sample reweighting approach" that assigns higher weight to training examples that reduce validation loss [9]. *Meta Weight Net* further "learns an explicit mapping for sample weighting," parameterizing a weight function by a meta network. [10] Finally, calibration focused work shows that training with focal loss "leads to better calibration than cross entropy while achieving similar level of accuracy," and *AdaFocal* adaptively modifies γ using validation bin statistics "switching from focal to inverse focal loss when focal loss fails to overcome under confidence" to maintain low calibration error across probability regions [13].

2 Related Work

2.1 Overview

Class imbalance significantly impacts deep learning tasks, particularly medical image classification, where minority classes are critical yet often overshadowed by dominant classes. Traditional methods to address imbalance typically fall into two main categories: data level techniques, such as oversampling and undersampling, and algorithm level adjustments, including specialized loss functions and weighting mechanisms [1, 2]. Oversampling may lead to overfitting, while undersampling risks the loss of valuable information [13]. Algorithmic strategies frequently involve modifying loss functions to better accommodate minority classes during optimization [3].

2.2 Popular Loss Functions

Categorical Cross Entropy (CCE). Categorical Cross Entropy (CCE) is widely employed as the fundamental loss function in multiclass classification. It quantifies the difference between the predicted probability distribution and the actual distribution:

$$\text{CCE}(p_t) = -\log(p_t) \tag{2}$$

where p_t represents the predicted probability of the true class. Although prevalent, CCE does not inherently address class imbalance, thus motivating extensions that balance attention across all classes [3].

Weighted Cross Entropy (WCCE). Weighted Cross Entropy extends CCE by introducing per class weights to better handle class imbalance:

$$\text{WCCE} = -\sum_i w_i \, y_i \log(\hat{y}_i) \tag{3}$$

Here w_i is the class weight (often inverse frequency), $y_i \in \{0, 1\}$ indicates the true class, and $\hat{y}_i$ is the predicted probability for class i [5].

Focal Loss (FL). Focal Loss modifies standard cross entropy by adding a modulating factor, $(1 - p_t)^\gamma$, to reduce the contribution of well classified examples and emphasize harder instances:

$$\text{FL}(p_t) = -\big(1 - p_t\big)^\gamma \log(p_t) \tag{4}$$

Here p_t is the predicted probability of the true class, and $\gamma \geq 0$ controls the degree of down weighting. Originally proposed for dense detection tasks, it effectively tackles imbalance by shifting training focus toward challenging examples [4].

Dynamic Weighted Focal Loss (DWFL). Dynamic Weighted Focal Loss enhances FL by adaptively modifying class specific weights throughout training. These adjustments dynamically rebalance the learning process according to model performance on imbalanced datasets, particularly beneficial in medical image scenarios [6].

2.3 Similar Works

Class Balanced and Decoupling Strategies. Class Balanced Loss employs weighting based on the effective number of samples, a principled alternative to simple inverse frequency weighting, particularly suited to long-tailed datasets [1]. Logit Adjustment introduces class prior corrections directly into logits, adjusting decision boundaries in long tailed learning scenarios [8]. Decoupled training frameworks further address imbalance by separating feature learning from classifier retraining phases, notably improving performance in highly imbalanced classification tasks [2]. At the data level, Balanced MixUp adjusts mixup based data augmentation, effectively promoting generalization to minority classes [7].

Meta Weighting and Adaptive Techniques. Meta learning methods, such as Learning to Reweight Examples, dynamically assign weights to training samples based on validation set performance, optimizing generalization under noisy labels and imbalance [9]. Meta Weight Net further refines this idea by explicitly learning a mapping from sample losses to weights, offering an adaptive and learnable reweighting scheme [10].

Extended Focal Loss Variants. Unified Focal Loss generalizes FL by combining Dice and cross entropy formulations, enhancing segmentation tasks under probabilistic modeling frameworks [11]. Focal Tversky Loss similarly extends focal mechanisms to segmentation, effectively capturing small or sparse regions by modifying the Tversky index [12]. Adaptive methods, such as AdaFocal, dynamically adjust focal parameters during training to achieve better calibration and robustness under distribution shifts [13]. Additionally, the Adaptive Class Weight based Dual Focal Loss introduces an adaptive class weight layer and evaluates both positive and negative classes to comprehensively mitigate imbalance in segmentation tasks [14].

3 Method

3.1 Motivation

Class imbalance in a dataset may arise from several sources, such as differences in the number of samples per class or variations in intra class diversity. To address this issue, we allow the model to first train without any weight modification, ensuring that no class is emphasized during the initial warmup phase. After this period, we evaluate classwise validation accuracies, which determines which classes are easier to learn (i.e., achieving higher accuracy) and which remain underrepresented or harder to learn. At this point, we begin adaptively increasing the weights of the harder classes to direct more learning capacity toward them. This adjustment is applied gradually to avoid instability and to maintain balanced optimization dynamics.

3.2 Validation-Adaptive Focal Loss

Let C denote the number of classes. For a sample with one-hot label $\mathbf{y} \in \{0,1\}^C$ and predicted class probabilities $\hat{\mathbf{y}} \in [0,1]^C$ (softmax outputs; `from_logits=False`), define

$$p_t \;=\; \sum_{c=1}^{C} y_c\, \hat{y}_c \tag{5}$$

Here $y_c \in \{0,1\}$ is the one hot label and $\hat{y}_c$ is the predicted probability for class c; thus p_t is the predicted probability of the true class. Categorical cross-entropy for that sample is $\mathrm{CE}(\mathbf{y}, \hat{\mathbf{y}}) = -\log p_t$. Let $\mathbf{w} \in \mathbb{R}_+^C$ be the non-trainable vector of per-class weights and $w_{c^*} = \sum_c y_c w_c$ the weight of the ground-truth class c^*. With focusing parameter $\gamma \geq 0$, the proposed loss used in training is

$$\mathcal{L}_{\mathrm{VALF}} \;=\; \frac{1}{N} \sum_{n=1}^{N} \Big(w_{c^*(n)}\Big) \Big(1 - p_t^{(n)}\Big)^{\gamma} \Big(-\log p_t^{(n)}\Big), \tag{6}$$

which corresponds exactly to the implementation

$$\texttt{loss} \;=\; \underbrace{\texttt{class\_weight}}_{w_{c^*}} \times \underbrace{(1-p_t)^{\gamma}}_{\texttt{focal\_factor}} \times \underbrace{\mathrm{CE}(\mathbf{y}, \hat{\mathbf{y}})}_{-\log p_t}.$$

When $\gamma = 0$ and $w_c \equiv 1$, (6) reduces to standard categorical cross-entropy; with $w_c \not\equiv 1$ it reduces to weighted cross-entropy; with $\gamma > 0$ and $w_c \not\equiv 1$ it recovers the weighted focal variant used here.

3.3 Initialization and Adaptive Update

Initialization. Class weights are initialized *uniformly* by default ($w_c^{(0)} = 1$ for all c). Optionally, a user-provided prior (list or dictionary keyed by class index) can be supplied at construction time; these values are stored in a non-trainable TensorFlow variable.

Warm-up and update cadence. Weights remain fixed for a warm-up of E_{warm} epochs (default 10). After warm-up, weights are *updated at the end of every epoch* (the optional periodic updater is disabled in code).

Table 1: Training-time input/augmentation settings (`albumentations`); probabilities and parameters mirror the implementation.

Transform	**Key parameters**	p
`HorizontalFlip`	–	0.5
`VerticalFlip`	–	0.5
`GridDistortion`	default grid (mild elastic warp)	0.2
`RandomSizedCrop`	min_max_height $(1000, 1200)$; out 1200×1600	0.4
`RandomGamma`	$\gamma \in [80, 120]$	0.5
`RandomBrightnessContrast`	brightness/contrast ± 0.2	0.2
`HueSaturationValue`	hue ± 5, sat ± 20, val ± 10	0.2
`Resize`	1200×1600 $\to (1200 \cdot r)\times(1600 \cdot r)$; r=0.25	–
`ToFloat`	scale pixel range to $[0, 1]$	–

Validation-driven rule (per-class accuracy). At epoch $t \geq E_{\text{warm}}$, compute per-class validation accuracies $a_c^{(t)}$ by argmax predictions on the held-out validation set (classes with no validation samples receive $a_c^{(t)} = 0$ in the implementation). Let $\bar{a}^{(t)} = \frac{1}{C}\sum_{c=1}^{C} a_c^{(t)}$. The unnormalized update is multiplicative:

$$\tilde{w}_c^{(t+1)} \;=\; w_c^{(t)} \left(1 + \frac{\bar{a}^{(t)} - a_c^{(t)}}{100}\right). \tag{7}$$

Thus, underperforming classes ($a_c^{(t)} < \bar{a}^{(t)}$) receive a slight increase; overperforming classes a slight decrease.

Mean normalization for stability. After applying (7), the weights are renormalized to keep the mean at 1:

$$w_c^{(t+1)} \;=\; \frac{\tilde{w}_c^{(t+1)}}{\frac{1}{C}\sum_{k=1}^{C} \tilde{w}_k^{(t+1)}}. \tag{8}$$

These updated weights are then used by (6) in the next epoch.

4 Experimental Setup

4.1 Dataset

We use the LungHist700 dataset, which contains 691 H&E histopathology images at 1200×1600 resolution from 45 patients, captured at 20× and 40× and released as `.jpg` files [21]. In the original release, images are annotated into seven subclasses (normal tissue; and malignant tissue subdivided by differentiation level, moderately, and poorly differentiated for both adenocarcinoma and squamous cell carcinoma) [21]. Following the dataset paperвЂ™s experimental protocol, we group these into three superclasses adenocarcinoma (ACA), squamous cell carcinoma (SCC), and normal (NOR). For dataset split, we used the split that was used in the paper's code available on Github to ensure a fair comparison. The release also includes a `.csv` mapping each image to a patient identifier [21].

The class distribution across our train/validation/test sets is shown in Fig. 1, and the overall distribution for ACA/NOR/SCC is shown in Fig. 2. Figure 3 provides sample tiles for the three superclasses at both magnifications, where the top row shows 20× images and the bottom row shows 40× images.

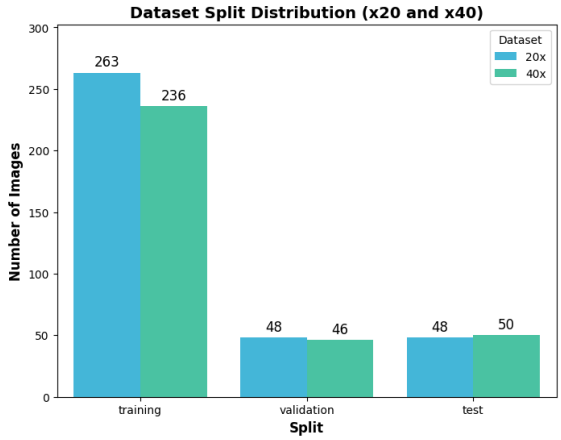

Figure 1: Data distribution of the LungHist700 dataset 20x and 40x across training, validation, and test splits.

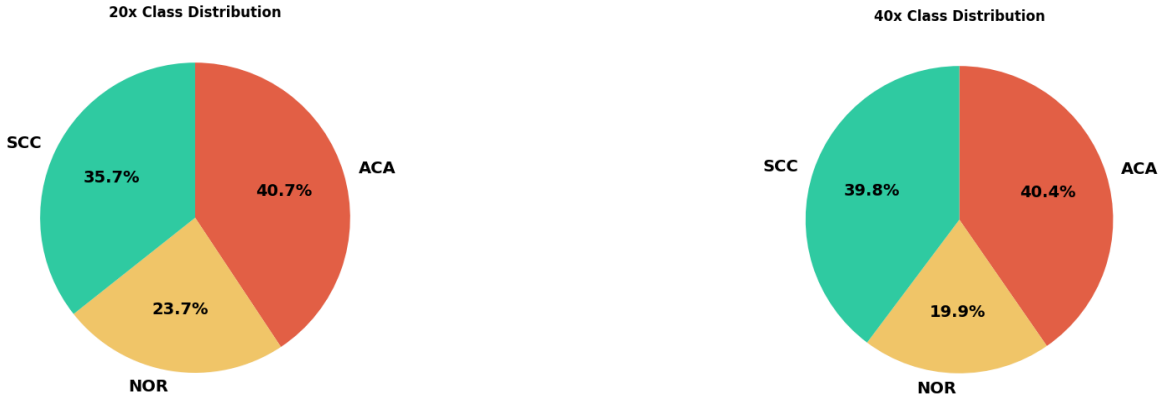

Figure 2: Overall class distribution (ACA, NOR, SCC) in LungHist700 dataset 20x and 40x.

4.2 Preprocessing and Data Augmentation

We standardize training inputs with a compact `albumentations` pipeline. The goal is to increase invariance to orientation, elastic slide artifacts, and stain or illumination shifts while preserving diagnostic morphology. As summarized in Table 1, each image (originally 1200×1600) undergoes random flips and mild grid distortion for geometric diversity; a `RandomSizedCrop` (to 1200×1600) to vary effective field of view; and light photometric jitter (`RandomGamma`, `RandomBrightnessContrast`, `HueSaturationValue`) to model acquisition/stain shifts. After augmentation, images are resized to 300×400 (i.e., 25% of the original; controlled by `percent_resize`) and then normalized to the $[0, 1]$ range via `ToFloat(max_value=255)`. Validation/test images receive only the final resize and the same $[0, 1]$ normalization. Following the LungHist700 paper, we reuse the authorsвЂ™ augmentation recipe and keep its transformations and hyperparameters unchanged to enable fair, like-for-like comparison with their reported results [21].

4.3 Backbone Architectures

We compare five widely used convolutional backbones under identical training protocols.

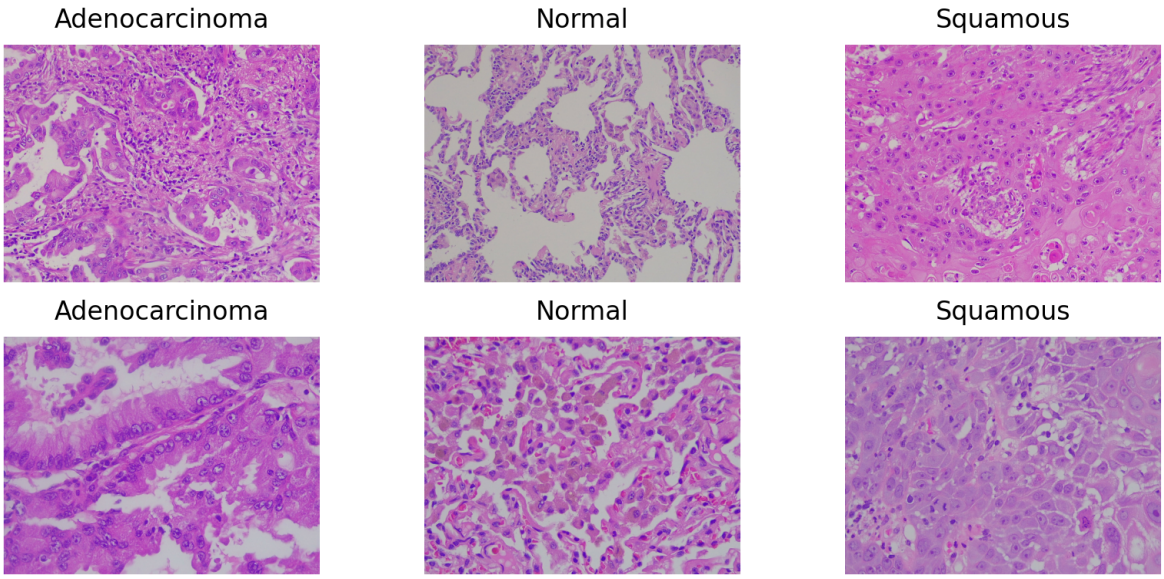

Figure 3: Sample images of each class at two magnifications: the top row shows 20× images and the bottom row shows 40× images.

ResNet. He *et al.* "present a residual learning framework to ease the training of networks that are substantially deeper than those used previously," reformulating layers "as learning residual functions with reference to the layer inputs." They report ImageNet models "with a depth of up to 152 layers" and an ensemble achieving "3.57% error on the ImageNet test set." [15]

DenseNet. DenseNet "connects each layer to every other layer in a feed-forward fashion," so that "for each layer, the feature-maps of all preceding layers are used as inputs, and its own feature-maps are used as inputs into all subsequent layers." This design "alleviate[s] the vanishing-gradient problem, strengthen[s] feature propagation, encourage[s] feature reuse, and substantially reduce[s] the number of parameters." [16] In line with this, a study on skin lesion classification reports that DenseNet 121 achieves superior validation performance compared with EfficientNet, ResNet 50, VGG16, GoogleNet, and MobileNet V3 Large. [17] The comparison covers accuracy, macro F1, G Means, and MCC on a dataset with class imbalance. Accordingly, we also evaluate our losses on a DenseNet 121 backbone.

Inception (v3). Szegedy *et al.* explore "ways to scale up networks in ways that aim at utilizing the added computation as efficiently as possible by suitably factorized convolutions and aggressive regularization," reporting "substantial gains over the state of the art: 21.2% top-1 and 5.6% top-5 error . . . using less than 25 million parameters." [18]

EfficientNetV2. EfficientNetV2 is introduced as "a new family of convolutional networks that have faster training speed and better parameter efficiency than previous models," developed via "training-aware neural architecture search and scaling," together with an "improved method of progressive learning, which adaptively adjusts regularization . . . along with image size." The paper concludes that EfficientNetV2 "trains up to 11x faster while being up to 6.8x smaller." [19]

ConvNeXt. Liu *et al.* "reexamine the design spaces and test the limits of what a pure ConvNet can achieve," gradually "moderniz[ing] a standard ResNet toward the design of a vision Transformer." The resulting "pure ConvNet models dubbed ConvNeXt ... compete favorably with Transformers in terms of accuracy and scalability," with results highlighting performance on ImageNet and downstream detection/segmentation benchmarks. [20]

4.4 Implementation Details

All experiments were run on a local workstation with an NVIDIA GeForce GTX 1660 Ti GPU using TensorFlow/Keras 2.15.0. To ensure a fair comparison across losses, the main hyperparameters learning rate, batch size, callbacks/schedulers, and the overall training protocol were kept identical for each backbone (see §4.5). We used the *official* LungHist700 train/validation/test split at both 20× and 40× magnifications and followed the datasetвЂ™s augmentation recipe as described in §4.2 to maintain comparability with the dataset paper.

We evaluated five widely used convolutional backbones-ResNet50, EfficientNetV2B3, DenseNet121, ConvNeXt and InceptionV3-initialized with ImageNet pretraining. Each backbone was trained under the same settings with four loss formulations: categorical cross entropy (CCE), weighted CCE (WCCE), focal loss (FL), and the proposed VALF; where applicable, we also include Dynamic Weighted Focal Loss (DWFL) as a focal-variant baseline.

Code availability. Our TensorFlow implementation of VALF, training/evaluation scripts, and experiment configurations are publicly available at `VALF`[2].

4.5 Training Hyperparameters

All models are trained for *up to* 100 epochs with a mini-batch size of 4. We used Adam and report the requested metrics per epoch. For learning rate, we ran each backbone with each loss using both 10^{-4} and 10^{-5} and then chose the one that on average performed better according to accuracy and used it for all the losses to achieve a fair comparison. To avoid overfitting and to adapt the learning rate, we enable EarlyStopping on validation loss with patience = 25, a ReduceLROnPlateau scheduler with patience = 10 and min_lr = 1×10^{-7}, and a ModelCheckpoint that keeps the best weights by validation loss (plus a CSVLogger for epoch-wise traces) excatly like the dataset paper. For VALF. class weights are held fixed for a 10-epoch warm-up and then *updated at the end of every epoch* using validation performance; the updated weights are mean-normalized and fed into the next epoch.

4.6 Evaluation Metrics

Given the class imbalance in LungHist700, we evaluate the model in a one-vs-rest (OvR) fashion for each class and report *macro-averaged* scores unless otherwise noted. Let C be the number of classes and, for class $c \in \{1, \ldots, C\}$, let TP_c, FP_c, FN_c, TN_c denote true positives, false positives, false negatives, and true negatives, respectively.

Accuracy

$$\text{Accuracy} \;=\; \frac{\sum_{c=1}^{C}(TP_c + TN_c)}{\sum_{c=1}^{C}(TP_c + TN_c + FP_c + FN_c)}. \tag{9}$$

Role: Overall correctness across all classes. Useful for a quick sanity check, but can overestimate performance under imbalance because majority classes dominate the denominator.

[2]Code: `https://github.com/kavehvajedsamie/VALF`

Precision (macro)

$$\text{Precision}_{\text{macro}} = \frac{1}{C}\sum_{c=1}^{C}\frac{TP_c}{TP_c + FP_c}. \tag{10}$$

Role: Fraction of predicted positives that are correct. Penalizes false positives; clinically, higher precision reduces over-diagnosis (e.g., fewer normal tissues flagged as cancer).

Recall / Sensitivity (macro)

$$\text{Recall}_{\text{macro}} = \frac{1}{C}\sum_{c=1}^{C}\frac{TP_c}{TP_c + FN_c}. \tag{11}$$

Role: Ability to find true positives. Penalizes false negatives; critical in medical settings to avoid missed cancer cases, especially for minority classes.

F1-score (macro)

$$\text{F1}_{\text{macro}} = \frac{1}{C}\sum_{c=1}^{C}\frac{2\cdot \text{Precision}_c \cdot \text{Recall}_c}{\text{Precision}_c + \text{Recall}_c}, \tag{12}$$

where $\text{Precision}_c = \frac{TP_c}{TP_c+FP_c}$ and $\text{Recall}_c = \frac{TP_c}{TP_c+FN_c}$. *Role:* Harmonic mean of precision and recall, balancing over- and under-diagnosis. More informative than accuracy under imbalance and sensitive to improvements VALF targets.

5 Results

5.1 Overall Comparison Across Backbones and Magnifications

Table 2 reports accuracy, precision, recall, and macro-F1 at 20× and 40× across five backbones and *four baselines* (WCCE, CCE, FL, DWFL) *plus VALF*. VALF attains the highest macro-F1 in most settings (8/10 configurations). Specifically, for ResNet50 it reaches **0.939** at 20× (best baseline FL: 0.925, Δ=+0.014) and **0.875** at 40× (best baseline FL: 0.855, Δ=+0.020). For EfficientNet, VALF yields **0.825** at 20× (best baseline CCE: 0.796, Δ=+0.029) and **0.945** at 40× (best baseline FL: 0.905, $\Delta = +0.040$). For DenseNet121, VALF leads at 40× with **0.940** (best baseline FL: 0.925, $\Delta = +0.015$) but is slightly behind at 20× where WCCE is best (**0.833** vs. VALF 0.825). For InceptionV3 at 20×, **DWFL** is strongest (**0.905**); at 40× VALF leads with **0.909**. Finally, for ConvNeXt, VALF attains **0.831** at 20× (best baseline CCE: 0.810, $\Delta = +0.021$) and **0.890** at 40× (CCE: 0.889).

5.2 Effect of Magnification

VALF shows its largest gains at 40×. EfficientNet reaches F1 = 0.945 (best baseline FL: 0.905, $\Delta = +0.040$). DenseNet121 reaches 0.940 (best baseline FL: 0.925, $\Delta = +0.015$). InceptionV3 reaches 0.909 (best baseline CCE: 0.884, $\Delta = +0.025$). ResNet50 peaks at 20× with **F1 = 0.939** (best baseline FL: 0.925, $\Delta = +0.014$), though VALF still leads at 40× (**0.875** vs. FL 0.855, $\Delta = +0.020$). For ConvNeXt, VALF is competitive at both scales, reaching **0.831** at 20× (vs. CCE 0.810) and **0.890** at 40× (vs. CCE 0.889).

Table 2: Performance on LungHist700 Test Set with Multiple Magnifications (20× and 40×)

Model	**Loss**	**20×**				**40×**			
		Acc	Prec	Rec	F1	Acc	Prec	Rec	F1
ResNet50	WCCE	0.90	0.92	0.87	0.895	0.82	0.82	0.84	0.830
	CCE	0.92	0.92	0.91	0.915	0.84	0.85	0.85	0.85
	FL	0.92	0.92	0.93	0.925	0.84	0.85	0.86	0.855
	DWFL	0.92	0.92	0.91	0.915	0.84	0.85	0.85	0.85
	VALF	**0.94**	**0.93**	**0.95**	**0.939**	**0.86**	**0.87**	**0.88**	**0.875**
EfficientNet	WCCE	0.79	0.78	0.77	0.775	0.88	0.88	0.89	0.885
	CCE	0.79	0.85	0.75	0.796	0.84	0.86	0.84	0.849
	FL	0.75	0.81	0.72	0.762	0.90	0.91	0.90	0.905
	DWFL	0.71	0.71	0.68	0.694	0.72	0.78	0.73	0.754
	VALF	**0.83**	**0.83**	**0.82**	**0.825**	**0.94**	**0.95**	**0.94**	**0.945**
DenseNet121	WCCE	**0.83**	**0.87**	0.80	**0.833**	0.90	0.90	0.91	0.905
	CCE	0.79	0.78	0.78	0.78	0.84	0.87	0.83	0.849
	FL	**0.83**	0.83	**0.82**	0.825	0.92	0.93	0.92	0.925
	DWFL	0.79	0.80	0.79	0.795	0.78	0.84	0.77	0.803
	VALF	**0.83**	0.83	**0.82**	0.825	**0.94**	**0.95**	**0.93**	**0.940**
InceptionV3	WCCE	0.83	0.83	0.82	0.824	0.78	0.80	0.80	0.80
	CCE	0.90	0.88	**0.90**	0.889	0.88	0.90	0.87	0.884
	FL	0.88	0.86	0.84	0.849	0.86	0.90	0.85	0.879
	DWFL	**0.92**	**0.91**	**0.90**	**0.905**	0.70	0.74	0.71	0.724
	VALF	0.90	0.89	0.89	0.89	**0.90**	**0.92**	**0.90**	**0.909**
ConvNeXt	WCCE	0.77	0.80	0.79	0.795	0.72	0.74	0.75	0.745
	CCE	**0.81**	0.81	**0.81**	0.81	0.88	0.90	0.88	0.889
	FL	0.69	0.69	0.69	0.69	0.84	0.86	0.85	0.855
	DWFL	0.77	0.84	0.76	0.798	0.74	0.86	0.77	0.812
	VALF	**0.81**	**0.89**	0.78	**0.831**	**0.88**	0.89	**0.89**	**0.89**

5.3 Precision-Recall Balance

Beyond accuracy, VALF frequently raises *both* precision and recall. Examples: ResNet50 at 20× improves from (0.92, 0.91)–(0.92, 0.93) under baselines to **(0.93, 0.95)**; at 40× it moves from (0.85, 0.86) (FL) to **(0.87, 0.88)**. EfficientNet at 40× lifts (0.91, 0.90) (FL) to **(0.95, 0.94)**; at 20× it trades a small precision drop (0.85 → **0.83**) for a sizable recall gain (0.75 → **0.82**), yielding a higher F1 (0.796 → **0.825**). DenseNet121 at 40× improves from (0.93, 0.92) (FL) to **(0.95, 0.93)**. For ConvNeXt at 20×, VALF substantially boosts precision (0.81 → **0.89**) with a slight recall decrease (0.81 → **0.78**), still raising F1 (0.81 → **0.831**).

Error patterns (Fig. 4). VALF collapses off-diagonal errors for *SCC* (recall ≈ 1.00), whereas WCCE/CCE/FL show leakage from SCC into ACA (e.g., 0.85 on the diagonal for CCE/FL). *NOR* remains essentially unchanged across losses (recall ≈ 0.92). *ACA* is not a minority class here but appears more confusable with SCC: its recall drops modestly under VALF (0.94 → 0.88) relative to WCCE/CCE, while still improving over FL (0.82). Overall, the matrices show VALF rebalancing decisions to better capture the harder SCC patterns without degrading NOR.

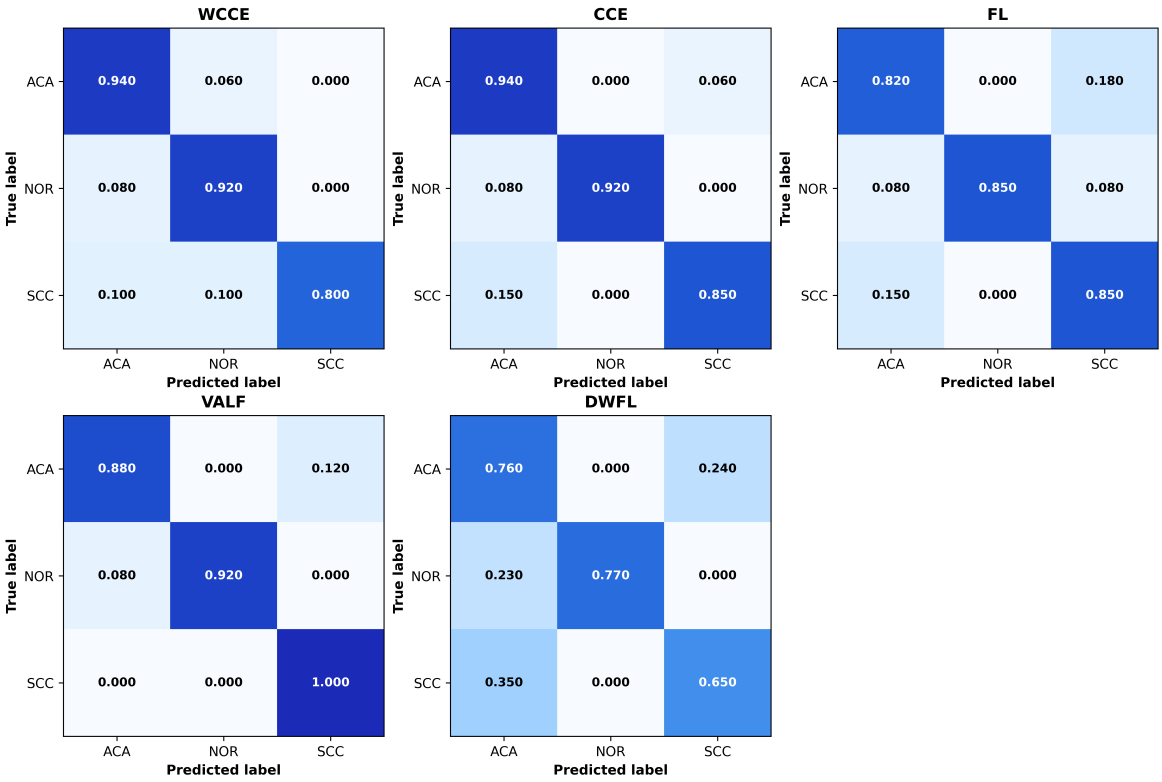

Figure 4: Normalized confusion matrices on the LungHist700 *test* split for WCCE, CCE, FL, and **VALF**. Rows are true labels (ACA, NOR, SCC); columns are predicted labels. Cell values denote per-class prediction rates (row-normalized); darker is better. VALF reduces off-diagonal errors-most notably between SCC and ACA-while preserving NOR performance.

6 Conclusion

This study presented a validation-adaptive focal loss, i.e., a plug and play modification of focal loss that initializes per-class weights uniformly or from any user-specified prior and then adapts them during training using validation feedback. By increasing the weight of under performing classes and softly decreasing the rest followed by normalization, VALF tracks evolving class difficulty without changing network architecture or training pipelines. On the LungHist700 histopathology dataset, VALF improved minority class recognition and raised overall performance across multiple backbones and magnifications. At 20×, we observed gains in AUC, accuracy, precision, recall, and F1 over focal loss baselines (e.g., up to 0.99 AUC and 0.94 accuracy). At 40×, VALF delivered consistent improvements (e.g., EfficientNet from 0.88 to 0.94 accuracy), indicating that adaptive, per-class weighting complements focal's instance-level focusing to better balance sensitivity and specificity. Because VALF operates solely at the loss level, it is easy to integrate and adds negligible overhead. Although VALF demonstrates promising improvements across multiple backbones, a key limitation remains: our evaluation is restricted to image level classification on a single moderate size dataset, and we did not assess segmentation or object detection or very large or multi center datasets, primarily because of limited computational resources. The future studies should test VALF beyond classification, particularly in segmentation and multi-instance learning where imbalance is more severe. Combining VALF with sampling or uncertainty-based methods may further improve minority

class performance. It is also important to validate the approach on datasets from different centers to check generalizability. Finally, analyzing how weights evolve during training could improve interpretability and help align the method with pathologists diagnostic needs.

8 Acknowledgment

During the preparation of this manuscript, we used OpenAI's ChatGPT-5 (accessed September 2025) to assist in summarizing our own findings, summarizing PDFs of related work, methods, and models, and for rewriting text to improve readability and grammatical clarity. All AI-assisted content was thoroughly reviewed, edited, and verified by the authors. Importantly, all scientific content including formulas and reported values, was carefully checked against original sources to ensure accuracy. Furthermore, the authors declare they had prior joint publications with one member of the journal editorial board.

References

[1] Cui Y., Jia M., Lin T.-Y., Song Y., and Belongie S., "Class-Balanced Loss Based on Effective Number of Samples," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, doi: 10.1109/CVPR.2019.00949.

[2] Kang B. *et al.*, "Decoupling Representation and Classifier for Long-Tailed Recognition," *arXiv*, Feb. 19, 2020. [Online]. Available: https://www.arxiv-vanity.com/papers/1910.09217/.

[3] Mao A., Mohri M., and Zhong Y., "Cross-Entropy Loss Functions: Theoretical Analysis and Applications," *arXiv*, Apr. 14, 2023. [Online]. Available: https://arxiv.org/abs/2304.07288.

[4] Lin T.-Y., Goyal P., Girshick R., He K., and Dollar P., "Focal Loss for Dense Object Detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, pp. 1–1, 2018, doi: 10.1109/TPAMI.2018.2858826.

[5] Ronneberger O., Fischer P., and Brox T., "U-Net: Convolutional Networks for Biomedical Image Segmentation," in *Lecture Notes in Computer Science*, vol. 9351, pp. 234–241, 2015, doi: 10.1007/978-3-319-24574-4_28.

[6] Chourasia P., Ali T. E., Ali S., and Patterson M., "DWFL: Enhancing Federated Learning through Dynamic Weighted Averaging," *arXiv*, Nov. 2024, doi: 10.48550/arXiv.2411.05173.

[7] Galdran A., Carneiro G., and GonzΓŸlez Ballester M. A., "Balanced-MixUp for Highly Imbalanced Medical Image Classification," in *Lecture Notes in Computer Science*, pp. 323–333, 2021, doi: 10.1007/978-3-030-87240-3_31.

[8] Menon A. K., Jayasumana S., Rawat A. S., Jain H., Veit A., and Kumar S., "Long-tail learning via logit adjustment," *arXiv*, Jul. 9, 2021. [Online]. Available: https://arxiv.org/abs/2007.07314

[9] Ren M., Zeng W., Yang B., and Urtasun R., "Learning to Reweight Examples for Robust Deep Learning," *arXiv*, Jan. 2018, doi: 10.48550/arXiv.1803.09050.

[10] Shu J. *et al.*, "Meta-Weight-Net: Learning an Explicit Mapping for Sample Weighting," *arXiv*, Feb. 2019, doi: 10.48550/arXiv.1902.07379.

[11] Yeung M., Sala E., Schonlieb C.-B., and Rundo L., "Unified Focal Loss: Generalising Dice and cross entropy-based losses to handle class imbalanced medical image segmentation," *Comput. Med. Imaging Graph.*, vol. 95, p. 102026, Jan. 2022, doi: 10.1016/j.compmedimag.2021.102026.

[12] Abraham N. and Khan N. M., "A Novel Focal Tversky Loss Function With Improved Attention U-Net for Lesion Segmentation," in *IEEE Int. Symp. Biomed. Imaging (ISBI) Workshops*, Apr. 2019. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/8759329.

[13] Ghosh A., Schaaf T., and Gormley M. R., "AdaFocal: Calibration-aware Adaptive Focal Loss," *arXiv*, Nov. 2022, doi: 10.48550/arXiv.2211.11838.

[14] Hossain M. S., Paplinski A. P., and Betts J. M., "Adaptive Class Weight based Dual Focal Loss for Improved Semantic Segmentation," *arXiv*, Sep. 2019, doi: 10.48550/arXiv.1909.11932.

[15] He K., Zhang X., Ren S., and Sun J., "Deep Residual Learning for Image Recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pp. 770–778, Jun. 2016, doi: 10.1109/CVPR.2016.90.

[16] Huang G., Liu Z., van der Maaten L., and Weinberger K. Q., "Densely Connected Convolutional Networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pp. 2261–2269, Jul. 2017, doi: 10.1109/CVPR.2017.243.

[17] Jalili A., Sajedi H., Tabrizchi H., and Mosavi A., "Skin Cancer Classification Using DenseNet," pp. 000333–000340, Sep. 2024, doi: https://doi.org/10.1109/sisy62279.2024.10737516.

[18] Szegedy C., Vanhoucke V., Ioffe S., Shlens J., and Wojna Z., "Rethinking the Inception Architecture for Computer Vision," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pp. 2818–2826, Jun. 2016, doi: 10.1109/CVPR.2016.308.

[19] Tan M. and Le Q. V., "EfficientNetV2: Smaller Models and Faster Training," *arXiv*, Apr. 2021, doi: 10.48550/arXiv.2104.00298.

[20] Liu Z., Mao H., Wu C.-Y., Feichtenhofer C., Darrell T., and Xie S., "A ConvNet for the 2020s," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, doi: 10.1109/CVPR52688.2022.01167.

[21] Diosdado J., Gilabert P., SeguГ S., and Borrego H., "LungHist700: A dataset of histological images for deep learning in pulmonary pathology," *Scientific Data*, vol. 11, no. 1, Oct. 2024, doi: 10.1038/s41597-024-03944-3.

Kaveh Vajedsamie,
Department of Computer Engineering,
University of Tehran, Tehran, Iran

Amirreza Jalili,
Department of Electrical and Computer Engineering, Sharif University of Technology, Tehran, Iran.

Hedieh Sajedi,
Department of Computer Science,
School of Mathematics, Statistics and Computer Science, University of Tehran, Tehran, Iran.
Email: hhsajedi@ut.ac.ir

Ardavan Delavar,
John von Neumann Faculty of Informatics,
Obuda University, Budapest, Hungary;
Doctoral School of Applied Informatics and Applied Mathematics, Obuda University, Budapest, Hungary;
ABB, Bonn, Germany;

Azamat Amirov,
Abylkas Saginov Karaganda Technical University, Karaganda, Kazakhstan.

Received 09.09.2025 , Accepted 12.12.2025, Available online 31.12.2025.