

Comparative Analysis of Transfer Learning Models with Explainable AI for Automated Nail Disease Classification

Satyam Mittal

School of Computer Science
and Engineering (SCOPE)
Vellore Institute of Technology
Vellore, Tamil Nadu, India

Shashwat Singh

School of Computer Science
and Engineering (SCOPE)
Vellore Institute of Technology
Vellore, Tamil Nadu, India

Progya Paromita Biswas

School of Computer Science
and Engineering (SCOPE)
Vellore Institute of Technology
Vellore, Tamil Nadu, India

Abstract—Nail disorders change the color, texture and shape of the nail, so nail photographs are a possible input for computer aided screening. This paper compares six pretrained networks, ResNet-50, VGG16, DenseNet-121, ConvNeXt-Tiny, ViT-B/16 and DINOv2, under frozen, partial and full fine-tuning on a small, imbalanced dataset of 1,008 images in 13 classes. Images are split into five grouped folds so that near duplicates never cross folds, every configuration receives the same tuning budget on one fold, and all results come from the other four folds with group bootstrap confidence intervals. Fine-tuning beat frozen features for every network by 11.6 to 18.5 accuracy points, and partial fine-tuning matched full fine-tuning for three networks. The best Macro-F1 was 0.857 for DINOv2 with full fine-tuning, but five other configurations, including ResNet-50 with 3.6 times fewer parameters, were statistically tied with it. The backbone learning rate mattered most, and the tuning fold overstated performance by about five Macro-F1 points. Grad-CAM and attention rollout maps of the CNNs and DINOv2 were more faithful than random maps and ignored image padding, unlike ViT-B/16, and hiding watermarks lowered the predicted probability more than hiding random regions. The models are screening aids, not diagnostic tools.

Keywords—Nail disease classification; transfer learning; fine-tuning strategies; vision transformers; DINOv2; small medical datasets; explainable AI; Grad-CAM

I. INTRODUCTION

The appearance of the nail changes in many local nail disorders and in some systemic conditions, but the visible signs overlap and access to dermatological assessment is uneven. Image-based machine learning could support low-cost screening, and transfer learning is the usual way to obtain accurate models when labeled medical images are scarce [1], [2].

Most nail-image studies fine-tune one network and report one accuracy figure on one split [3], [4]. Three issues limit what such results show. First, the best way to adapt a pretrained network (freeze it, unfreeze its last stage, or train all layers) depends on the architecture and on the amount of data, and large benchmarks report mixed outcomes [2], [5], [6]. Second, networks are often compared at one shared hyperparameter setting, which can measure how well that setting suits each network rather than the network itself. Third, web-collected medical images contain near duplicates, watermarks and source-specific styles that can leak across a random split or act as shortcuts [7], [8].

This paper compares six pretrained networks from four families under three transfer strategies, 18 configurations in total, on a 13-class nail disease dataset of 1,008 images. Its contributions are: (i) a controlled comparison in which every configuration receives the same tuning budget; (ii) a grouped-fold protocol that separates the fold used for tuning from the folds used for reporting, with group-level bootstrap intervals and paired tests; (iii) class-wise and source-wise analysis, including a prediction-level check for a watermark cue; and (iv) an explainability analysis of the best configurations covering faithfulness, padding saliency, agreement between models and a watermark occlusion test.

II. RELATED WORK

Transfer learning reuses features learned on a large source dataset: early layers capture general patterns while later layers become task specific [1]. Kornblith et al. found that ImageNet accuracy correlates with transfer accuracy for fine-tuned networks [5], whereas Raghu et al. found limited gains from ImageNet pretraining on medical images and comparable results from small architectures [2]. On MedMNIST+, Doerrich et al. reported that end-to-end training beat linear probing and that CNNs remained competitive with vision transformers [6]. The architectures studied here span these families: ResNet-50 [9], VGG16 [10], DenseNet-121 [11], ConvNeXt-Tiny [12], the supervised ViT-B/16 [13], [14], and DINOv2, a self-supervised transformer trained on 142 million curated images [15].

For nail images, Coşar Soğukkuyu and Ata fine-tuned VGG16 and VGG19 on 723 images of three conditions and reported 94 and 93 percent accuracy [3], and Shandilya et al. proposed a hybrid capsule network for six categories with heavy augmentation, reporting 99.25 percent validation accuracy [4]. Class sets, dataset sizes and evaluation protocols differ, so these figures are not comparable with the values reported here. Shortcut learning, in which a model exploits features that merely correlate with the label, has been documented in medical imaging [7], [8]. Post-hoc explanations such as Grad-CAM [16] and attention rollout [17] show where a model looks but do not prove that it reasons correctly, and their reliability should itself be checked [18].

III. DATASET AND EVALUATION PROTOCOL

A. Dataset

The dataset contains 1,008 RGB images in 13 classes (Table I), collected from four online sources labeled BASE (543 images), archive6 (232), archive5 (158) and B1B2CD (75) [19]-[22]. Four classes (clubbing, onychomycosis, psoriasis and healthy) hold 150 images each and are drawn from several sources, whereas the other nine classes (38 to 62 images) come only from BASE. The largest to smallest class ratio is 3.95 and 46.1 percent of the images come from sources other than BASE. Two further classes (acral lentiginous melanoma and onychogryphosis) were excluded because all their images came from one augmented source, and images with conflicting labels between sources were removed. The images are PNG files with a long side of at most 160 pixels. Labels come from the web sources and no pathology confirmation is available. Source, image size and watermarks are partly recognizable: 21 of the 40 BASE psoriasis images carry a visible watermark or overlay.

TABLE I
 CLASSES, IMAGES BY SOURCE AND MEAN RECALL OF THE TIED CONFIGURATIONS

Class	Images	BASE	Other	Recall
Clubbing	150	39	111	0.95
Healthy	150	56	94	0.98
Onychomycosis	150	0	150	0.86
Psoriasis	150	40	110	0.87
Splinter hemorrhage	62	62	0	0.83
Onycholysis	50	50	0	0.63
Bluish nail	49	49	0	0.92
Darier's disease	45	45	0	0.89
Eczema	43	43	0	0.83
Alopecia areata	42	42	0	0.85
Beau's lines	41	41	0	0.79
Half-and-half nails	38	38	0	0.75
Koilonychia	38	38	0	0.77

B. Grouped folds and two-phase protocol

Images were assigned to five stratified group folds of 201, 196, 201, 204 and 206 images. A group is a cluster of near-duplicate images or a set of images sharing a source stem; there are 959 groups and none crosses a fold. Groups are not patients, so patient-level separation is not established. In the tuning phase, fold 0 is the test fold and each configuration is trained with a default setting and three trials. In the confirmation phase the selected setting is retrained and tested on each of folds 1 to 4. All reported results come from the confirmation phase (807 pooled images), so no test image of a reported fold influenced hyperparameter selection. The dataset is too small for a separate validation set; every run trains for a fixed number of epochs and the final-epoch model is evaluated.

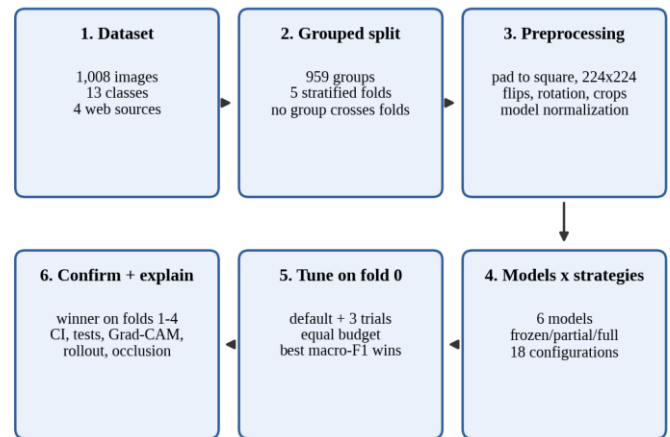


Fig. 1. Study pipeline: grouped split, preprocessing, six networks with three strategies, tuning on fold 0, confirmation and explanation on folds 1 to 4.

IV. METHODOLOGY

A. Networks and transfer strategies

Six networks were loaded through the timm library [23] with pretrained weights (torchvision ImageNet-1k weights for ResNet-50, VGG16 and DenseNet-121, ImageNet-1k weights for ConvNeXt-Tiny and for ViT-B/16 trained with the AugReg recipe, and LVD-142M self-supervised weights for DINOv2), and the classifier was replaced by a 13-way linear layer. Under the frozen strategy only the new classifier is trained, and frozen BatchNorm layers stay in evaluation mode. Under partial fine-tuning the classifier and the last stage are trained (Table II); under full fine-tuning all parameters are trained. The same rule unfreezes very different shares of each network.

TABLE II
 NETWORKS AND PARTIAL FINE-TUNING

Network	Partial unfreezes	Params (M)	Partial trainable
DINOv2	last 4 of 12 blocks + norm	85.7	28.4 (33%)
ViT-B/16	last 4 of 12 blocks + norm	85.8	28.4 (33%)
ResNet-50	layer4	23.5	15.0 (64%)
DenseNet-121	dense block 4 + norm5	7.0	2.2 (31%)
ConvNeXt-Tiny	stage 4 + head norm	27.8	15.5 (56%)
VGG16	conv block 5 (fc layers frozen)	134.3	7.1 (5%)

B. Preprocessing and training

Images are padded to a square with the normalization mean as fill color, resized to 224 by 224 and normalized with each network's own statistics. Training augmentation uses random resized crops (minimum scale 0.70), horizontal and vertical flips, rotation up to 15 degrees and mild brightness and contrast jitter without hue change, because color defines several classes. All runs use AdamW [24] with separate learning rates for the new head and the unfrozen backbone layers, weight decay 0.01, linear warm-up followed by cosine decay [25], label smoothing 0.1 [26], gradient clipping at norm 1.0, FP16 mixed precision [27], batch size 32, weighted cross-entropy with inverse-frequency class weights, and 30, 40 and 50 epochs for frozen, partial and full fine-tuning. The default head learning rate is 1e-3; default backbone rates range from 1e-5 to 1e-4 depending on the network. Experiments ran on a Kaggle Tesla T4 GPU with PyTorch 2.11.0 [28], timm 1.0.29 and scikit-learn 1.6.1, seed 42; the 144 training runs took about 9.5 GPU hours.

C. Hyperparameter search

Every configuration received exactly four fold-0 runs: its default and three trials. For frozen heads the trials raised the head learning rate threefold, additionally trained for 60 epochs, and additionally used lighter augmentation. For partial and full fine-tuning the trials multiplied the backbone learning rate by 0.3 and by 3 and tested a stronger regularization recipe (minimum crop scale 0.5, rotation 25 degrees, jitter 0.15, weight decay 0.05, dropout 0.3). The trial with the highest fold-0 Macro-F1 was selected, but only if it beat the default by at least 0.01 (about two test images).

D. Evaluation and statistics

Macro-F1 is the primary metric; accuracy and balanced accuracy are also reported. Metrics use the pooled predictions of folds 1 to 4. Confidence intervals are 95 percent bootstrap intervals that resample whole groups [29]. Paired comparisons use the exact McNemar test on per-image correctness [30]; the six planned full-versus-partial comparisons use Holm correction [31]. Two configurations are called tied when the interval of their paired accuracy difference includes zero.

E. Explainability analysis

Explanations were computed for five of the six configurations tied for the best (DINOv2 full, ResNet-50 full and partial, ViT-B/16 full, DenseNet-121 full) and for frozen ViT-B/16 as a reference. Each image was explained by the checkpoint that did not train on it, so all 1,008 images were explained out of fold; recomputed predictions matched the stored ones for 99.9 to 100 percent of images. Grad-CAM [16] was taken for the predicted class at the last convolutional stage of the CNNs and at the last-block patch tokens of ViT-B/16 and DINOv2, and attention rollout [17] was added for the two transformers. Faithfulness used deletion and insertion curves [32]: pixels are removed from, or added to, a mean-color image in order of saliency in 21 steps, and the area under the predicted-class probability curve is compared with that of a random map (gain = difference, positive is better than random). The padding saliency ratio is the share of saliency on the padding divided by the padding's area share (1 means uniform). As a sanity check [18], Grad-CAM maps were compared by Spearman correlation with maps of a randomly initialized network. For the 21 watermarked images (14 DermNet logos, 4 Te Whatu Ora texts, 2 iStock credits and 1 video-player overlay, all BASE psoriasis), boxes were first proposed by an AI assistant (Claude, Anthropic) and checked on magnified crops. Occluding a box with the mean color was compared with five random boxes of the same size using a one-sided Wilcoxon signed-rank test [33] with Holm correction. No expert annotation of the nail region was available, so overlap with the nail was not measured.

V. RESULTS AND DISCUSSION

A. Overall comparison

Table III and Fig. 2 give pooled results on folds 1 to 4. Frozen networks reach Macro-F1 of 0.59 to 0.68, and fine-tuned networks reach 0.73 to 0.86. The best configuration is DINOv2 with full fine-tuning (Macro-F1 0.857, accuracy 0.881, balanced accuracy 0.846). The 95 percent interval of its accuracy advantage includes zero for DINOv2 partial (+0.1

points), ViT-B/16 full (+0.6), ResNet-50 full (+1.0), ResNet-50 partial (+2.2) and DenseNet-121 full (+2.1), so six configurations form a statistical tie at the top. All other configurations are significantly lower. A metadata-only random forest on image size and aspect ratio reached 20.2 percent accuracy and Macro-F1 0.171 under the same folds, and the majority class gives 14.9 percent accuracy.

TABLE III
 POOLED MACRO-F1 ON FOLDS 1 TO 4 (95% INTERVAL); * STATISTICALLY TIED WITH THE BEST

Network	Frozen	Partial	Full
DINOv2	0.671 (0.63-0.71)	0.856 (0.82-0.89)*	0.857 (0.82-0.89)*
ViT-B/16	0.677 (0.64-0.71)	0.819 (0.78-0.85)	0.852 (0.82-0.88)*
ResNet-50	0.647 (0.61-0.68)	0.835 (0.80-0.87)*	0.851 (0.82-0.88)*
DenseNet-121	0.600 (0.56-0.64)	0.792 (0.75-0.83)	0.831 (0.79-0.86)*
ConvNeXt-Tiny	0.663 (0.62-0.70)	0.814 (0.78-0.85)	0.808 (0.77-0.84)
VGG16	0.592 (0.55-0.63)	0.734 (0.69-0.77)	0.793 (0.75-0.82)

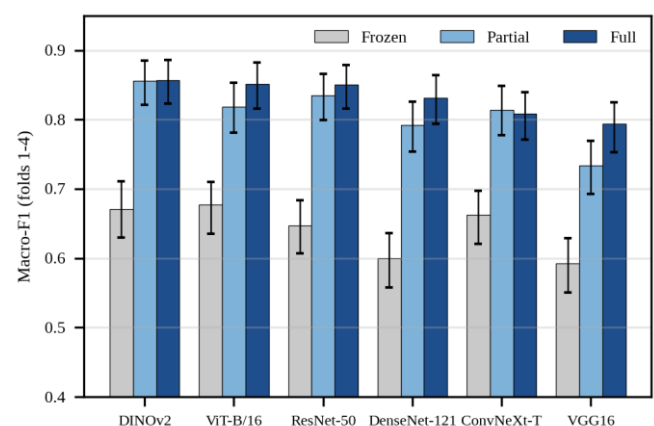


Fig. 2. Macro-F1 on the pooled held-out folds by network and strategy. Error bars are 95 percent group-bootstrap intervals.

B. Frozen, partial and full fine-tuning

Partial fine-tuning exceeded the frozen backbone by 12.0 to 16.1 accuracy points and full fine-tuning by 11.6 to 18.5 points within every network (all McNemar p below 0.0001). Full fine-tuning was not better than partial fine-tuning for DINOv2 (+0.1 points), ConvNeXt-Tiny (-0.4) or ResNet-50 (+1.2), but was significantly better for VGG16 (+4.7) and DenseNet-121 (+3.3) after Holm correction (Table IV); ViT-B/16 (+2.5) was borderline (Holm p = 0.090). VGG16 partial unfreezes only the last convolutional block (about 5 percent of its parameters), which may explain its gap. Because the same rule unfreezes different shares of each network, how much to unfreeze should be reported as a share of parameters.

TABLE IV
 FULL MINUS PARTIAL FINE-TUNING: ACCURACY DIFFERENCE IN POINTS

Network	Difference	95% interval	Holm p
DINOv2	+0.1	[-2.1, +2.3]	1.000
ViT-B/16	+2.5	[+0.5, +4.4]	0.090
ResNet-50	+1.2	[-1.1, +3.6]	0.996
DenseNet-121	+3.3	[+1.0, +5.8]	0.037
ConvNeXt-Tiny	-0.4	[-2.5, +1.7]	1.000
VGG16	+4.7	[+2.6, +7.1]	<0.001

C. Hyperparameters and evaluation protocol

Across the 12 partial and full configurations, lowering the backbone learning rate (x0.3) was worse than the default in all 12, raising it (x3) was better in 8, and the regularization recipe was better in only 3. For frozen heads, all three trials improved

on the default in every network. The best trial lay at the upper edge of the searched range in eight configurations, so the optimum may be higher. Fold 0, used for tuning, was also the easiest fold: for the four configurations whose default was retained (no selection effect), its Macro-F1 exceeded the mean of folds 1 to 4 by 0.055. Single-split results therefore overstate performance by about five Macro-F1 points. Partial and full runs reached 99 percent training accuracy by a median epoch of 9 to 20, so longer training is unlikely to help.

D. Efficiency

ResNet-50 with full fine-tuning matches DINOv2 with full fine-tuning with 23.5 million instead of 85.7 million parameters, 3.3 instead of 9.4 minutes of training per run, 6.3 instead of 21.1 ms per image and a 94 MB instead of 343 MB checkpoint. Partial fine-tuning cut training time relative to full by 27 percent for ResNet-50 and 54 percent for DINOv2.

E. Class-wise errors and image source

Recall was highest for healthy nails, clubbing and bluish nails and lowest for onycholysis (0.63 averaged over the tied configurations), half-and-half nails (0.75) and koilonychia (0.77). Of 625 errors made by the six tied configurations, 145 (23 percent) were confusions between onychomycosis and psoriasis. Accuracy was lower on BASE images than on external images, but BASE holds all the small classes, so this largely reflects class mix. Within the same class, healthy nails and clubbing showed no clear BASE penalty, whereas BASE psoriasis was recognized less often than archive6 psoriasis. Moreover, BASE psoriasis images with a visible watermark were classified correctly far more often than those without (0.98 against 0.66 per-image accuracy over the tied configurations; permutation $p = 0.002$), and this held in all 18 configurations. This is consistent with a watermark cue but also with watermarked images coming from a different photographic source.

F. Explainability results

Faithfulness and padding results are in Table V and Fig. 3. For the CNNs and DINOv2, Grad-CAM was more faithful than a random map on both curves (deletion gains 0.029 to 0.150; insertion gains 0.092 to 0.258; all intervals excluded zero) and placed little saliency on padding (ratio 0.33 to 0.45). ViT-B/16 behaved differently: its Grad-CAM deletion gain was indistinguishable from zero (-0.008), its rollout deletion gain was negative (-0.061), and its padding ratios were 0.81 (Grad-CAM) and 1.23 (rollout).

TABLE V
 FAITHFULNESS GAINS AND PADDING RATIO (GC: GRAD-CAM, AR: ATTENTION ROLLOUT)

Map	Deletion gain	Insertion gain	Padding ratio
ResNet-50 full, GC	+0.078	+0.258	0.45
ResNet-50 partial, GC	+0.029	+0.239	0.45
DenseNet-121 full, GC	+0.075	+0.249	0.33
ViT-B/16 full, GC	-0.008	+0.125	0.81
ViT-B/16 full, AR	-0.061	+0.052	1.23
ViT-B/16 frozen, GC	+0.007	+0.083	0.84
ViT-B/16 frozen, AR	-0.017	+0.035	1.19
DINOv2 full, GC	+0.150	+0.092	0.34
DINOv2 full, AR	+0.048	+0.046	0.67

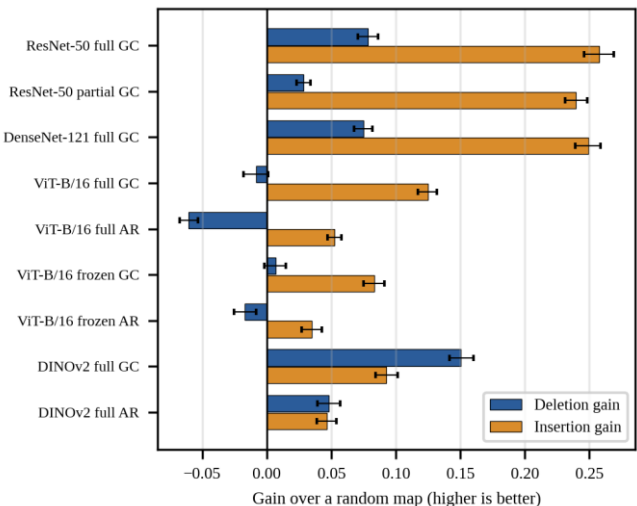


Fig. 3. Deletion and insertion gains over a random map with 95 percent group-bootstrap intervals (GC: Grad-CAM, AR: attention rollout).

Maps of different networks agreed poorly. The two CNNs agreed with each other (mean per-image Spearman correlation 0.72), but any other pair of Grad-CAM maps correlated only 0.13 to 0.25, and ViT-B/16 rollout correlated negatively with CNN Grad-CAM (-0.26). Models of similar accuracy therefore look at different regions, and the explanation method changes the picture (Fig. 4). In the sanity check, CNN Grad-CAM maps correlated 0.35 to 0.47 with random-network maps, against 0.10 to 0.13 for the transformers, so fine spatial detail of CNN maps should be read cautiously.

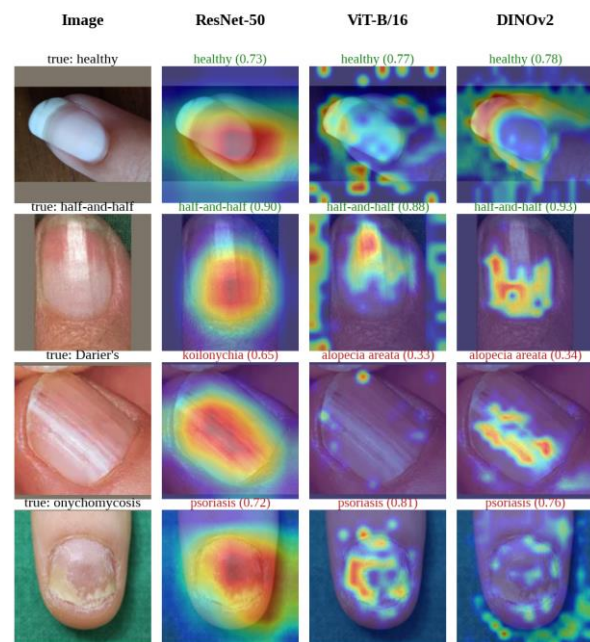


Fig. 4. Grad-CAM maps for images chosen by a fixed random rule (seed 0): two images that all four fully fine-tuned networks classified correctly (top) and two that all four misclassified (bottom). Titles give the predicted class and probability (green correct, red wrong).

Hiding the watermark lowered the predicted-class probability by 0.064 to 0.097 on average, against 0.006 to 0.045 for random boxes of the same size (Table VI; Holm p between 0.0001 and 0.025), but changed the predicted class in

only 0 to 3 of the 21 images. Grad-CAM placed 1.3 to 2.3 times as much saliency inside the watermark box as expected from its area. The models therefore use watermarks modestly, but this effect is too small to explain the large recall difference in Section V-E; other differences between watermarked and unwatermarked images, such as the photographic source, may contribute. Mean-color occlusion is itself an artifact, only 21 images were available, and the boxes were proposed by an AI assistant and checked visually.

TABLE VI
WATERMARK OCCLUSION ON 21 IMAGES: PROBABILITY DROP

Configuration	Watermark hidden	Random box	Holm p	Class changed
ResNet-50 full	0.094	0.022	0.009	2
ResNet-50 partial	0.093	0.045	0.025	0
DenseNet-121 full	0.084	0.023	0.019	3
ViT-B/16 full	0.064	0.009	0.012	0
ViT-B/16 frozen	0.097	0.009	0.011	3
DINOv2 full	0.066	0.006	<0.001	1

G. Limitations

The dataset is small, imbalanced and web-sourced, labels are not pathology-confirmed, and groups are not patients. Hyperparameters were chosen on fold 0 without a validation set, and the tuned settings were not compared with the defaults on folds 1 to 4. One random seed was used, the search was small, partial fine-tuning unfreezes different shares of each network, and the pretraining data differ (ImageNet-1k against 142 million images). Images were at most 160 pixels and were enlarged to 224. No source-held-out experiment was run, no expert nail-region annotation was available, DINOv2 with partial fine-tuning, ConvNeXt-Tiny and VGG16 were not explained, and external validation on independent data is still needed.

VI. CONCLUSION

Six pretrained networks were compared under frozen, partial and full fine-tuning for 13-class nail disease classification with equal tuning budgets and evaluation on folds not used for tuning. Fine-tuning is essential: every network gained 11.6 to 18.5 accuracy points over frozen features. Partial fine-tuning of the last stage was enough for DINOv2, ConvNeXt-Tiny and ResNet-50 at 27 to 54 percent lower training cost, while full fine-tuning was significantly better for VGG16 and DenseNet-121. No single network is best: DINOv2 with full fine-tuning reached Macro-F1 0.857, but five configurations were statistically tied, and ResNet-50 is the most economical of them. The backbone learning rate was the most influential hyperparameter, and the tuning fold overstated performance by about five Macro-F1 points, so models should be compared at tuned settings on several grouped folds. The explainability analysis showed that CNN and DINOv2 maps are more faithful than random maps and avoid padding, that different networks highlight different regions, and that all explained configurations use watermarks modestly; it does not show that the models reason in a clinically valid way. Future work should add expert nail-region annotations, source-held-out and multi-seed experiments, and external validation. The models are screening aids and not diagnostic tools.

ACKNOWLEDGMENT

The authors thank Prof. G Kaushik Hariharan, School of Computer Science and Engineering (SCOPE), Vellore Institute of Technology, for his guidance and supervision of this work. An AI assistant (Claude, Anthropic) was used to propose the watermark bounding boxes, which the authors checked, and to assist with code, analysis and drafting; the authors verified the results and take responsibility for the content. The images were obtained from public online sources and are used for research and education only.

REFERENCES

- [1] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural networks?," in *Advances in Neural Information Processing Systems*, vol. 27, 2014, pp. 3320-3328.
- [2] M. Raghu, C. Zhang, J. Kleinberg, and S. Bengio, "Transfusion: Understanding transfer learning for medical imaging," in *Advances in Neural Information Processing Systems*, vol. 32, 2019.
- [3] D. Y. Coşar Soğukkuyu and O. Ata, "Classification of melanonychia, Beau's lines, and nail clubbing based on nail images and transfer learning techniques," *PeerJ Computer Science*, vol. 9, Art. no. e1533, 2023, doi: 10.7717/peerj-cs.1533.
- [4] G. Shandilya, S. Gupta, et al., "Autonomous detection of nail disorders using a hybrid capsule CNN: a novel deep learning approach for early diagnosis," *BMC Medical Informatics and Decision Making*, vol. 24, Art. no. 414, 2024, doi: 10.1186/s12911-024-02840-5.
- [5] S. Kornblith, J. Shlens, and Q. V. Le, "Do better ImageNet models transfer better?," in *Proc. IEEE/CVF Conf. Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 2661-2671.
- [6] S. Doerrich, F. Di Salvo, J. Brockmann, and C. Ledig, "Rethinking model prototyping through the MedMNIST+ dataset collection," *Scientific Reports*, vol. 15, Art. no. 7669, 2025, doi: 10.1038/s41598-025-92156-9.
- [7] R. Geirhos et al., "Shortcut learning in deep neural networks," *Nature Machine Intelligence*, vol. 2, pp. 665-673, 2020, doi: 10.1038/s42256-020-00257-z.
- [8] A. J. DeGrave, J. D. Janizek, and S.-I. Lee, "AI for radiographic COVID-19 detection selects shortcuts over signal," *Nature Machine Intelligence*, vol. 3, pp. 610-619, 2021, doi: 10.1038/s42256-021-00338-7.
- [9] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770-778.
- [10] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. Int. Conf. Learning Representations (ICLR)*, 2015, arXiv:1409.1556.
- [11] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 2261-2269.
- [12] Z. Liu, H. Mao, C.-Y. Wu, C. Feichtenhofer, T. Darrell, and S. Xie, "A ConvNet for the 2020s," in *Proc. IEEE/CVF Conf. Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 11976-11986.
- [13] A. Dosovitskiy et al., "An image is worth 16x16 words: Transformers for image recognition at scale," in *Proc. Int. Conf. Learning Representations (ICLR)*, 2021, arXiv:2010.11929.
- [14] A. Steiner, A. Kolesnikov, X. Zhai, R. Wightman, J. Uszkoreit, and L. Beyer, "How to train your ViT? Data, augmentation, and regularization in vision transformers," *Transactions on Machine Learning Research*, 2022, arXiv:2106.10270.
- [15] M. Oquab et al., "DINOv2: Learning robust visual features without supervision," *Transactions on Machine Learning Research*, 2024, arXiv:2304.07193.
- [16] R. R. Selvaraju et al., "Grad-CAM: Visual explanations from deep networks via gradient-based localization," in *Proc. IEEE Int. Conf. Computer Vision (ICCV)*, 2017, pp. 618-626, doi: 10.1109/ICCV.2017.74.
- [17] S. Abnar and W. Zuidema, "Quantifying attention flow in transformers," in *Proc. 58th Annual Meeting of the Association for Computational Linguistics (ACL)*, 2020, pp. 4190-4197.

- [18] J. Adebayo, J. Gilmer, M. Muelly, I. Goodfellow, and M. Hardt, "Sanity checks for saliency maps," in *Advances in Neural Information Processing Systems*, vol. 31, 2018.
- [19] S. S. Han, G. H. Park, W. Lim, M. S. Kim, J. I. Na, I. Park, and S. E. Chang, "Deep neural networks show an equivalent and often superior performance to dermatologists in onychomycosis diagnosis: Automatic construction of onychomycosis datasets by region-based convolutional deep neural network," *PLoS ONE*, vol. 13, no. 1, Art. no. e0191493, Jan. 2018, doi: 10.1371/journal.pone.0191493.
- [20] S. S. Han, "Model Onychomycosis Training Datasets (JPG thumbnails) and Validation Datasets (JPG images)," figshare, 2017. [Dataset]. doi: 10.6084/m9.figshare.5398573.v2.
- [21] N. Gurav, "Nail Disease Detection," Kaggle. [Online]. Available: <https://www.kaggle.com/datasets/nikhilgurav21/nail-disease-detection-dataset>. [Accessed: Oct. 28, 2026].
- [22] Satyam2527, "Nail Disease Detection," Kaggle. [Online]. Available: <https://www.kaggle.com/datasets/satyam2527/nail-disease-detection>. [Accessed: Oct. 28, 2026].
- [23] R. Wightman, "PyTorch Image Models (timm)," GitHub repository, 2019, doi: 10.5281/zenodo.4414861.
- [24] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," in *Proc. Int. Conf. Learning Representations (ICLR)*, 2019, arXiv:1711.05101.
- [25] I. Loshchilov and F. Hutter, "SGDR: Stochastic gradient descent with warm restarts," in *Proc. Int. Conf. Learning Representations (ICLR)*, 2017, arXiv:1608.03983.
- [26] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the Inception architecture for computer vision," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 2818-2826.
- [27] P. Micikevicius et al., "Mixed precision training," in *Proc. Int. Conf. Learning Representations (ICLR)*, 2018, arXiv:1710.03740.
- [28] A. Paszke et al., "PyTorch: An imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems*, vol. 32, 2019.
- [29] B. Efron and R. J. Tibshirani, *An Introduction to the Bootstrap*. New York, NY, USA: Chapman & Hall, 1993.
- [30] Q. McNemar, "Note on the sampling error of the difference between correlated proportions or percentages," *Psychometrika*, vol. 12, no. 2, pp. 153-157, 1947.
- [31] S. Holm, "A simple sequentially rejective multiple test procedure," *Scandinavian Journal of Statistics*, vol. 6, no. 2, pp. 65-70, 1979.
- [32] V. Petsiuk, A. Das, and K. Saenko, "RISE: Randomized input sampling for explanation of black-box models," in *Proc. British Machine Vision Conference (BMVC)*, 2018, arXiv:1806.07421.
- [33] F. Wilcoxon, "Individual comparisons by ranking methods," *Biometrics Bulletin*, vol. 1, no. 6, pp. 80-83, 1945.