Available online at http://scik.org

Commun. Math. Biol. Neurosci. 2025, 2025:93

https://doi.org/10.28919/cmbn/9417

ISSN: 2052-2541

CRCDKD: A NOVEL ARCHITECTURE FOR MEDICAL SKIN CANCER CLASSIFICATION ON THE IMBALANCED HAM10000 DATASET

FRANKY SETIAWAN*, BENFANO SOEWITO

Master of Computer Science BINUS Graduate Program (BGP), BINUS UNIVERSITY, Jakarta 11530, Indonesia

Abstract: Addressing the critical challenge of imbalanced data in medical skin cancer classification, this paper proposes a novel Categorical Relation-Preserving Contrastive Decoupled Knowledge Distillation (CRCDKD) framework tailored for the HAM10000 dataset, a widely recognized benchmark for skin disease image analysis. To mitigate biases toward majority classes and enhance diagnostic reliability across all categories, the architecture integrates a mean-teacher paradigm with categorical relation-preserving contrastive learning, augmented by a newly proposed Decoupled Mean Teacher Knowledge Distillation (DMTKD) module. This synergistic approach decouples feature learning and knowledge distillation, enabling dynamic optimization of performance trade-offs between underrepresented and dominant categories while accelerating model convergence. The results demonstrated that the proposed framework achieved a balanced multiclass accuracy (BMA) of 84.45%, alongside an overall accuracy of 89.41%, precision of 83.27%, recall of 84.85%, specificity of 97.50%, F1-score of 83.39%, and an area under the curve (AUC) of 98.41%, surpassing state-of-the-art techniques. These metrics highlight significant improvements over existing methods, particularly in minority-class accuracy and balanced performance (BMA), with the DMTKD module offering unprecedented flexibility to adapt decision boundaries. The proposed framework not only advances skin cancer detection for imbalanced medical datasets but also introduces a generalizable paradigm for fairness-aware deep learning in healthcare applications, ensuring robustness across diverse clinical scenarios.

*Corresponding author

E-mail address: franky.setiawan@binus.ac.id

Received June 12, 2025

Keywords: medical image analysis; deep learning; imbalanced classification; knowledge distillation; HAM10000; melanoma; skin lesions; skin cancers; image classification; contrastive learning.

2020 AMS Subject Classification: 68T01, 68T10.

1. INTRODUCTION

Skin diseases represent a significant global health concern, with warts being the most prevalent type, affecting 41.3% of individuals, followed by acne (19.2%) and dermatitis (15%) [1]. Research indicates that approximately 185,103,774 people worldwide are affected by at least one dermatological condition, and skin diseases are projected to rank fourth among all health conditions in terms of global disease burden [2]. These conditions can profoundly diminish quality of life [3]. Given the anticipated future strain on healthcare systems, there is an urgent need for artificial intelligence (AI) models to support the diagnosis of skin diseases. Such models could assist clinicians by improving diagnostic accuracy, minimizing misdiagnosis, and facilitating early detection in underserved regions. Encouragingly, growing interest in AI-driven dermatology has led to the availability of numerous publicly accessible datasets—such as HAM10000 and BCN20000—that researchers can utilize to develop and validate these tools [4].

Previous studies have explored the potential of using artificial intelligence for diagnosing skin diseases. Mamun et al. [5] employed the Inception-V3 model to classify five common skin diseases, including Vascular Lesion, Solar Lentigo, Actinic Keratosis, Squamous Cell Carcinoma, and Basal Cell Carcinoma. Their research achieved a high accuracy of 98.16%, as measured by metrics such as accuracy, F1 score, and the ROC curve. Bozkurt [6] focused on skin lesion classification using Inception-ResNet-v2 and data augmentation techniques on dermatoscopic images, increasing accuracy from 83.59% to 95.09%. Other studies have incorporated Convolutional Neural Networks (CNNs) such as MobileNet v1 and Inception-V3. Purnama et al. [7] implemented these models in a web-based classifier. Their research found that the Inception-V3 model achieved an accuracy of 72%, while MobileNet v1 attained 58%. Another approach is the hybrid method proposed by Cengil et al. [8], which combines CNN-based feature extraction with K-Nearest Neighbors, Support Vector Machine (SVM), and Decision Tree methods using the HAM10000 dataset. The study found that the SVM-based architecture achieved the highest accuracy, with results as follows: AlexNet (77.16%), AlexNet+Tree (H1, 63.02%), AlexNet+KNN (H2, 75.63%), AlexNet+SVM (H3, 77.80%), ResNet18 (H4, 74.44%), ResNet18+Tree (H5, 60.65%), ResNet18+KNN (H6, 74.17%), and ResNet18+SVM (H7, 76.23%). Khan *et al.* [9] also

experimented with ResNet-50 and ResNet-101, achieving accuracies of 89.8%, 95.60%, and 90.20% on the HAM10000, ISBI2017, and ISBI2016 datasets, respectively .

Datasets like HAM10000 [10] face significant challenges due to severe class imbalances, where certain classes are underrepresented. This issue poses risks in deploying models in real-world scenarios, particularly in the medical domain, where minority classes often carry critical diagnostic importance [11]. Addressing these imbalances is essential for comprehensive model development, training, and evaluation, especially in medical contexts where gathering real-world data—particularly for rare diseases—is challenging [12].

Balancing performance between minority and majority classes in classification tasks remains a key challenge. Improving the minority class's performance can sometimes reduce accuracy for majority classes, which may represent common diseases requiring high precision. While not all applications necessitate prioritizing minority classes, removing them in specific scenarios can enhance majority-class performance. For instance, in disease screening, Tahir et al. [13] focused on only four classes, while Almaraz-Damian et al. [14] restricted their analysis to two majority classes (melanoma and nevus), achieving notable performance gains. This highlights the need for architectures that excel in handling underrepresented classes while allowing flexibility to adjust performance trade-offs between minority and majority classes based on application-specific requirements.

To address persistent challenges in medical AI research—particularly in skin cancer classification—this study proposes a skin disease classification model leveraging a novel knowledge distillation-based architecture. Building on prior work, our approach aims to mitigate class imbalance during training while delivering robust overall performance. This research contributes not only to advancing skin disease classification models but also to developing adaptable architectures capable of maintaining accuracy in imbalanced medical datasets.

2. RELATED WORKS

Over the past decade, significant advancements in artificial intelligence (AI), particularly in deep learning and Convolutional Neural Networks (CNNs), have enabled the development of reliable medical systems for image-based screening and diagnosis [15]. These techniques allow for the identification of skin conditions using standard images, leveraging image processing methods such as transformation, equalization, enhancement, edge detection, and segmentation. Skin images captured for disease identification and classification are processed and fed into advanced AI

methods, including Machine Learning, Deep Learning, Artificial Neural Networks (ANNs), CNNs, Backpropagation Neural Networks, and classifiers like Support Vector Machines (SVMs) and Bayesian classifiers, to predict skin disease types [16].

Andronescu et al. [17] developed multiple CNNs using Python, the Keras API, and the TensorFlow framework, with an emphasis on dataset curation. Their research produced a classifier achieving high accuracy on the training dataset but moderate performance on new data. Using the HAM10000 dataset, the model classified seven skin disease classes with an accuracy of 72.1%.

A. [18] employed a transfer learning approach, utilizing AlexNet as the pretrained model. The final three layers were replaced with layers optimized for binary classification. The HAM10000 dataset was split into a 90% training set and 10% test set, with images resized to 227×227 pixels to match AlexNet's input requirements. Trained using stochastic gradient descent with momentum, the model achieved 84% accuracy, 81% sensitivity, and 88% specificity on the test set at a confidence threshold of 0.5.

Almaraz-Damian et al. [14] proposed a novel Computer-Aided Detection (CAD) system for melanoma classification, integrating handcrafted features based on the Asymmetry, Borders, Colors, and Dermatoscopic Structures (ABCD) rule. The HAM10000 dataset was partitioned into 75% training and 25% testing subsets. After preprocessing, ABCD-based features were fused with deep learning models, including VGG16, VGG19, MobileNet V1/V2, ResNet-50, DenseNet-201, Inception V3, and Xception. MobileNet V2 yielded the highest performance: 92.4% accuracy, 86.41% specificity, 92.08% precision, 89.16% F1-score, 0.90 G-Mean, 0.80 IBA, and 0.7953 Matthews Correlation Coefficient (MCC).

Liu et al. [19] introduced a semi-supervised framework for medical image classification, the Sample Relation Consistency (SRC) method, which leverages unlabeled data by encoding inter-sample relationships. Applied to the HAM10000 and ChestX-ray14 datasets (split into 70% train, 10% validation, 20% test sets), the SRC framework incorporated teacher-student knowledge distillation, noise addition, and consistency regularization. On HAM10000, it achieved 84.73% accuracy, 73.88% average precision, 76.55% balanced multi-class accuracy, and 74.63% F1-score across seven classes.

Thurnhofer-Hemsi & Domínguez [20] explored skin cancer detection via transfer learning on five CNNs—DenseNet201, GoogLeNet, Inception-ResNetV2, InceptionV3, and MobileNetV2—to design a hierarchical two-stage classifier addressing class imbalance. The HAM10000 dataset was split into 70% train, 20% validation, and 10% test sets. The DenseNet201-based model achieved

87.7% accuracy and 83% F1-score.

Miglani & Bhatia [21] investigated skin lesion classification using transfer learning with fine-tuned EfficientNet-B0 and ResNet-50 models on HAM10000. Both models classified seven classes, with EfficientNet-B0 achieving 93% micro-averaged AUC and 97% macro-averaged AUC, compared to ResNet-50's 91% micro-averaged AUC and 96% macro-averaged AUC.

Srinivasu et al. [22] proposed a hybrid approach combining MobileNet V2 and Long Short-Term Memory (LSTM) networks for skin disease classification. Tested on HAM10000 (85% train, 5% validation, 10% test split), the model achieved 85.34% accuracy, 88.24% sensitivity, 92% specificity, 91.07% Jaccard Similarity Index (JSI), and 86.00% MCC for seven classes.

Xing et al. [23] presented the Categorical Relation-Preserving Contrastive Knowledge Distillation (CRCKD) framework for medical image classification. This approach included a Contrastive Class Distillation (CCD) module to separate positive/negative image pairs and a Categorical Relation Preserving (CRP) loss to distill relational knowledge in a class-balanced manner. Evaluated on HAM10000 and APTOS datasets (80% train, 20% test split with 5-fold cross-validation), CRCKD achieved 85.66% accuracy, 76.35% precision, 78.07% balanced multi-class accuracy, and 76.45% F1-score on HAM10000, and 84.87% accuracy, 73.18% precision, 71.90% balanced accuracy, and 72.22% F1-score on APTOS.

Jain et al. [24] investigated skin cancer classification using transfer learning and image replication to address dataset imbalance. They conducted a comparative analysis of six transfer learning networks—VGG19, InceptionV3, Inception-ResNetV2, ResNet50, Xception, and MobileNet—on the HAM10000 dataset, achieving the best results with Xception: 89.66% accuracy, 89.57% average recall, 88.76% average precision, and 89.02% average F-measure. The dataset was split into 72% train, 8% validation, and 20% test sets.

Chaturvedi et al. [25] developed a MobileNet model for efficient seven-class skin cancer classification, achieving performance comparable to or exceeding that of expert dermatologists. Using the HAM10000 dataset (split into 90.63% train and 9.37% test sets), the model achieved 83.1% accuracy, 91.36% top-2 accuracy, 95.34% top-3 accuracy, 89% weighted average precision, 83% average recall, and 83% average F1-score.

H.-W. Huang et al. [26] proposed a lightweight deep learning model for cloud-based and remote skin cancer diagnosis, achieving 85.8% accuracy on HAM10000 and 72.1% accuracy on the KCGMH dataset. Their multi-class classification model handled seven classes for HAM10000 and five classes for KCGMH.

Popescu et al. [27] designed a Collective Intelligence-based System (CIS) for skin lesion classification, combining nine neural networks (AlexNet, GoogLeNet, GoogLeNet-Places365, MobileNet-V2, Xception, ResNet-50, ResNet-101, Inception-ResNet-V2, DenseNet201) on HAM10000. The CIS achieved 86.71% accuracy, surpassing individual model performance.

Alam et al. [28] explored a deep learning-based skin cancer classifier for imbalanced datasets, fine-tuning AlexNet, InceptionV3, and RegNetY-320 with hyperparameter optimization. Their contributions included normalization, image resizing, data augmentation, and validation using state-of-the-art detectors. The HAM10000 dataset was split into 70% train and 30% test sets, achieving 91% accuracy, 88.1% F1-score, and an ROC AUC of 0.95.

Hoang et al. [29] developed a multiclass skin lesion classification framework using a novel entropy-based weighting and first-order cumulative moment (EW-FCM) segmentation method to isolate lesions from backgrounds, followed by wide-ShuffleNet classification. Tested on HAM10000 and ISIC2019 datasets across three experiments with varying train-test splits, the model achieved accuracies of 84.80%, 86.33%, and 82.56%, with corresponding sensitivities (84.80%, 86.33%, 82.56%) and specificities (97.48%, 97.72%, 97.51%).

Fraiwan & Faouri [30] evaluated raw deep transfer learning models for seven-class skin lesion classification using 13 architectures without explicit feature extraction, preprocessing, or segmentation. On HAM10000, DenseNet201 outperformed other models, achieving F1 scores of 64.8%, 66.1%, and 74.4% across three train-test splits.

Alwakid et al. [31] proposed a melanoma detection method combining Enhanced Super-Resolution Generative Adversarial Networks (ESRGAN) for image enhancement, segmentation, and data augmentation to address class imbalance. Using CNN and modified ResNet-50 on HAM10000, they achieved high accuracy and other performance metrics.

Nguyen et al. [32] applied deep learning models (DenseNet, InceptionNet, ResNet) with Soft-Attention for skin lesion classification on imbalanced data. InceptionResNetV2 yielded the best results: 90% accuracy, 86% precision, 86% F1-score, 81% recall, and 0.99 AUC on HAM10000.

Gajera et al. [33] analyzed dermoscopy images for melanoma detection using deep CNN features. They extracted features from eight CNN models and validated combinations of CNNs and classifiers on benchmark datasets (PH, ISIC 2016, ISIC 2017, HAM10000). DenseNet-121 with a multi-layer perceptron achieved the highest accuracy: 98.33% (PH), 80.47% (ISIC 2016), 81.16% (ISIC 2017), and 81% (HAM10000).

Tahir et al. [13] introduced DSCC_Net for classifying four skin cancers (MEL, BCC, MN, SCC)

using combined ISIC-2020 and HAM10000 datasets augmented with SMOTE Tomek. The model achieved 94.17% accuracy, 93.76% recall, 93.93% F1-score, 94.28% precision, and 99.42% AUC. Hossain et al. [34] employed a max voting ensemble of 10 pretrained models for skin cancer detection, achieving 93.19% precision, 93.18% recall, and 93.18% F1-score on ISIC2018 Task 1–2, and 95% accuracy, precision, recall, and F1-score on HAM10000.

Among five studies [14], [20], [30], [32], [33], DenseNet outperformed other base models in three, consistently outperforming ResNet101, Xception, ShuffleNet, DarkNet-53, and EfficientNetB0. In the remaining two studies [14], [32], DenseNet's suboptimal performance was attributed to integration with other methods, suggesting external factors influenced results. DenseNet also demonstrated superior performance among 13 models tested [30]. Regarding the sophistication of skin disease research involving DenseNet, Hossain et al. [34] and Xing et al. [23] stood out. While Hossain's max voting ensemble achieved higher performance, Xing et al. [23] introduced the more original Categorical Relation-Preserving Contrastive Knowledge Distillation (CRCKD) framework, emphasizing innovation over raw accuracy.

3. MATERIALS AND METHODS

3.1. CONCEPTUAL FRAMEWORK

The conceptual framework outlines the structure of this research. It is divided into three main phases: research preparation, experimentation, and research reporting.

For the **research preparation phase**, the focus is on identifying problems, conducting preliminary literature reviews, exploring the background of the problem, defining research problem statements, examining research objectives and benefits, justifying the research fundamentals (including its problems, objectives, benefits, and scope), conducting a thorough literature review, identifying the research methodology, and formulating the methodology in detail.

For the **experimentation phase** involves preparing the experimental platform, software, datasets, and conducting the experiments.

For the **research reporting phase** includes presenting and discussing the research results, drawing conclusions, and outlining future work. The entire conceptual framework is illustrated in Figure 1.

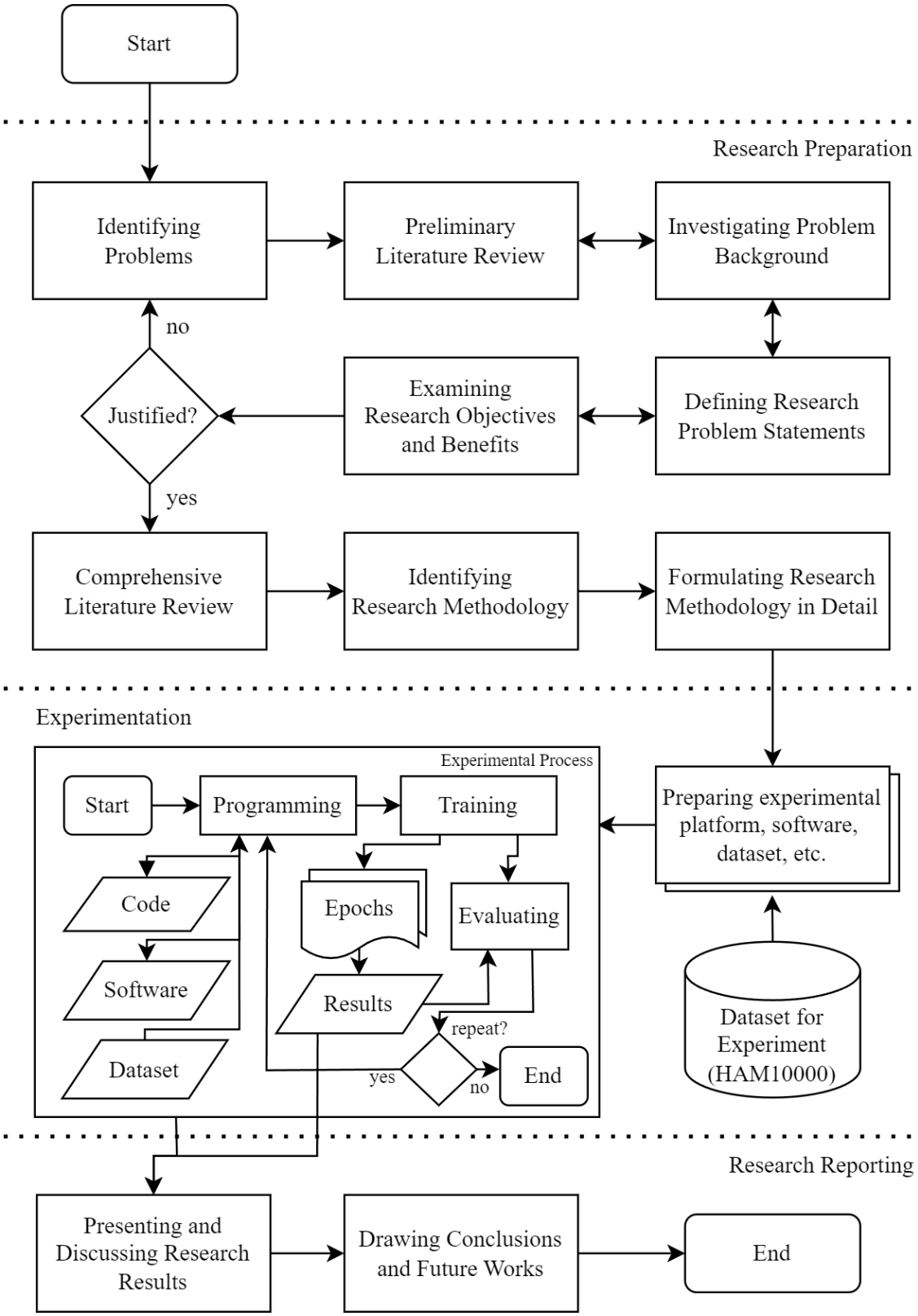


Figure 1. The whole of conceptual framework showing three main phases of this research: research preparation, experimentation, and research reporting.

3.2. EXPERIMENT PROCEDURE

The experiment procedure is divided into three main phases: data preprocessing, training and testing, and evaluation. During the data preprocessing phase, the dataset undergoes profiling, is split into 5-fold cross-validation sets, loaded into dataloaders, transformed, and normalized. In the training and testing phase, both baseline and proposed models undergo repetitive training and testing, data collection of results, and finally, the proposed models are saved. In the evaluation phase, data collected from the previous phases (both baseline and proposed models) are compared. Further details will be discussed in the following sections. The entire experiment procedure is illustrated in Figure 2.

3.3. EXPERIMENTAL PLATFORM

The experimental platform used for the experiments is specified in Table 1. To reproduce this experiment, it doesn't need to be identical. However, at least 22 GB of VRAM is required due to model size.

Table 1. Hardware specifications used for the experiments.

Part	Specification
CPU	Intel Xeon® E5-2696 v3 18C36T @ 2.3GHz
GPU	NVIDIA® RTX 4090 24.0GB VRAM
RAM	DDR3 64.0GB RAM
SSD	Netac NVMe SSD 4TB

3.4. DATASET PROFILE

The dataset used in this experiment is HAM10000 (Human Against Machine with 10,000 Training Images). It comprises a collection of human skin images used as training data for artificial intelligence models to recognize various skin lesion types (Tschandl et al., 2018 [10]).

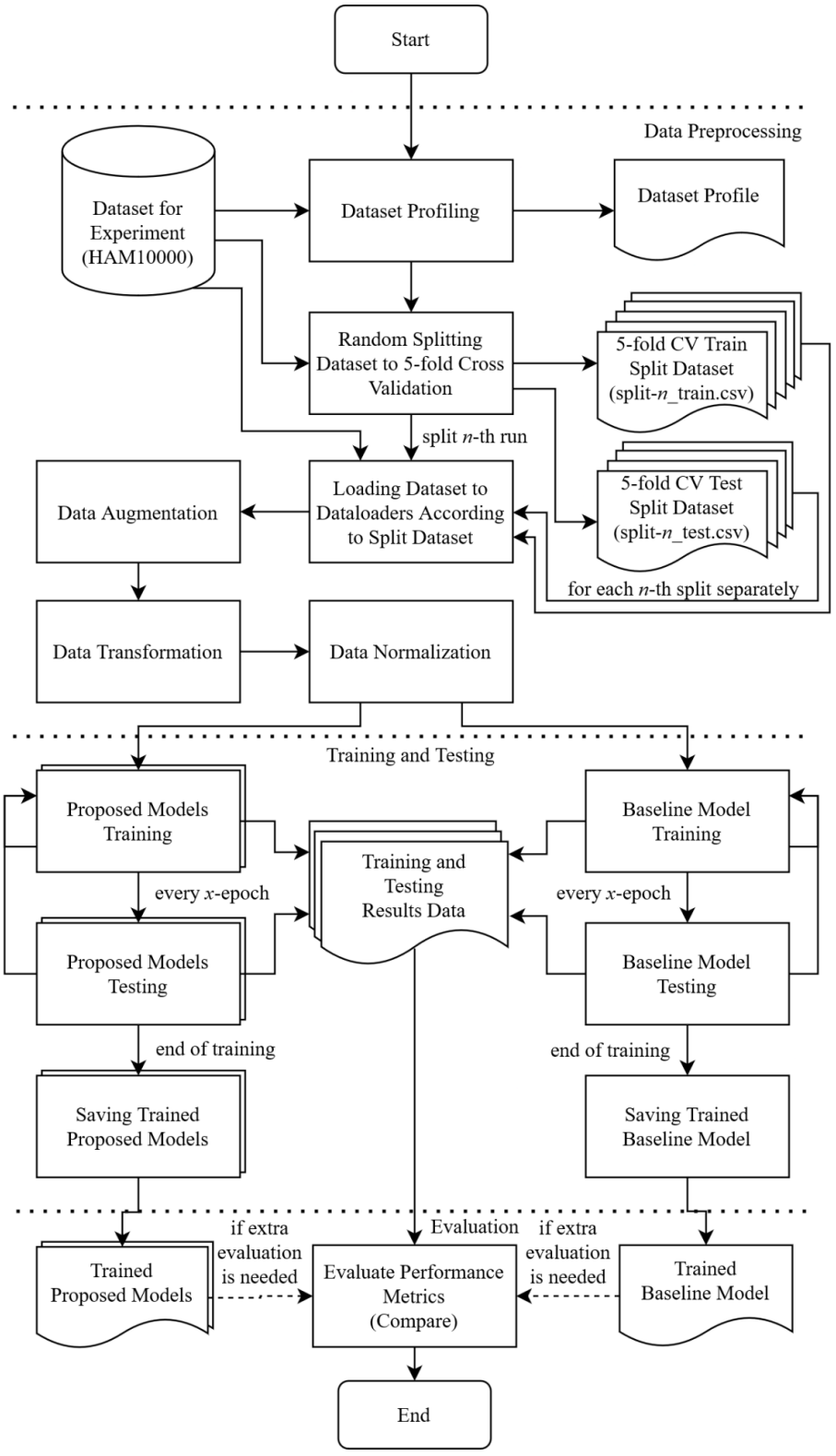


Figure 2. The whole of experimental procedure indicating three main phases of experiment procedures: data preprocessing, training and testing, and evaluation.

The HAM10000 database includes dermatoscopic images collected from individuals worldwide, totaling 10,015 images. It provides metadata in CSV format, containing details such as gender, age, and lesion class. This dataset is part of the ISIC 2018 challenge, which comprises three tasks: lesion boundary segmentation (Task 1), lesion attribute detection (Task 2), and disease classification (Task 3).

The dataset covers seven skin disease categories: actinic keratoses and intraepithelial carcinoma (AKIEC), basal cell carcinoma (BCC), benign keratosis-like lesions (BKL), dermatofibroma (DF), melanoma (MEL), *melanocytic nevi* (NV), and vascular lesions (VASC). The distribution of disease classes in the HAM10000 dataset is presented in Table 2. and Figure 3.

Table 2. The class distribution of HAM10000 dataset.

#	Class	Amount of Images	Representing
1	*Melanocytic nevi* (NV)	6705	66.94%
2	Melanoma (MEL)	1113	11.11%
3	Benign keratosis-like lesions (solar lentigines or seborrheic keratoses and lichen-planus like keratoses) (BKL)	1099	10.97%
4	Basal cell carcinoma (BCC)	514	5.132%
5	*Actinic keratoses* and intraepithelial carcinoma or Bowen's disease (AKIEC)	327	3.265%
6	Vascular lesions (*angiomas*, *angiokeratomas*, pyogenic granulomas and hemorrhage) (VASC)	142	1.418%
7	Dermatofibroma (DF)	115	1.148%

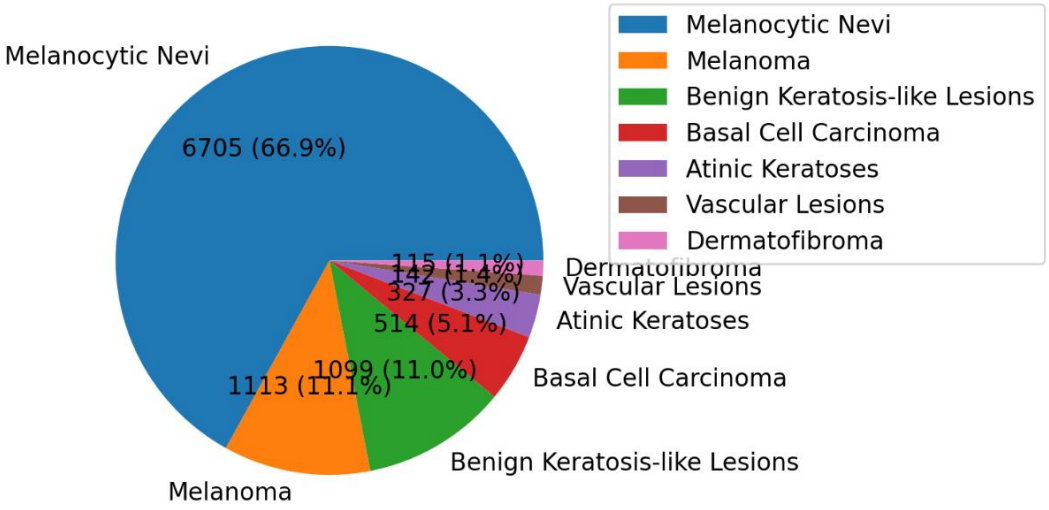


Figure 3. Visualization of HAM10000 Dataset Classes Distribution

One notable challenge with this dataset is the imbalance among classes, particularly with NV comprising the majority at 66.9% of the total images. This imbalance leads to significant training challenges due to the imbalanced distribution of skin disease instances. The second largest class is BKL, constituting roughly 11.1% of the images, while the remaining classes have significantly fewer representations. For instance, the DF class accounts for less than 2% of the total images and is notably challenging for prediction. Figure 4 provides a visual representation of random sample images from the HAM10000 dataset.

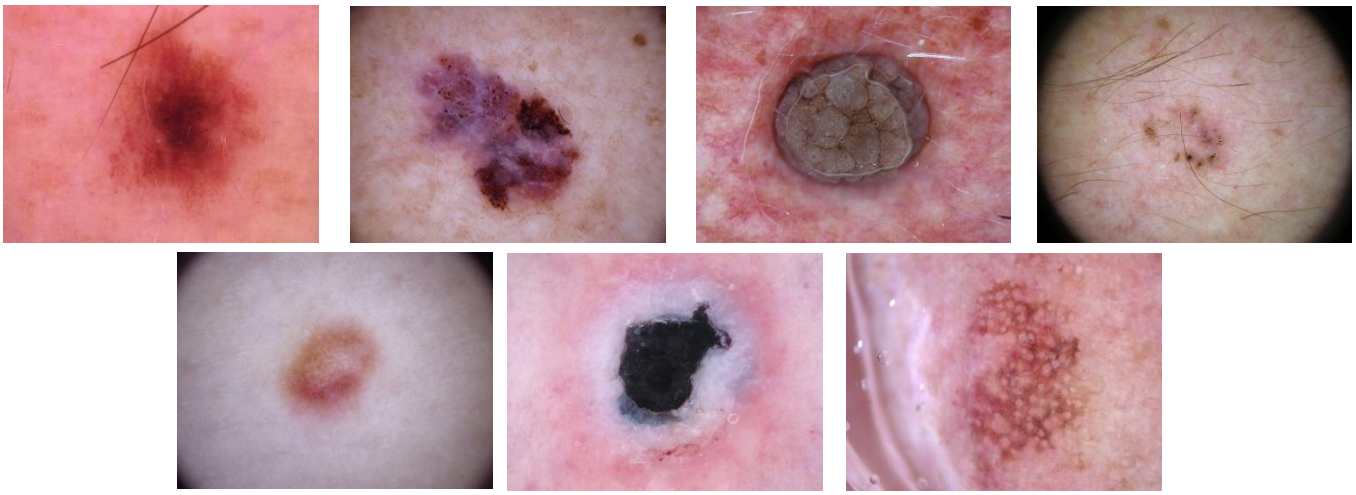


Figure 4. An example image from each class of the HAM10000 dataset, arranged clockwise from top-left to bottom-left: NV, MEL, BKL, BCC, AKIEC, VASC, DF.

3.5. PROPOSED ARCHITECTURE

A novel CRCDKD architecture is proposed. The approach builds up on the CRCKD [23] and the mean teacher method [35], incorporating empirically-validated architectural decisions and a novel DMTKD module. The aim is to improve both the performance and adjustability of the model. CRCDKD is illustrated in Figure 5.

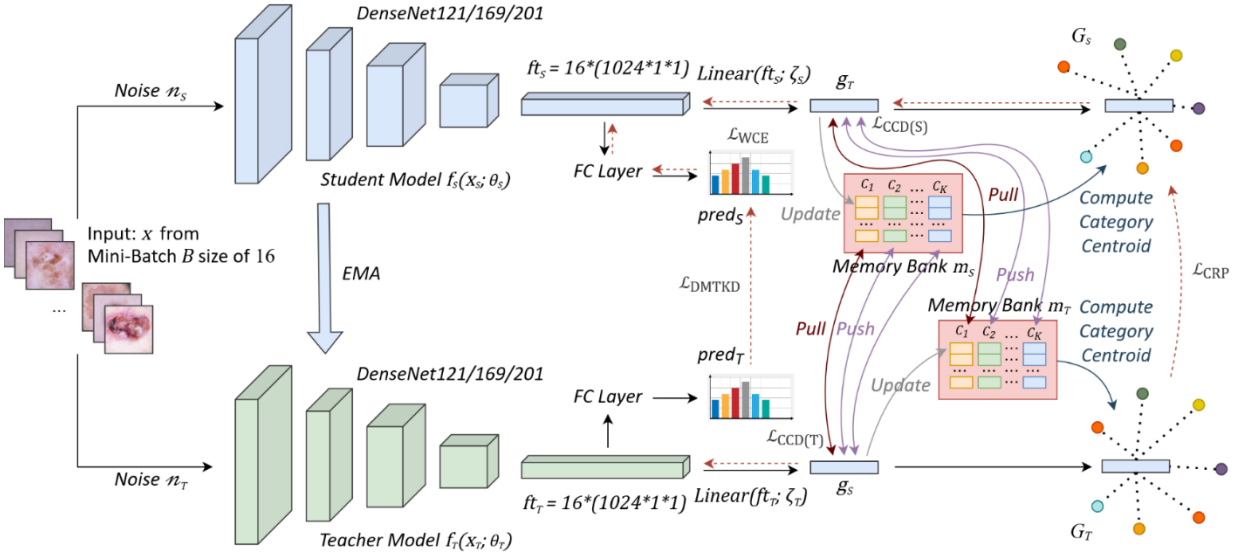


Figure 5. Overview of CRCDKD framework for semi-supervised medical skin disease image classification. CRCDKD utilizes a mini-batch B size of 16 and employs DenseNet-121/169/201 as the base architecture. Teacher weight is updated by Exponential Moving Average (EMA) of student weight. The student model is optimized using a combination weighted cross-entropy loss ($\mathcal{L}_{WCE}$), DMTKD loss ($\mathcal{L}_{DMTKD}$), and unsupervised consistency losses ($\mathcal{L}_{CCD}$ and $\mathcal{L}_{CRP}$). The dashed red lines represent the flow of gradients.

3.6. ALGORITHMS

As depicted in Figure 5, the CRCDKD architecture comprises a student and a mean-teacher model. The input x is a mini-batch B with a size of 16 images. Upon receiving x as input, the images undergo augmentation twice, and perturbations are applied, resulting in x_S and x_T. The noises η_S and η_T represent the application of input perturbations, including random affine transformations and random horizontal flips, which are only applied to training images. Here, $f(\cdot)$ denotes the classification networks, which are divided into $f_S(\cdot)$ and $f_T(\cdot)$ representing the student and teacher networks, respectively. Both networks utilize a pretrained DenseNet-169 as the backbone. θ_S and θ_T represent the parameter weights of the student model and teacher model, respectively. θ_T is updated through the EMA of θ_S, while θ_S is updated through stochastic gradient descent. $f_S(\cdot)$ takes x_S and produces feature representation ft_S, while $f_T(\cdot)$ takes x_T and produces feature representation ft_T. Respective fully connected (FC) layers then

output the student prediction soft labels $pred_S$ and the teacher prediction soft labels $pred_T$. Soft labels $pred_S$ are penalized with a weighted cross-entropy loss $\mathcal{L}_{WCE}$ and the proposed DMTKD loss $\mathcal{L}_{DMTKD}$ by comparing them to $pred_T$. ζ_S and ζ_T represent trainable parameters in linear transformation layers $Linear(ft_S;\zeta_S)$ and $Linear(ft_T;\zeta_T)$ respectively. These linear layers project g_S and g_T embeddings respectively. m_S and m_T are memory banks of the student and teacher respectively. g_S and g_T are saved to memory banks m_T and m_S respectively. $\mathcal{L}_{CCD(S)}(\cdot)$ and $\mathcal{L}_{CCD(T)}(\cdot)$ denote the CCD loss of the student and teacher respectively. $\mathcal{L}_{CCD(S)}(\cdot)$ takes g_S and m_T, producing gradients that flow to both θ_S and ζ_S, while $\mathcal{L}_{CCD(T)}(\cdot)$ takes g_T and m_S, producing gradients that only flow to ζ_T. The CRP loss $\mathcal{L}_{CRP}$ takes both m_S and m_T to build G_S and G_T respectively, which are then compared to produce gradients that flow to both θ_S and ζ_S. The CRCDKD algorithm is detailed in Table 3.

Table 3. The detailed step-by-step process of the CRCKD algorithm.

Algorithm 1: **CRCDKD Algorithm**

Input:
$x \leftarrow$ mini-batch B with size of 16 images
loader ← dataset containing images with test or train split type

1. Initialize $f_S(\theta_S)$ and $f_T(\theta_T)$ with pretrained DenseNet-121/169/201, $ra(\cdot)$ as random affine function, $rhf(\cdot)$ as random horizontal flipping function, $norm(\cdot)$ as normalization function, $Linear(\zeta_S)$ and $Linear(\zeta_T)$ as linear transformations layers, m_S and m_T as memory banks, $centroid(\cdot)$ as centroid constructing function, $grad(\cdot)$ as gradient backward-descent function.
2. **for** x in *loader* **do**
3. Define $perturb(\cdot) \leftarrow norm\left(rhf\left(ra(\cdot)\right)\right)$
4. **if** type(*loader*) = train **do**
5. $x_S \leftarrow perturb(x)$
6. $x_T \leftarrow perturb(x)$
7. $ft_S, pred_S \leftarrow f_S(x_S;\theta_S)$
8. $ft_T, pred_T \leftarrow f_T(x_T;\theta_T)$
9. $loss \leftarrow \mathcal{L}_{WCE}(pred_S, loader)$
10. $loss \leftarrow loss + \mathcal{L}_{DMTKD}(pred_S, pred_T)$
11. $g_S \leftarrow Linear(ft_S;\zeta_S)$
12. $g_T \leftarrow Linear(ft_T;\zeta_T)$
13. $m_S \leftarrow$ Push$(m_S,\ g_T)$
14. $m_T \leftarrow$ Push$(m_T,\ g_S)$
15. $loss \leftarrow loss + \mathcal{L}_{CCD(S)}(g_S, m_T)$
16. $loss \leftarrow loss + \mathcal{L}_{CCD(T)}(g_T, m_S)$
17. $G_S \leftarrow$ centroid(m_S)
18. $G_T \leftarrow$ centroid(m_T)
19. $loss \leftarrow loss + \mathcal{L}_{CRP}(G_S, G_T)$
20. **if** type(*loader*) = train **do**
21. $\theta_S, \zeta_S, \zeta_T \leftarrow grad(\theta_S, \zeta_S, \zeta_T; loss)$

3.7. COMPONENTS

3.7.1. EMPIRICALLY-VALIDATED ARCHITECTURAL DECISIONS

In previous work, (Xing et al., 2021[23]) used a mini-batch $B = 64$, which is suspected to be excessive. Increasing B can lead to oversmoothing of the gradient ∇L, resulting in more averaged gradient estimates during training and potentially causing the model to converge to a flatter minimum. This compromise in convergence behavior may negatively impact the model's generalization performance—particularly concerning the HAM10000 dataset, which is known for its high level of noise and severe class imbalance. To address this issue, the proposed CRCDKD approach employs a smaller mini-batch size of $B = 16$, which will be experimentally evaluated to determine its effect on improving model performance.

Additionally, DenseNet-121 might be insufficient in capturing all the features within the images from the challenging HAM10000 dataset, which could hinder the model's ability to learn intricate patterns effectively. Upgrading the model size to a larger variant like DenseNet-169 or DenseNet-201 may potentially enhance perfor-mance. The utilization of larger models, specifically DenseNet-169 up to DenseNet-201, will be experimentally validated to assess their impact on improving the model's performance.

3.7.2. MEAN TEACHER METHOD

Updating the weights of the teacher model with the EMA student weights at different training iterations enhances the reliability of the teacher model, aiding in generating consistent targets [35]. The teacher weights $\theta_{T,t}$ at training iteration t are updated as follows:

(1) $$\theta_{T,t} = \theta_{T,t-1} + (1-\alpha)\theta_{S,t}$$

where $\theta_{T,t-1}$ represents the teacher weights at training iteration $t-1$, α is a smoothing coefficient hyperparameter, and $\theta_{S,t}$ denotes student weights at training iteration t. The teacher weights $\theta_{T,t}$ are updated only during training phase.

3.7.3. DECOUPLED MEAN TEACHER KNOWLEDGE DISTILLATION (DTMKD)

Inspired by DKD proposed by Zhao et al. [36], which achieved equal or better performance by decoupling target class and non-target class knowledge distillation, we propose a novel decoupling module called DMTKD. DMTKD decouples the distillation of target class and non-target class using logits from the mean teacher, introducing a novel approach for knowledge distillation in the context of medical skin image classification with the mean teacher method. This approach potentially improves flexibility and enhances network performance for specific classification tasks. Sometimes, the original coupled knowledge distillation is still useful when fine-grained

adjustments are needed to improve the model. For example, introducing distillation of some coupled knowledge can be convenient to further improve the model. However, with fully decoupled knowledge distillation, it is impossible to adjust for the addition of vanilla coupled knowledge distillation, since it has already been strictly decoupled. As a compromise, the DMTKD loss is defined as follows:

$$\begin{aligned}\theta_{T,t} &= \theta_{T,t-1} + (1-\alpha)\theta_{S,t}\mathcal{L}_{DMTKD}(pred_S, pred_T) \\ &= \lambda_{orig} * \left(\mathcal{L}_{KL}(pred_S, pred_T)\right) + \lambda_{dc} \\ &* (\lambda_{tckd} * \mathcal{L}_{TCKD}(pred_S, pred_T, \tau_{DMT}) \\ &+ \lambda_{nckd} * \mathcal{L}_{NCKD}(pred_S, pred_T, \tau_{DMT}))\end{aligned} \tag{2}$$

where $\mathcal{L}_{orig}$ denotes the weight hyperparameter of KL (Kullback–Leibler) divergence loss $\mathcal{L}_{KL}$, λ_{dc} denotes the weight hyperparameter of the decoupled target class knowledge distillation loss, with λ_{tckd} denotes weight of $\mathcal{L}_{TCKD}$ and λ_{nckd} denotes the weight of non-target class knowledge distillation loss $\mathcal{L}_{NCKD}$, while τ denotes the distillation temperature.

KL divergence measures the difference or divergence between one probability distribution p^s and a second, expected probability distribution p^t. Formally, KL divergence can be expressed as:

$$KL(p^t||p^s) = \sum_y p_y^t \left(log(p_y^t) - log(p_y^s)\right) \tag{3}$$

KL divergence loss $\mathcal{L}_{KL}$, which measures the difference between the student's probability distribution $pred_S$, and the teacher's probability distribution $pred_T$, utilizes Equation 3. in slightly different form. Given *C* as total classes in $pred$, both $pred_S$ and $pred_T$ are transferred to the log probability domain by $softmax(\cdot)$, $\mathcal{L}_{KL}$ can be defined as:

$$\mathcal{L}_{KL} = \sum_i^C softmax(pred_{T,i}) * \left(log(softmax(pred_{T,i})) - log(softmax(pred_{S,i}))\right) \tag{4}$$

Meanwhile, the target class knowledge distillation loss $\mathcal{L}_{TCKD}$ measures the difference between the student's and teacher's binary probabilities of the target class. Let z_i represent the logit of i-th class in $pred$, and let *C* be the number of classes in $pred$. Therefore, the probabilities of the target class $p^{\mathcal{T}}$ and all the other non-target classes $p^{\backslash\mathcal{T}}$ can be calculated by:

$$p^{\mathcal{T}} = \frac{exp(z_t)}{\sum_{j=1}^C exp(z_j)}, p^{\backslash\mathcal{T}} = \frac{\sum_{k=1,k\neq\mathcal{T}}^C exp(z_k)}{\sum_{j=1}^C exp(z_j)} \tag{5}$$

On the other hand, given $\hat{\mathbf{p}} = [\hat{p}_1, \hat{p}_2, \dots, \hat{p}_{t-1}, \hat{p}_{t+1}, \dots, \hat{p}_C] \in \mathbb{R}^{1\times(C-1)}$, the probability of the i-

th class among non-target classes $\hat{p}_i$ inside $\hat{\mathbf{p}}$ can be calculated as the follows:

(6) $$\widehat{p_i} = \frac{exp(z_i)}{\sum_{j=1,j\neq\mathcal{T}}^{C} exp(z_j)}$$

Given $\mathbf{p} = [p_1, p_2, \dots, p_t, \dots, p_C] \in \mathbb{R}^{1\times C}$ is the classification probabilities and $\mathbf{b} = [p^{\mathcal{T}}, p^{\backslash\mathcal{T}}] \in \mathbb{R}^{1\times 2}$ is the binary probabilities of the target class $p^{\mathcal{T}}$ and all the other non-target classes $p^{\backslash\mathcal{T}}$, thus $\mathcal{L}_{TCKD}$ can be calculated as follows:

(7) $$\mathcal{L}_{TCKD} = KL(\boldsymbol{b_T}||\boldsymbol{b_S})$$

where $KL(\cdot)$ is KL divergence loss.

Meanwhile, $\mathcal{L}_{NCKD}$ measures the difference between the student's and teacher's probabilities of the target class. $\mathcal{L}_{NCKD}$ is expressed as:

(8) $$\mathcal{L}_{NCKD} = KL(\widehat{\boldsymbol{p}}_{\boldsymbol{T}}||\widehat{\boldsymbol{p}}_{\boldsymbol{S}})$$

3.7.4. Class-guided Contrastive Distillation (CCD)

The CCD method proposed by Xing et al. [23] guides the distillation process using class-label information. It considers samples from the same class as positive pairs, bringing their representations closer, and treats samples from different classes as negative pairs, pushing their representations apart. This approach aims to achieve a more refined alignment of features.

For the student model, the CCD loss is expressed as:

(9) $$\mathcal{L}_{CCD}^{(S)}(\theta_s, \zeta_s) = -\frac{1}{k_P}\sum_{i=1}^{k_P}\left(log\frac{e^{(g_s\cdot g_{t,i}/\tau_{CRC})}}{e^{(g_s\cdot g_{t,i}/\tau_{CRC})}+\frac{k_N}{M}} + \sum_{j=1}^{k_N} log\left(1-\frac{e^{(g_s\cdot g_{t,j}/\tau_{CRC})}}{e^{(g_s\cdot g_{t,j}/\tau_{CRC})}+\frac{k_N}{M}}\right)\right)$$

where τ is temperature which controls concentration level, k_P is the number of positive samples and k_N is the number of negative samples, and M is cardinality of the dataset.

For the teacher model, CCD loss is expressed as:

(10) $$\mathcal{L}_{CCD}^{(T)}(\zeta_t) = -\frac{1}{k_P}\sum_{i=1}^{k_P}\left(log\frac{e^{(g_s\cdot g_{s,i}/\tau_{CRC})}}{e^{(g_s\cdot g_{s,i}/\tau_{CRC})}+\frac{k_N}{M}} + \sum_{j=1}^{k_N} log\left(1-\frac{e^{(g_s\cdot g_{s,j}/\tau_{CRC})}}{e^{(g_s\cdot g_{s,j}/\tau_{CRC})}+\frac{k_N}{M}}\right)\right)\theta_{T,t}$$

$$= \theta_{T,t-1} + (1-\alpha)\theta_{S,t}$$

In this case, $\mathcal{L}_{CCD}^{(T)}(\zeta_t)$ only updates the projection head of teacher model.

3.7.5. Categorical Relation Preserving (CRP)

The CRP loss proposed by Xing et al. [23] utilizes category centroids to create a relation graph

that captures nuanced relational knowledge, leading to a deeper understanding of relationships between classes. The centroid of the i-th category is calculated by averaging the features of all samples in the i-th class from the memory bank, denoted as m_S for the student model and m_T for the teacher model:

$$C_i^S = \frac{1}{|C_i|} \sum_{m_S \in C(i)} m_S \, , C_i^T = \frac{1}{|C_i|} \sum_{m_T \in C(i)} m_T \tag{11}$$

where $|C_i|$ is the number of samples in the i-th class. For each query x_s for the student model and x_t for the teacher model in the mini-batch, their cosine similarity with all category centroids is computed in the student and teacher models, respectively. After applying softmax across all classes, the categorical relation between the sample x_s for the student model and x_t for the teacher model can be expressed as:

$$\left(x_s, C_i^S\right) = \frac{e^{g_s \cdot C_i^S}}{\sum_{i=1}^{K} e^{g_s \cdot C_i^S}}, R(x_t, C_i^T) = \frac{e^{g_t \cdot C_i^T}}{\sum_{i=1}^{K} e^{g_t \cdot C_i^T}} \tag{12}$$

where *K* represents the overall number of classes present in the dataset, while g_s and g_t refer to the representations of the sample extracted by the student and teacher, respectively. Then, the CRP loss can be calculated as follows:

$$\mathcal{L}_{CRP} = \sum_{x_s, x_t \in \mathcal{T}} \sum_{i=1}^{K} R(x_t, C_i^T) \, log \frac{R(x_t, C_i^T)}{R\left(x_t, C_i^S\right)} \tag{13}$$

3.7.6. OBJECTIVE FUNCTION

Combining all the loss functions ($\mathcal{L}_{WCE}$, $\mathcal{L}_{DMTKD}$, $\mathcal{L}_{CCD}$, and $\mathcal{L}_{CRP}$), total objective for the architecture weights (θ_S, ζ_S, and ζ_T) is defined as follows:

$$arg\ min\ L_{total}(\theta_S, \zeta_S, \zeta_T) = arg\ min\ (\mathcal{L}_{WCE} + \lambda_\alpha * \mathcal{L}_{DMTKD} + \lambda_\beta * \mathcal{L}_{CCD} + \lambda_\gamma * \mathcal{L}_{CRP}) \tag{14}$$

where λ_α, λ_β, and λ_γ denote hyperparameters that control the weights of $\mathcal{L}_{DMTKD}$, $\mathcal{L}_{CCD}$, and $\mathcal{L}_{CRP}$, respectively.

3.8. TRAINING, TESTING, AND PERFORMANCE EVALUATIONS

Training and testing are conducted in a 5-fold cross-validation manner. The five-fold cross-validation is performed with the following details: Each of the five folds is run with 80% training (8,012 total images) and 20% test (2,003 total images) data, shuffled over the entire dataset, with no overlap between test sets across folds. Training and testing are repeated until all folds have been used for testing.

The images from the splits are loaded and augmented twice: once for the student model and once for the teacher model. For both the train and test splits, a series of image transformations and normalization processes are applied to enhance the data. However, the test split undergoes only the essential transformations to ensure a fairer comparison. These transformations and normalization configurations are presented in Table 4. for the train split and Table 5. for the test split.

Table 4. Various configurations of transformations and normalization applied to the train split.

Technique	Configuration
Resize	[224, 224]
Random Affine	10°, [0.02, 0.02]
Random Horizontal Flip	-
Normalize	[[0.485, 0.456, 0.406], [0.229, 0.224, 0.225]]

Table 5. Various configurations of transformations and normalization applied to the test split.

Technique	Configuration
Resize	[224, 224]
Normalize	[[0.485, 0.456, 0.406], [0.229, 0.224, 0.225]]

Hyperparameters used for training, where applicable, are specified in Table 6.

Table 6. Hyperparameter settings for learning process.

Technique	Configuration
Max Epochs	100
Learning Rate Policy	One Cycle
EMA Start	20
Consistency Rampup	30
λ_{orig}	0.25
λ_{dc}	0.75 / (λ_{tckd} + λ_{nckd})
λ_{tckd}	1
λ_{nckd}	8
τ_{DMT}	4
τ_{CRC}	0.07

Evaluation is conducted using the following metrics: accuracy, precision, recall (sensitivity), specificity, F1-score, AUC, balanced accuracy, confusion matrix, and Cohen's kappa. These metrics should reflect minority class performance well.

$$Accuracy = \frac{Number\ of\ Correct\ Predictions}{Total\ Number\ of\ Predictions} \tag{14}$$

$$Precision = \frac{True\ Positives}{True\ Positives + False\ Positives} \tag{15}$$

$$Recall = \frac{True\ Positives}{True\ Positives + False\ Negatives} \tag{16}$$

$$Specificity = \frac{True\ Negatives}{True\ Negatives + False\ Positives} \tag{17}$$

$$F_1\text{-}score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{18}$$

Given N is the number of instances in a sorted set of predicted probabilities for the positive class in descending order, where for each i-th instance, $FPR[i]$ is the false positive rate at the i-th threshold, $TPR[i]$ is the true positive rate at the i-th threshold, the AUC can be approximated using trapezoidal rule as follows:

$$AUC = \sum_{i=1}^{N-1} \frac{1}{2}(FPR_{i+1} - FPR_i) \times (TPR_{i+1} + TPR_i) \tag{19}$$

An example confusion matrix is illustrated in Table 7.

Table 7. Example of confusion matrix.

	Predicted Class A	**Predicted Class B**	**Predicted Class C**
Actual Class A	83.06%*	0.33%	16.61%
Actual Class B	2.91%	97.09%*	0.00%
Actual Class C	16.67%	0.00%	83.33%*

* True positives

Cohen's kappa can be calculated as follows:

$$\kappa = \frac{p_o - p_e}{1 - p_e} \tag{20}$$

where p_o is relative observed agreement between the two raters (classifiers), whilst p_e represents the expected agreement probability, the probability that the raters agree by chance.

The experiment includes an ablation study to assess the impact of various components or parts of the architecture on performance. This study is conducted using the combinations specified in Table 8.

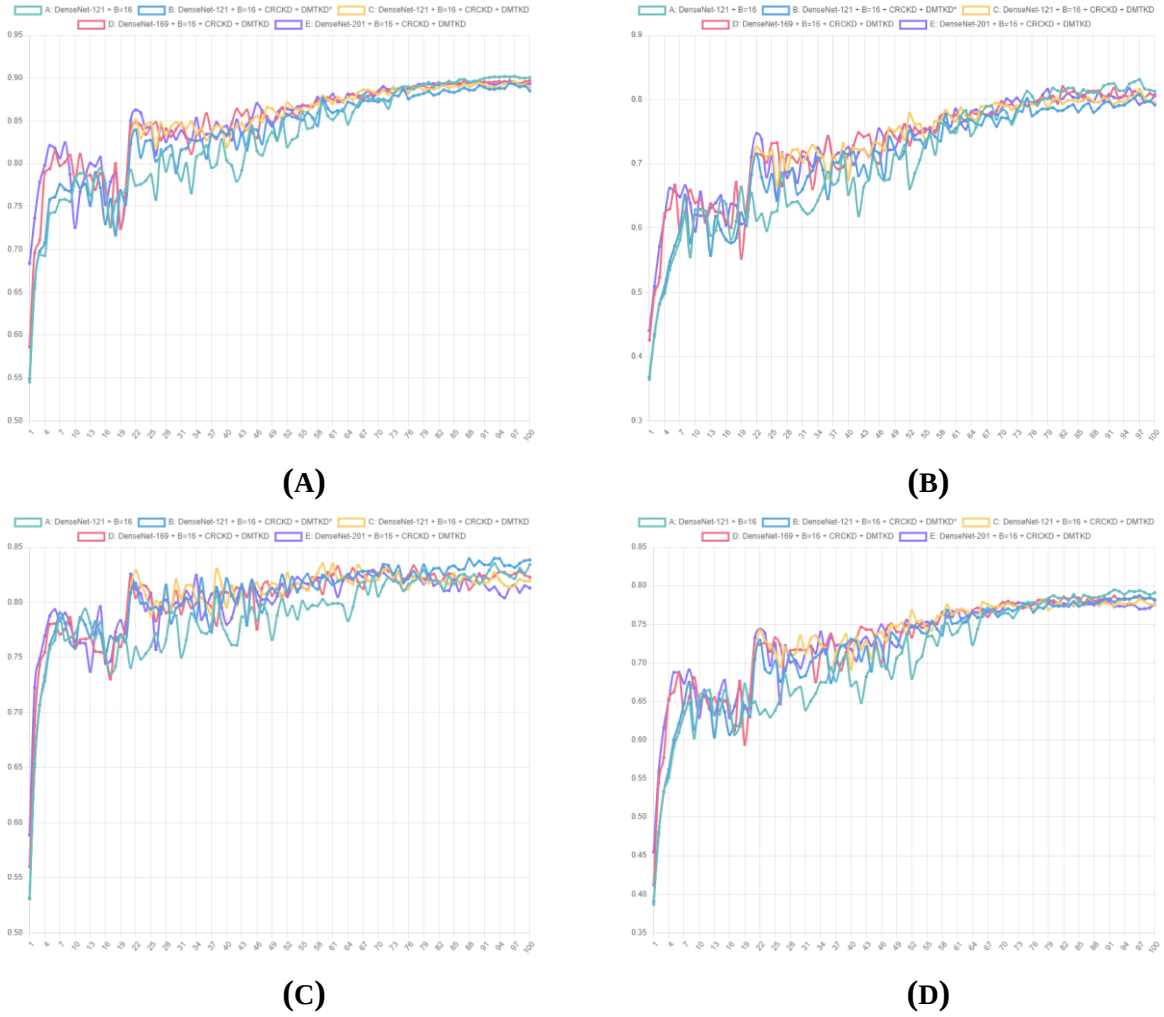
Table 8. Combination of ablation study used in the experiment.

Model	B=16	CRCKD	DMTKD	Barebone Model Variations		
				121	169	201
A	✓			✓		
B	✓	✓	*	✓		
C	✓	✓	✓	✓		
D	✓	✓	✓		✓	
E	✓	✓	✓			✓

* equals to model C using $\lambda_{orig} = 1$ and $\lambda_{dc} = 0$

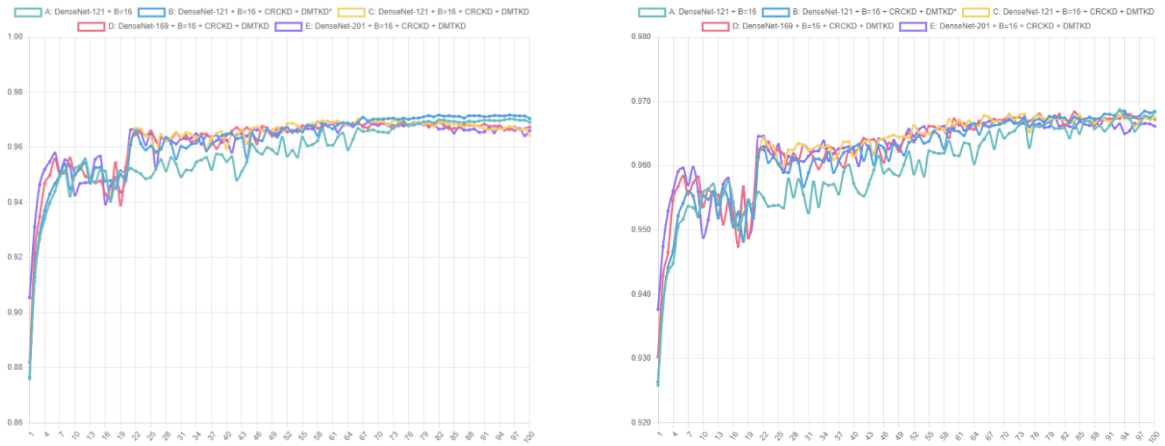
4. Average Results Of Five Cross-Validation Folds

Figure 6 displays the test results per epoch averaged over five cross-validation folds.

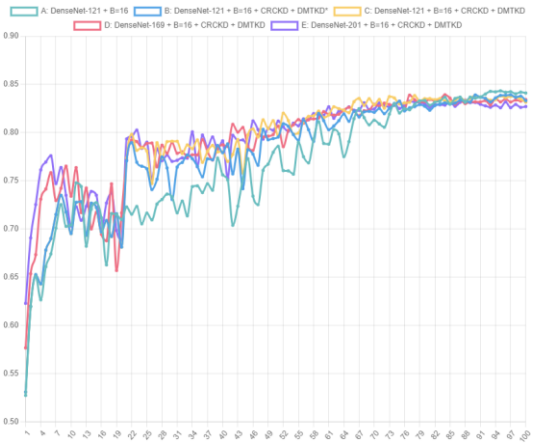


(A) **(B)**

(C) **(D)**



(E) **(F)**



(G)

Figure 6. Line graph showing the test results of each epoch, averaged over five cross-validation folds: (**a**) Accuracy; (**b**) Precision; (**c**) Recall; (**d**) F_1-score; (**e**) AUC; (**f**) Specificity; (**g**) Cohen's Kappa.

The best models for accuracy, precision, recall, F1 score, AUC, specificity, and kappa performance, respectively, based on the maximum of average of the five cross-validation folds, are as follows: Model E (89.72%), Model D (85.30%), Model B (86.11%), Model D (84.66%), Model B (98.25%), Model D (97.74%), and Model D (87.83%). These results are further summarized in Table 9.

Table 9. Maximum of average of the five cross-validation folds

Model	B=16	CRCKD	DMTKD	Barebone Model Variations			Metrics						
				121	169	201	ACC	PRE	REC	F1	AUC	SPC	KAP
A	✓			✓			**90.19%** (97)	**85.55%** (97)	84.35% (93)	**84.11%** (92)	98.23% (96)	**97.51%** (93)	**88.08%** (95)
B	✓	✓	*	✓			89.41% (96)	83.27% (97)	**84.85%** (93)	83.39% (97)	**98.41%** (96)	97.50% (93)	87.68% (97)
C	✓	✓	✓	✓			89.62% (97)	84.04% (97)	84.36% (**59**)	82.92% (97)	98.32% (67)	97.45% (**74**)	87.69% (90)
D	✓	✓	✓		✓		89.70% (100)	84.52% (**82**)	84.18% (77)	83.42% (92)	98.16% (**60**)	97.49%(84)	87.77% (**84**)
E	✓	✓	✓			✓	89.64% (**89**)	84.24% (95)	83.63% (74)	83.01% (**79**)	98.16% (68)	97.43% (80)	87.23% (88)

* equals to model C using $\lambda_{\text{orig}} = 1$ and $\lambda_{\text{dc}} = 0$

Meanwhile, Table 10 presents the average performance over the last ten epochs. The highest average values for AUC, accuracy, precision, balanced multiclass accuracy, and F1 score across the last ten epochs are achieved by Models B (98.35%), A (90.06%), A (84.43%), B (84.45%), and A (83.76), respectively. However, it should be noted that averaging performance over the last ten epochs may be susceptible to overfitting, thereby reducing its reliability as a robust performance metric. Therefore, the maximum value of the epoch-averaged performance across the five cross-validation folds should also be considered as an additional measure of model performance.

Table 10. The average of last ten epochs performance.

Model	B=16	CRCKD	DMTKD	Barebone Model Variations			Metrics				
				121	169	201	AUC	ACC	PRE	BMA	F1
A	✓			✓			98.17%	**90.06%**	**84.43%**	83.49%	**83.76%**
B	✓	✓	*	✓			**98.35%**	88.93%	81.90%	**84.45%**	82.95%
C	✓	✓	✓	✓			97.84%	89.21%	82.46%	82.69%	82.37%
D	✓	✓	✓		✓		97.94%	89.55%	83.16%	83.55%	83.06%
E	✓	✓	✓			✓	97.83%	89.39%	82.74%	82.16%	82.11%

* equals to model C using $\lambda_{\text{orig}} = 1$ and $\lambda_{\text{dc}} = 0$

The confusion matrices of models are shown in Figure 7. It shows that model A is the best NV (5) classifier, achieving the true positives of 96.14%, while model B (92.59%) is the worst. On the contrary, model D is the worst minority classes classifier with 71.34% in MEL (4).

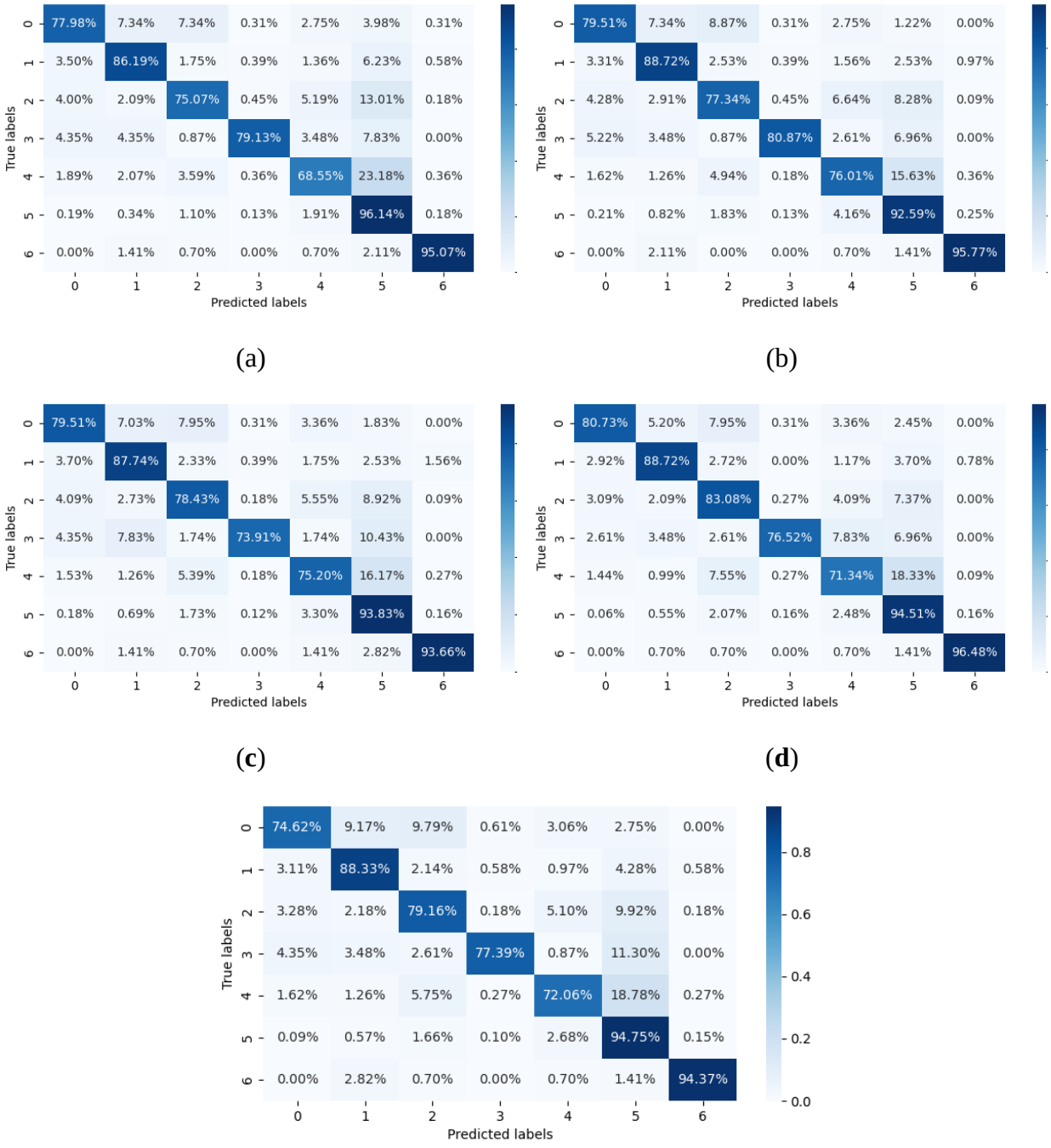


Figure 6. Confusion matrices averaged over five cross-validation folds. (**a**) Model A; (**b**) Model B; (**c**) Model C; (**d**) Model D; (**e**) Model E.

5. DISCUSSIONS

Table 11 demonstrates that reducing the batch size to 16 has empirically led to improved performance. This improvement is significant, as all models in this experiment outperform the previously established state-of-the-art mean teacher methods evaluated on the HAM10000 dataset. Notably, even a basic mean teacher approach based on DenseNet121, without any additional modules (Model A), exceeds prior performance benchmarks.

Table 11. Comparison with similar state-of-the-art mean teacher methods.

Methods	B	ACC	AP	BMA	F1
Liu et. al. [19]	48	84.73%	73.88%	76.55%	74.63%
Xing et. al. [23]	64	85.66%	76.35%	78.07%	76.45%
Model A	16	**90.06%**	**84.43%**	83.49%	**83.76%**
Model B	16	88.93%	81.90%	**84.45%**	82.95%

The increase in barebone model size does improve performance up to a certain point. It is found that the medium-sized barebone model used in Model D offers the best overall performance, as shown in Table 12. This improvement is reflected in higher values for accuracy, precision, F1-score, specificity, and kappa. However, there is a slight decrease in recall and AUC performance, a trend that becomes more pronounced in the larger barebone Model E. Nevertheless, the overall performance gain outweighs the associated penalties for Model D.

Table 12. Maximum of average model performance across folds.

Model	Metrics						
	ACC	PRE	REC	F1	AUC	SPC	KAP
C	89.62%	84.04%	**84.36%**	82.92%	**98.32%**	97.45%	87.69%
D	**89.70%** **(+0.08)**	**84.52%** **(+0.48)**	84.18% (-0.18)	**83.42%** **(+0.50)**	98.16% (-0.16)	**97.49%** **(+0.04)**	**87.77%** **(+0.08)**
E	89.64% (+0.02)	84.24% (+0.20)	83.63% (-0.73)	83.01% (+0.09)	98.16% (-0.16)	97.43% (-0.02)	87.23% (-0.46)

Table 10. highlights the difference in accuracy and balanced multi-class performance between

model A and B. Model A performs best for majority class classification, while model B is superior for minority class classification. The improvement in multi-class accuracy of the CRCKD module comes at the expense of overall accuracy. Model A achieves a balanced multi-class accuracy of 83.49% compared to 84.45% for model B, representing a 0.96% improvement for model B.

However, this improvement comes at the expense of overall accuracy. Model A achieves an accuracy of 90.06%, compared to 88.93% for Model B, resulting in a performance difference of −1.13%. In other words, enhancing performance on the minority class in this case leads to a decline in performance on the majority class.

This trade-off is also evident when comparing the confusion matrix results between the model that employs CRCKD (Figure 6(a)) and the model without CRCKD (Figure 6(b)) for the NV (5) class, where the respective performances are 96.14% and 92.59%. This corresponds to a 3.55% decrease in performance for the majority class (NV).

It is shown that adjusting the CRCKD model with DMTKD enhances the model's performance profile adjustability. Model B is equivalent to Model C under DMTKD settings where $\lambda_{\text{orig}} = 1$ and $\lambda_{\text{dc}} = 0$, while Model C uses $\lambda_{\text{orig}} = 0.2$ and $\lambda_{\text{dc}} = 0.75/(\lambda_{tckd}+\lambda_{nckd})$, with $\lambda_{tckd} = 1$, and $\lambda_{nckd} = 8$. As shown in Table 13, when greater emphasis on majority (NV) classification performance is desired, decreasing λ_{orig} and increasing λ_{dc} can generally shift the performance profile toward the majority class, as demonstrated by Model C.

Table 13. Comparison of majority and minority class accuracy between model B and model C.

Model	Majority	Minority					
	NV	**AKIEC**	**BCC**	**BKL**	**DF**	**MEL**	**VASC**
B	92.59%	79.51%	**88.72%**	77.34%	**80.87%**	**76.01%**	95.77%
C	93.83% (+1.24%)	79.51% (0.00)	87.74% (-0.98%)	78.43% (+1.09%)	73.91% (-6.96%)	75.20% (-0.81%)	93.83% (-1.94%)
D	**94.51%** **(+1.92%)**	**80.73%** **(+1.22%)**	88.72% (0.00)	**83.08%** **(+5.74%)**	76.52% (-4.35%)	71.34% (-4.67%)	**96.48%** **(+0.71%)**

When the DMTKD weights of Model C are combined with an increase in barebone model size to form Model D, the performance penalty previously observed in the minority class is largely mitigated. For instance, performance improvements are seen across several classes: BCC increases from 87.74% to 88.72%, BKL from 78.43% to 83.08%, DF from 73.91% to 76.52%, and VASC from 93.83% to 96.48%. The only exception is MEL, which experiences a decline from 75.20% to 71.34%. This combination of architectural decisions and DMTKD outperforms Model B when considering how many minority classes achieve improved performance (3 vs. 2). Additionally, there is a modest improvement in majority class (NV) performance, increasing from 93.83% to 94.51%. These results demonstrate that the proposed method—combining architectural enhancements with DMTKD—effectively improves performance for both majority and minority classes.

The average number of epochs required to reach maximum performance is summarized in Table 14. An additional finding is that the inclusion of λ_dc in DMTKD noticeably accelerates convergence. This is evident in the reduction in required epochs—from 94.71 in Model A and 95.57 in Model B to 83.00, 82.71, and 81.86 in Models C, D, and E, respectively. This suggests that decoupling may introduce additional implicit knowledge transfer mechanisms that facilitate faster distillation. In contrast, increasing the model size only marginally affects convergence speed, as evidenced by the relatively small differences among Models C, D, and E.

Table 14. Average epochs of convergence to maximum performance.

Model	Metrics							
	ACC	**PRE**	**REC**	**F1**	**AUC**	**SPC**	**KAP**	**AVGC**
A	97	97	93	92	96	93	95	94.71
B	96	97	93	97	96	93	97	95.57
C	97	97	**59**	97	67	**74**	90	83.00
D	100	**82**	77	92	**60**	84	**84**	82.71
E	**89**	95	74	**79**	68	80	88	**81.86**

Lastly, Table 15 presents a comparative analysis between the proposed method and existing state-of-the-art approaches. In this comparison, our proposed method demonstrates superior performance in terms of accuracy, specificity, and AUC, achieving values of 89.41%, 97.5%, and 98.41%, respectively.

Table 15. A comparative analysis of maximum model performance conducted against several state-of-the-art methods.

Method	Maximum Model Performance					
	Accuracy	**Precision**	**Recall**	**Specificity**	**F_1-score**	**AUC**
[17]	72.1%					
[18]	84.0%		81.0%	88.0%		
[14]	92.4%	92.08%		90%	89.16%	
[19]						
[20]	87.7%				83.0%	
[21]						97%
[22]	85.3%	88.2%	92.0%			
[23]						
[24]	89.66%	88.76%	89.57%		89.02%	
[25]	83.1%		83%		83%	
[26]	85.8%					
[27]	86.7%					
[28]	91%				88.1%	
[29]	86.3%		86.3%	97.7%		
[30]	77.4%	74.7%	66.5%	95%	68.4%	
[31]	85.98%	84%	86%		85.98%	
[32]	90%	86%	81%		86%	99%
[33]	81.0%					
[13]	94.17%	94.28%	93.76%		93.93%	99.42%
[34]	95%	95%	95%		95%	
Proposed Method B	**89.41%**	**83.27%**	**84.85%**	**97.50%**	**83.39%**	**98.41%**

Table 16. A comparative analysis based on 5-fold cross-validation conducted against several state-of-the-art methods.

Method	5-fold Cross Validation			
	Accuracy	**AP**	**BMA**	**F_1-score**
[17]				
[18]				
[14]				
[19]	84.73%	73.88%	76.55%	74.63%
[20]				
[21]				
[22]				
[23]	85.66%	76.35%	78.07%	76.45%
[24]				
[25]				
[26]				
[27]				
[28]				
[29]				
[30]				
[31]				
[32]				
[33]				
[13]				
[34]				
Proposed Method B	**88.93%**	**81.90%**	**84.45%**	**82.95%**

5. CONCLUSIONS

A novel CRCDKD architecture is proposed for medical skin cancer classification. This architecture integrates the mean teacher framework and combines CRCKD with empirically validated architectural choices to achieve high overall performance. Additionally, a newly introduced DMTKD strategy—improved from DKD—is incorporated to enhance the flexibility of adjusting the performance profile between minority and majority classes, allowing adaptation to various application requirements. The resulting architecture demonstrates both high performance and adjustability, as reflected in both general performance metrics and those specifically accounting for the minority class on the medical skin disease dataset HAM10000.

This study has limitations in the range of experimental conditions explored. Future work may investigate a broader range of DMTKD weight values to further validate the effects of varying weight configurations in combined coupled and decoupled distillation settings. Additionally, exploring dynamic adjustment of DMTKD weights during training could offer further insights into performance optimization. Future research may also assess the generalizability of the CRCDKD architecture across a wider variety of medical datasets or backbone architectures. Exploring advanced data augmentation techniques—such as stable diffusion methods for addressing class imbalance—could also provide valuable improvements in model training and performance.

ABBREVIATIONS

The following abbreviations are used in this manuscript:

ACC	Accuracy
AP	Average Precision
BMA	Balanced Multi Class Accuracy
F1	F1-Score
AUC	Area Under The Receiver Operating Characteristic Curve
SPC	Specificity
KAP	Cohen’s Kappa
AVGC	Averaged Speed of Convergence

DATA AVAILABILITY

Data supporting this study are openly available from Harvard Dataverse (Tschandl et al., 2018 [10]) at https://doi.org/10.7910/DVN/DBW86T.

AUTHOR CONTRIBUTION

Franky Setiawan: Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Resources, Data Curation, Writing – Original Draft, Writing – Review & Editing, Visualization. **Benfano Soewito**: Supervision, Project administration, Funding acquisition.

CONFLICT OF INTERESTS

The authors declare that there is no conflict of interests.

REFERENCES

[1] A. Svensson, R. Ofenloch, M. Bruze, L. Naldi, S. Cazzaniga, P. Elsner, M. Goncalo, M. Schuttelaar, T. Diepgen, Prevalence of Skin Disease in a Population-Based Sample of Adults From Five European Countries, Br. J. Dermatol. 178 (2018), 1111-1118. https://doi.org/10.1111/bjd.16248.

[2] M. Richard, C. Paul, T. Nijsten, P. Gisondi, C. Salavastru, C. Taieb, M. Trakatelli, L. Puig, A. Stratigos, Prevalence of Most Common Skin Diseases in Europe: a Population‐based Study, J. Eur. Acad. Dermatol. Venereol. 36 (2022), 1088-1096. https://doi.org/10.1111/jdv.18050.

[3] V. Lewis, A.Y. Finlay, 10 Years Experience of the Dermatology Life Quality Index (DLQI), J. Invest. Dermatol. Symp. Proc. 9 (2004), 169-180. https://doi.org/10.1111/j.1087-0024.2004.09113.x.

[4] M. Goyal, T. Knackstedt, S. Yan, S. Hassanpour, Artificial Intelligence-Based Image Classification Methods for Diagnosis of Skin Cancer: Challenges and Opportunities, Comput. Biol. Med. 127 (2020), 104065. https://doi.org/10.1016/j.compbiomed.2020.104065.

[5] M.A. Mamun, M.S. Kabir, M. Akter, M.S. Uddin, Recognition of Human Skin Diseases Using Inception-V3 with Transfer Learning, Int. J. Inf. Technol. 14 (2022), 3145-3154. https://doi.org/10.1007/s41870-022-01050-4.

[6] F. Bozkurt, Skin Lesion Classification on Dermatoscopic Images Using Effective Data Augmentation and Pre-Trained Deep Learning Approach, Multimed. Tools Appl. 82 (2022), 18985-19003. https://doi.org/10.1007/s11042-022-14095-1.

[7] I.K.E. Purnama, A.K. Hernanda, A.A.P. Ratna, I. Nurtanio, A.N. Hidayati, M.H. Purnomo, S.M.S. Nugroho, R.F. Rachmadi, Disease Classification Based on Dermoscopic Skin Images Using Convolutional Neural Network in Teledermatology System, in: 2019 International Conference on Computer Engineering, Network, and Intelligent Multimedia (CENIM), IEEE, Surabaya, Indonesia, 2019: pp. 1–5. https://doi.org/10.1109/cenim48368.2019.8973303.

[8] E. Cengil, A. Çinar, M. Yildirim, Hybrid Convolutional Neural Network Architectures for Skin Cancer Classification, Eur. J. Sci. Technol. 28 (2021), 694-701. https://doi.org/10.31590/ejosat.1010266.

[9] M.A. Khan, M.Y. Javed, M. Sharif, T. Saba, A. Rehman, Multi-Model Deep Neural Network Based Features Extraction and Optimal Selection Approach for Skin Lesion Classification, in: 2019 International Conference on Computer and Information Sciences (ICCIS), IEEE, Sakaka, Saudi Arabia, 2019: pp. 1–7. https://doi.org/10.1109/iccisci.2019.8716400.

[10] P. Tschandl, C. Rosendahl, H. Kittler, The Ham10000 Dataset, a Large Collection of Multi-Source Dermatoscopic Images of Common Pigmented Skin Lesions, Sci. Data 5 (2018), 180161. https://doi.org/10.1038/sdata.2018.161.

[11] R. Singh, T. Ahmed, A. Kumar, A.K. Singh, A.K. Pandey, S.K. Singh, Imbalanced Breast Cancer Classification Using Transfer Learning, IEEE/ACM Trans. Comput. Biol. Bioinform. 18 (2021), 83-93. https://doi.org/10.1109/tcbb.2020.2980831.

[12] S. Decherchi, E. Pedrini, M. Mordenti, A. Cavalli, L. Sangiorgi, Opportunities and Challenges for Machine Learning in Rare Diseases, Front. Med. 8 (2021), 747612. https://doi.org/10.3389/fmed.2021.747612.

[13] M. Tahir, A. Naeem, H. Malik, J. Tanveer, R.A. Naqvi, S. Lee, Dscc_net: Multi-Classification Deep Learning Models for Diagnosing of Skin Cancer Using Dermoscopic Images, Cancers 15 (2023), 2179. https://doi.org/10.3390/cancers15072179.

[14] J. Almaraz-Damian, V. Ponomaryov, S. Sadovnychiy, H. Castillejos-Fernandez, Melanoma and Nevus Skin Lesion Classification Using Handcraft and Deep Learning Feature Fusion via Mutual Information Measures, Entropy 22 (2020), 484. https://doi.org/10.3390/e22040484.

[15] N. Khasawneh, M. Fraiwan, L. Fraiwan, B. Khassawneh, A. Ibnian, Detection of COVID-19 From Chest X-Ray Images Using Deep Convolutional Neural Networks, Sensors 21 (2021), 5940. https://doi.org/10.3390/s21175940.

[16] P. Naga Srinivasu, T. Srinivasa Rao, A.M. Dicu, C.A. Mnerie, I. Olariu, A Comparative Review of Optimisation Techniques in Segmentation of Brain Mr Images, J. Intell. Fuzzy Syst. 38 (2020), 6031-6043. https://doi.org/10.3233/jifs-179688.

[17] A.D. Andronescu, D.I. Nastac, G.S. Tiplica, Skin Anomaly Detection Using Classification Algorithms, in: 2019 IEEE 25th International Symposium for Design and Technology in Electronic Packaging (SIITME), IEEE, Cluj-Napoca, Romania, 2019: pp. 299–303. https://doi.org/10.1109/siitme47687.2019.8990764.

[18] A. Ameri, A Deep Learning Approach to Skin Cancer Detection in Dermoscopy Images, J. Biomed. Phys. Eng. 10 (2020), 801-806. https://doi.org/10.31661/jbpe.v0i0.2004-1107.

[19] Q. Liu, L. Yu, L. Luo, Q. Dou, P.A. Heng, Semi-supervised Medical Image Classification with Relation-Driven Self-Ensembling Model, IEEE Trans. Med. Imaging 39 (2020), 3429-3440. https://doi.org/10.1109/tmi.2020.2995518.

[20] K. Thurnhofer-Hemsi, E. Domínguez, A Convolutional Neural Network Framework for Accurate Skin Cancer Detection, Neural Process. Lett. 53 (2020), 3073-3093. https://doi.org/10.1007/s11063-020-10364-y.

[21] V. Miglani, M. Bhatia, Skin Lesion Classification: A Transfer Learning Approach Using EfficientNets, in: Advances in Intelligent Systems and Computing, Springer, Singapore, 2021: pp. 315–324. https://doi.org/10.1007/978-981-15-3383-9_29.

[22] P.N. Srinivasu, J.G. SivaSai, M.F. Ijaz, A.K. Bhoi, W. Kim, J.J. Kang, Classification of Skin Disease Using Deep Learning Neural Networks with MobileNet V2 and LSTM, Sensors 21 (2021), 2852. https://doi.org/10.3390/s21082852.

[23] X. Xing, Y. Hou, H. Li, Y. Yuan, H. Li, M.Q.-. Meng, Categorical Relation-Preserving Contrastive Knowledge Distillation for Medical Image Classification, arXiv:2107.03225 (2021). http://arxiv.org/abs/2107.03225v1.

[24] S. Jain, U. Singhania, B. Tripathy, E.A. Nasr, M.K. Aboudaif, A.K. Kamrani, Deep Learning-Based Transfer Learning for Classification of Skin Cancer, Sensors 21 (2021), 8142. https://doi.org/10.3390/s21238142.

[25] S.S. Chaturvedi, K. Gupta, P.S. Prasad, Skin Lesion Analyser: An Efficient Seven-Way Multi-Class Skin Cancer Classification Using MobileNet, in: Advances in Intelligent Systems and Computing, Springer, Singapore, 2021: pp. 165–176. https://doi.org/10.1007/978-981-15-3383-9_15.

[26] H. Huang, B.W. Hsu, C. Lee, V.S. Tseng, Development of a Light-Weight Deep Learning Model for Cloud Applications and Remote Diagnosis of Skin Cancers, J. Dermatol. 48 (2020), 310-316. https://doi.org/10.1111/1346-8138.15683.

[27] D. Popescu, M. El-khatib, L. Ichim, Skin Lesion Classification Using Collective Intelligence of Multiple Neural Networks, Sensors 22 (2022), 4399. https://doi.org/10.3390/s22124399.

[28] T.M. Alam, K. Shaukat, W.A. Khan, I.A. Hameed, L.A. Almuqren, M.A. Raza, M. Aslam, S. Luo, An Efficient Deep Learning-Based Skin Cancer Classifier for an Imbalanced Dataset, Diagnostics 12 (2022), 2115. https://doi.org/10.3390/diagnostics12092115.

[29] L. Hoang, S. Lee, E. Lee, K. Kwon, Multiclass Skin Lesion Classification Using a Novel Lightweight Deep Learning Framework for Smart Healthcare, Appl. Sci. 12 (2022), 2677. https://doi.org/10.3390/app12052677.

[30] M. Fraiwan, E. Faouri, On the Automatic Detection and Classification of Skin Cancer Using Deep Transfer Learning, Sensors 22 (2022), 4963. https://doi.org/10.3390/s22134963.

[31] G. Alwakid, W. Gouda, M. Humayun, N.U. Sama, Melanoma Detection Using Deep Learning-Based Classifications, Healthcare 10 (2022), 2481. https://doi.org/10.3390/healthcare10122481.

[32] V.D. Nguyen, N.D. Bui, H.K. Do, Skin Lesion Classification on Imbalanced Data Using Deep Learning with Soft Attention, Sensors 22 (2022), 7530. https://doi.org/10.3390/s22197530.

[33] H.K. Gajera, D.R. Nayak, M.A. Zaveri, A Comprehensive Analysis of Dermoscopy Images for Melanoma

Detection via Deep Cnn Features, Biomed. Signal Process. Control. 79 (2023), 104186. https://doi.org/10.1016/j.bspc.2022.104186.

[34] M.M. Hossain, M.M. Hossain, M.B. Arefin, F. Akhtar, J. Blake, Combining State-Of-The-Art Pre-Trained Deep Learning Models: a Noble Approach for Skin Cancer Detection Using Max Voting Ensemble, Diagnostics 14 (2023), 89. https://doi.org/10.3390/diagnostics14010089.

[35] A. Tarvainen, H. Valpola, Mean Teachers Are Better Role Models: Weight-Averaged Consistency Targets Improve Semi-Supervised Deep Learning Results, arXiv:1703.01780 (2017). http://arxiv.org/abs/1703.01780v6.

[36] B. Zhao, Q. Cui, R. Song, Y. Qiu, J. Liang, Decoupled Knowledge Distillation, arXiv:2203.08679 (2022). http://arxiv.org/abs/2203.08679v2.