Late Feature Fusion of Lightweight Deep Learning Architectures for Skin Lesion Classification

Abhinav Neelam
Department of Computer Science
California State University
Northridge, CA, USA
abhinav.neelam.591@my.csun.edu

Abhishek Verma
Department of Computer Science
California State University
Northridge, CA, USA
abhishek.verma@csun.edu

***Abstract*—Classifying skin lesions is an important and active area of research because of the importance of enabling early detection, which can greatly improve the chances of cure. There are many challenges in skin lesion images including scales, angles, rotations, lighting variations, background noise, etc. In addition, the focus on enabling detection using lightweight architectures is not explored thoroughly. We propose a late feature fusion using three separate lightweight architectures: MobileNet-v3 Large, FasterNet-T1, and shViT-s4. In addition, to reduce image variation and improve generalization of all models, on-the-fly augmentation is employed to generate realistic skin lesion samples. The ISIC 2019 dataset is used to train and evaluate the proposed models to classify eight different categories of skin lesion. For late feature fusion, the level two classifiers tested after fusion are XGBoost, LightGBM, Logistic Regression (LR) and Support Vector Machine (SVM). Various evaluation metrics are computed to adequately assess the performance of the proposed models. Fusion using Logistic Regression and SVM improve upon the lightweight architectures. The Logistic Regression classifier achieves an F1-Score of 89.63% compared to the best individual F1-Score of 88.97% by FasterNet-T1 while increasing inference time by only 0.11 seconds. In addition, the confusion matrix and the t-SNE plots are visualized to explore and understand various weaknesses of the models that classify the ISIC 2019 dataset.**

***Index Terms*—Skin Cancer, Deep Learning, Feature Fusion, CNN, Vision Transformers, Mobile Systems, Transfer Learning**

I. Introduction

Skin cancer is a highly prevalent and an invasive disease that affects many lives. Early detection is challenging because of the lack of enough trained dermatologists and the lack of accessible vision technologies that can detect them accurately, efficiently, and at an early stage of disease progression. According to the authors in [1], early detection of skin cancer can vastly improve cure rates. Once diagnosed, the proper form of treatment can be utilized to reduce mortality rates. In addition, the availability of low cost and accessible deep learning neural networks can improve early detection with high accuracy and reliability while being accessible to all with access to a handheld device. With this purpose in mind, our research addresses the key goal of creating novel lightweight fused architectures with high classification performance.

An important issue to address for skin lesion classification is how to handle an imbalanced skin lesion dataset. Class imbalance is a serious problem in skin cancer detection as there are many more samples of normal non-cancerous skin images compared to cancerous skin images. In order to mitigate this issue, the authors in [2] have proposed to utilize focal loss instead of cross-entropy loss to assign higher weights to misclassified samples. In addition, data augmentation is used to create more samples similar to the real distribution of skin lesion images. Furthermore, it is of growing importance for deep learning models to classify multiple categories of skin lesions accurately, as early detection of the correct type of skin cancer can greatly improve cure chances. Another way to address the imbalance is to use on-the-fly data augmentation. The authors in [3] utilize on-the-fly data augmentation to mitigate class imbalance to improve the generalization of unseen images.

Typical deep learning approaches rely on a convolutional neural network (CNN) or a vision transformer (ViT) to extract the backbone features necessary for an accurate classification of skin lesion images. ViTs with their self-attention mechanisms can capture global context with high precision [4]. However, the big downside is that vision transformers need a large amount of data to train. In addition, ViTs often have to be pretrained in order to classify accurately. However, CNNs are far more efficient during training and inference. Furthermore, CNN's can exploit spatially local features that are common in natural images. With these reasons, often a hybrid approach is utilized that combines both the architectural design of the CNN and the ViT, which can make best use of images for classification. State-of-the-art architectures make clever use of their designs to create a versatile architecture that can effectively combine the strengths of CNNs and ViTs. However, feature fusion from lightweight and efficient models to improve upon the base models is not well explored in several domains. Our research proposes a late feature fusion approach.

The paper is organized as follows. Section II presents recent related work and briefly discusses common approaches. Section III describes the chosen dataset and summarizes it's characteristics. Section IV describes the proposed fusion approach as well as the architectures chosen for fusion. Section V describes the experimental setup. Section VI describes the results. Finally, Section VII concludes the research work and discusses the focus of future research.

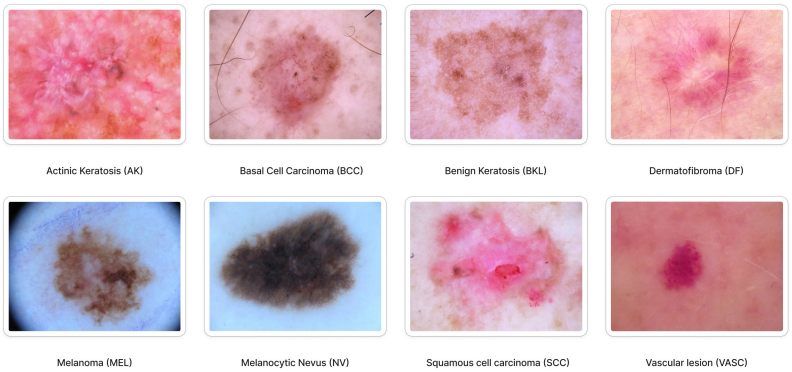

Fig. 1: Example images of the eight different categories from the ISIC 2019 dataset [5].

II. Related Work

A. Hybrid Architectures

There have been many advances in vision architectures in recent years. Often, the best architectures for segmentation and classification employ a hybrid of vision transformer networks (ViT) and Convolutional Neural Networks (CNN) [6]. This balance of architectural design is highly validated by the need for higher accuracy while also ensuring it is cost efficient for deployment. The authors in this article showed that their ConvNeXt based approach is highly reliable by utilizing the strengths of both ViTs and CNNs [6]. ViT blocks can effectively model long-range feature dependencies that would be difficult to capture with CNN blocks. CNN blocks are highly suited for local features that are closer together as these blocks are known to exploit spatial locality. The authors tested their novel architecture on the HAM10000 [7] and the ISIC 2019 datasets [5] and found that the proposed models deliver state-of-the-art results. The authors note that the combination of such diverse networks is likely to gain popularity for early detection.

Another study also employs this strategy by using the Swin Transformer architecture to classify skin lesions from the ISIC 2019 dataset [8]. The authors highlight the usage of the Swin transformer model, which utilizes the combination of both ViTs and CNNs to improve detection across vastly different image variations while also keeping it efficient enough for inference. Ultimately, their approach outperforms other methods in achieving higher multiclass balanced accuracy on the ISIC 2019 dataset.

B. Lightweight Architectures

As the field of deep learning advances and the models improve their performance, a major improvement is focusing on lightweight deep learning models, which can run efficiently in a resource constrained environment [1]. There have been several advances in this area on various skin classification and segmentation tasks. For example, authors enhanced the MobileNetV3 architecture [9] with the Convolutional Block Attention Module (CBAM) [10] and finetuned the network with Bayesian optimization [11]. CBAM can enhance the representation of feature maps through advanced attention operations while also keeping the mechanism cheap to employ. The authors utilize the ISIC 2024 SLICE-3D dataset [12], which groups images into four distinct categories. In addition, the authors used data augmentation to help reduce class imbalance and used pretrained weights from ImageNet-1K [13] for transfer learning, which significantly boosts accuracy compared to training from scratch. The incorporation of CBAM improved results compared to previous models on the ISIC 2024 SLICE-3D dataset.

Another study utilized a MobileNet architecture with transfer learning techniques to detect melanoma and benign skin lesions [14] [15]. Based on their results, the authors found that transfer learning is an effective form of training lightweight architectures for downstream tasks. Additionally, in order to deploy the model to mobile devices, the authors converted the Keras model to CoreML and ProtoBuf for testing on iOS and Android phones respectively using their prototype app (ChekSkin). The authors also utilized 4 public datasets (ISIC 2018, DermNet NZ, MED-NODE, PH2) and heavy augmentation techniques in order to mitigate overfitting. Their work on deploying the model on mobile devices shows that it is becoming increasingly viable to do inference on many medical classification and segmentation tasks in a resource constrained environment.

Another study from 2023 utilizes a lightweight ShuffleNet architecture and combines the channel wise attention mechanism (CA) and squeeze-and-excitation (SE) blocks for enhancement [16]. In addition, the authors utilize 3 separate datasets (HAM10000, ISIC 2019, and ISIC 2020) for merging while also applying data augmentation techniques to help mitigate class imbalance. The proposed approach beats the original ShuffleNet while also reducing the latency and parameter count. From these approaches, it's clear that the future of imaging tasks involves utilization of efficient architectures that can effectively operate across many environments with ease.

III. Dataset Description

A. ISIC 2019

Typically, every few years, ISIC hosts a challenge in which the goal is to classify the many complex forms of skin cancer

TABLE I: Throughput and Top-1 Accuracy Metrics of lightweight architectures on ImageNet-1K dataset.

Performance and Efficiency Metrics		
Architectures	**Throughput (images/second)**	**Top-1 Accuracy**
MobileNet-v3-Large	13,994	75.2%
FasterNet-T1	17,827	76.2%
shViT-s4	14,283	79.4%

using various forms of modality. They provide various datasets consecutively from ISIC 2016 to ISIC 2020 [5] with recent additions of ISIC 2024 [12] and MILK10K [17]. The ISIC 2019 dataset is chosen due to the complexity of classifying multiple types of melanoma and because there has been a lot of effort in improving classification results on this dataset [7] [18] [19]. Another advantage of ISIC datasets is that they are publicly provided and free to use.

The ISIC 2019 dataset provides various categories for classification. Figure 1 shows examples of eight categories. In addition, ISIC images share high inter-class similarity, which makes classification challenging even for trained dermatologists [20]. On top of this, there can be several variations of image characteristics such as lighting, angles, rotations, and background noise. A model learning to classify these categories must learn to ignore these features in order to achieve high classification performance. Moreover, one can observe the various lighting conditions in which these images were captured. Generalization to all these variations is a must in order for models to be robust.

Class imbalance is also an issue for the ISIC 2019 dataset. From the training set, Dermatofibroma (DF) has 194 images while Melanocytic Nevus (NV) has the largest count at 10,307 images. These 8 categories are spread across a total of 20,265 images for the training set. ISIC images also have widely varying range of resolution from 600 x 450 pixels to 1000 x 1000 pixels. The file size for an ISIC image can be less than 100 KB and be more than 1000 KB. ISIC images also have background pixels that have high average intensity. In addition, for the region of center that makes up the category, average intensity is much lower.

To divide the ISIC 2019 dataset for training and evaluation, the dataset is split into 80%, 10%, and 10% for training, validation and testing respectively. This makes up 20,265 images for the training set and 2,533 images each for validation and testing.

IV. Research Methodology

This section will go over each of the baseline architectures used for the late feature fusion. Each baseline model was pretrained using ImageNet-1K dataset [13] before applying the fusion. By pretraining each baseline model, downstream tasks have much better generalization performance. In order to focus on lightweight architectures with fast inference times, the models chosen have strong focus in enabling high throughput while also having low resource footprint for inference. This ensures that the baseline architectures can run efficiently, which is highly desirable in many environments. The chosen models were MobileNet-v3 [9], FasterNet-T1 [21], and shViT-s4 [22]. The throughput and top-1 accuracy metrics for each baseline architecture are listed in Table I. For each baseline model, since we save all model weights per epoch, the model from the epoch with the highest validation accuracy is chosen as the baseline.

A. MobileNet-v3 Large

The MobileNet-v3 family of architectures focuses on the reduction of latency in comparison to its predecessor [9] through usage of network architecture search (NAS) techniques and the addition of a squeeze and excitation bottleneck in its inverted residual bottleneck block. The architecture search is optimized using an accuracy and latency trade off and the NetAdapt algorithm is used to search for the number of filters in the convolutional blocks. In addition, the h-swish activation function is introduced to improve the accuracy of neural networks while also being efficient to execute on mobile devices. There is a large and small variant of MobileNet-v3. For the fusion approach, we decided to choose the large variant. In terms of performance and efficiency, the MobileNet-v3 large architecture achieves a remarkable accuracy of 75.2% from 72.0% on the ImageNet-1K dataset while lowering multiply-adds from 300 to 219 when compared to MobileNet-v2 1.0.

B. FasterNet-T1

Designing lightweight architectures is challenging, since there are many parameters to consider. Due to this, there are several architectures that have focused on reducing the parameter count or reducing the FLOPs in designing efficient architectures. However, the authors behind FasterNet show that in many cases, reducing FLOPs does not lead to a reduction in latency [21]. For their architecture, the authors redesign the depth wise convolution block to a partial convolution block which can effectively contribute towards reducing the latency without a major performance hit. The partial convolution block works by applying a convolution on a small set of input channels while propagating the rest of channels as normal. For the baseline, we chose to use the FasterNet-T1 variant, which has 7.6 million parameters and is pretrained on the ImageNet-1K dataset. FasterNet-T1 achieves a top-1 accuracy of 76.2% with a processor latency of 17.7 milliseconds (ms).

C. shViT-s4

In recent years, for vision transformers (ViT), there have been many advances in creating efficient transformer designs. One of the vision transformers that is lightweight and has high accuracy is shViT [22]. The design of the shViT architecture comes from the motivation that traditional transformers use a small patchify stem, which dramatically increases the total number of tokens that need to be processed. In addition, the secondary motivation is that the authors showed that many attention heads have high cosine similarity to each other, which creates poor redundancy for downstream processing. Armed with this knowledge, the authors for shViT utilize a large

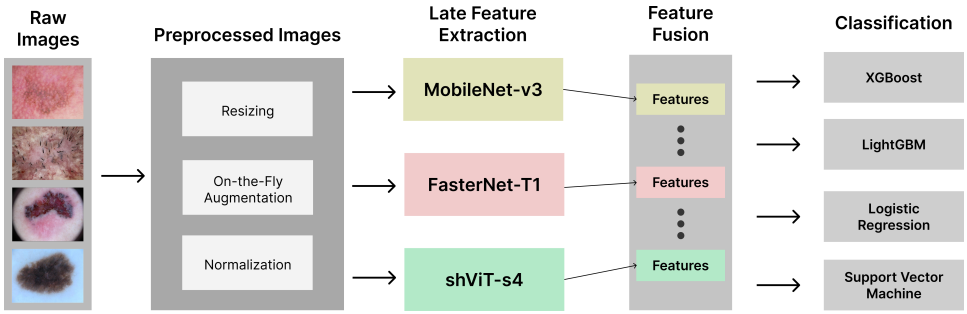

Fig. 2: Proposed system pipeline for the late feature fusion of three lightweight architectures.

16 x 16 stride patchify stem to reduce the token count and use a partial channel single-head self-attention block design in order to enhance diverse representations of self-attention heads for their architecture design. For choosing a specific variant of shViT, we decided to choose the strongest one, which is shViT-s4 with a total parameter count of 16.6 million. The model achieves an accuracy of 79.4% on the ImageNet-1K dataset with a throughput of 14,283 images per second using the Nvidia A100 GPU.

D. Proposed Late Feature Fusion Methodology

For the proposed fusion approach, we fuse the three architectures together by using late feature fusion. MobileNet-v3 large, FasterNet-T1, and the shViT-s4 architectures are chosen for feature fusion. After fine-tuning on the ISIC 2019 dataset, the features from all lightweight architectures from the last hidden layer are extracted. Each feature vector is then normalized by dividing each vector by its magnitude. This ensures that feature vectors across architectures are at similar scales before being used as input to the classifiers. The features are then concatenated after normalization. The features now represent a rich and compressed representation of the image from different architectures. Each feature sample is then reshaped to a tabular format before feeding it to the classifiers. For classification using the fused features, tree based models such as LightGBM and XGBoost were tested [23] [24]. Support vector machine (SVM) and logistic regression (LR) were also tested as classifiers and we used the scikit-learn implementation for these classifiers [25]. Figure 2 shows the high level diagram for the proposed late feature fusion pipeline.

E. Data Preprocessing

Preprocessing is a crucial part of a machine learning pipeline for many datasets. For ISIC 2019, before being fed as input to the base models, the images undergo an on-the-fly augmentation procedure. On-the-fly augmentation is different from regular augmentation in that instead of augmenting only once and storing the newly created images on the disk, the augmentation is applied during training for each batch. This procedure typically produces better results than with disk augmentation due to increased variance of the new images [26]. We noticed improved accuracy in our experiments due to on-the-fly augmentation in comparison with disk augmentation.

The exact augmentation procedure performed in order is described as: Random Affine - rotation from 45° to 180°; translation of ± 0.125 width and height ratio; scaling from 0.90 to 1.10 factor intervals. Horizontal Flip with 0.5 probability. Vertical Flip with 0.5 probability. Random color jitter - brightness factor of 0.20; contrast factor of 0.15; saturation factor of 0.10.

The random affine augmentation is created to augment images with various rotations, translations and scales. In addition, since various ISIC images can have diverse color variations, random color jitter is also applied that changes brightness, contrast, and saturation of the image [5]. With this procedure, augmented ISIC images can enhance diversity and improve generalization of the model, which helps reduce overfitting. After augmentation, images are resized to 224 x 224 square resolution since the pretrained baseline models expect this format. Then, the RGB channels of each image are normalized independently using mean=[0.485, 0.456, 0.406] and std=[0.229, 0.224, 0.225].

V. Experiment Setup

This section will go over the hardware and software setup used in training all of the baseline models and the fusion models. Several evaluation metrics are also described in detail in order to properly understand the performance of the models. Lastly, hyperparameters are provided to enable easier replication efforts for the benefit of research community.

A. Hardware and Software Setup

All of the benchmarks were run using a consistent setup. An NVIDIA GeForce RTX 4070 Ti SUPER GPU with 8,448 cores and 16 GB of dedicated memory was used to train the lightweight baseline architectures. For the CPU, 2 Intel Xeon Silver processors were used, which has 48 cores in total. The setup was configured on a Linux machine with 256 GB of RAM.

For the training scripts, PyTorch 2.8.0 was used to train and evaluate the models [27]. In addition, for each of the baseline architectures, the timm training and validation scripts [28] were modified and used. For GPU access in PyTorch, the CUDA 12.6 version was used. For visualization of results,

Matplotlib [29] and Seaborn [30] libraries were used to display the confusion matrix and the t-SNE plots. scikit-learn was primarily used to compute the evaluation metrics of the fused models [25].

B. Evaluation Metrics and Plots

The eight classes for the ISIC 2019 dataset is highly imbalanced. For this reason, we report multiple key evaluation metrics that are crucial to identifying performance given the imbalance of the dataset. To establish a well used metric for comparison, we report the accuracy. In addition, to understand the performance of the baseline architectures and the fusion classifier, we also report weighted precision and recall. To evaluate a single metric, we also compute the weighted F1-Score. These metrics are crucial to compute, since they show the true performance of the models. A confusion matrix is also used to understand what classes the model struggles to classify correctly. In addition, t-SNE visualization is utilized to give a clear picture of how well a model can separate out the classes and also to understand which classes have high similarity to each other [31].

C. Hyperparameter Setup

The three baseline models use pretrained weights from ImageNet-1K before fine-tuning on the ISIC 2019 dataset. For fine-tuning, all of the weights for all layers are unfrozen. To ensure consistent results and data splits, the Torch and NumPy [32] libraries are seeded to 42. We trained each baseline for 100 epochs. To ensure overfitting is reduced without too much performance impact, we set dropout probability to 0.2 for all of the base models. A small learning rate of 0.0001 is used to ensure smooth training and the AdamW algorithm is used with a weight decay of 0.00002 [33]. The small weight decay ensures that weights stay constrained during training, which helps regularize the models. In addition, a cosine schedule was used with a warm up of five epochs to help stabilize training of the base models [34]. The cross-entropy loss function is employed for training the base models [35]. To regularize further, label smoothing is also set to 0.1 to ensure model predictions are not overconfident [36].

For tree based models, the hyperparameters were the same for LightGBM and XGBoost. Maximum number of estimators and trees were set to 500. Maximum depth for each tree is set to 10. Learning rate for boosting rounds was set to 0.05. To reduce overfitting, weight decay is set to 1.0. To further reduce overfitting, we also set early stopping to five rounds.

For the support vector machine (SVM), we mostly use the default hyperparameters provided by scikit-learn. The RBF kernel is used to create non-linear decision boundaries for classification. To ensure overfitting isn't a problem, the regularization parameter is left as 1.0. Since there are many input features, probability estimates are disabled. Since there are 8 classes in ISIC 2019, one-vs-one approach is used for training while one-vs-rest approach is used for inference. For logistic regression, we used the L-BFGS solver with the same regularization parameter of 1.0.

TABLE II: Efficiency Performance on ISIC 2019

Model	Inference Time (s)	Throughput (images/s)	Model Size (MB)
Base Models			
MobileNet-v3 Large	33.40	75	**49.70**
FasterNet-T1	12.82	197	74.29
shViT-s4	**7.23**	**350**	190.01
Proposed Fusion Models *(Results only for the classifier stage of fusion)*			
XGBoost	0.39	6,441	0.84
LightGBM	**0.07**	**35,591**	0.48
SVM	25.47	99	40.58
LR	0.11	22,350	**0.23**

TABLE III: Classification Performance on ISIC 2019

Model	Accuracy	Precision	Recall	F1-Score
Base Models				
MobileNet-v3 Large	88.43	88.36	88.43	88.29
FasterNet-T1	**89.14**	**88.96**	**89.14**	**88.97**
shViT-s4	87.68	87.65	87.68	87.58
Proposed Fusion Models				
Fusion + XGBoost	87.96	87.88	87.95	87.79
Fusion + LightGBM	87.41	87.40	87.40	87.20
Fusion + SVM	**89.18**	**89.06**	**89.18**	**89.01**
Fusion + LR	**89.81**	**89.67**	**89.81**	**89.63**

VI. Results and Discussion

This section will go over the evaluation results for the base models and the classifiers on the ISIC 2019 dataset. We look at various evaluation metrics to compare the baselines with our proposed fusion approach. In addition, confusion matrix and t-SNE will be used to understand weaknesses of the models for classification.

A. Classification Performance

To check performance, we evaluate all baseline models and fusion models using the test set of ISIC 2019. Other than accuracy, all other metrics are computed using weighted averages. Table III shows the classification performance using the three base models as well as the performance of the four proposed fusion models. All three baseline models are competitive and achieve at least 87% testing accuracy and 87% weighted F1-Score. For each baseline model, the validation accuracies per epoch are shown in Figure 4. From the figure, all of the baseline models converge quickly, reaching at least 80% validation accuracy by the 10th epoch. From there, all baseline models continue to improve their accuracy, although at diminishing returns. From Table III, FasterNet-T1 produces the best results across three base models achieving 88.97% weighted F1-Score. Using late feature fusion, both SVM and LR based fusion models achieve higher accuracies and F1-scores compared to the baseline models. The XGBoost and LightGBM based fusion models failed to improve the classification performance compared to the baselines. However, when comparing SVM and LR, LR achieves the absolute best accuracy of 89.81% and 89.63% weighted F1-Score. For all of

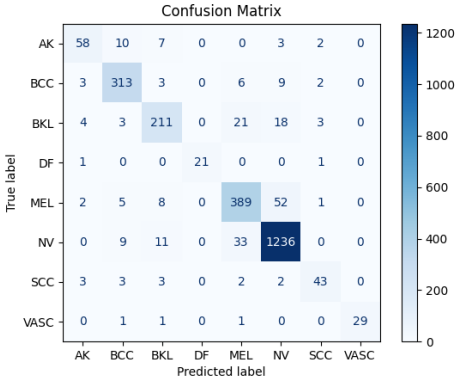

Fig. 3: Confusion Matrix using Fusion + Logistic classifier (LR) model on ISIC 2019.

the trained models, the small gap between weighted precision and recall shows that the models do not have big biases.

B. Efficiency Performance

To compare how efficient the different models and classifiers are during inference, we also provide inference time in seconds after data is preprocessed and prepared for the models, throughput in images per second for the entire validation set and size of each model in megabytes. From Table II, the baseline models spend much longer time to classify the entire validation data while the classifiers in fusion based models are much faster with the exception of SVM. LightGBM is the fastest classifier in our testing with a throughput of 35,591. LR however, still achieves a throughput of 22,350 with the inference time of 0.11 seconds. In addition, LR is the cheapest to store on disk, using only 0.23 MB. These results highlight the potential of Logistic Regression as an efficient, lightweight, and an accurate classifier to use for late feature fusion.

C. Confusion Matrix

To understand potential limitations in the fused model, the confusion matrix using the validation set's fused features + LR classifier is also shown in Figure 3. From the confusion matrix, the model tends to misclassify 52 MEL instances as NV and 33 NV instances as MEL. In addition, 21 and 18 BKL instances are often misclassified with MEL and NV respectively. This indicates that these classes share many similar features, so distinguishing them is challenging.

D. Visualization using t-SNE Plot

The t-SNE plot from Figure 5 is generated using the same setup as the Confusion Matrix in Figure 3. t-SNE visualization helps us understand which classes are most similar to each other by creating a two-dimensional representation of the fused features. From the t-SNE plot, most clusters are well separated from each other, which shows that the predictions will be much easier for the LR classifier. However, not all clusters are equal. The NV cluster being the biggest cluster in the plot shows that the NV class has high variance in comparison to other classes. This also makes sense as most non-cancerous samples are observationally diverse because of the large number of samples and the t-SNE plot reinforces this. Additionally, many MEL and NV samples are close to each other in the plot. This shows that the model confuses many melanoma images with nevus images and vice versa.

VII. Conclusion and Future Work

Our work shows that late feature fusion of lightweight architectures with the SVM and LR classifiers can achieve higher accuracy than individual models with negligible efficiency impact. Based on the F1-Score for LR, the classifier is able to achieve 89.63% weighted score and is 0.66% higher than FasterNet-T1's F1-Score. The procedure of on-the-fly augmentation also greatly improved generalization of all models. The confusion matrix shows that the fused model classifies most categories really well and the t-SNE plot reinforces this observation despite large variances that are predominant in some classes.

Future work should focus on lightweight architectures with better design schemes and smarter fusion strategies. A common limitation in fusing lightweight architectures is that many parameters across the architectures share similar information, and this could be optimized by smarter fusion approaches. Additional research should focus on efficient deployment of the models on real hardware. Since ISIC 2019 provides additional metadata for each image, research into multimodal integration is another crucial direction for accurate classification.

References

[1] Y. Wu, B. Chen, A. Zeng, D. Pan, R. Wang, and S. Zhao, "Skin cancer classification with deep learning: a systematic review," *Frontiers in Oncology*, vol. 12, p. 893972, 2022.

[2] E. Gayatri and S. Aarthy, "Reduction of overfitting on the highly imbalanced isic-2019 skin dataset using deep learning frameworks," *Journal of X-Ray Science and Technology*, vol. 32, no. 1, pp. 53–68, 2024.

[3] I. Pacal, B. Ozdemir, J. Zeynalov, H. Gasimov, and N. Pacal, "A novel cnn-vit-based deep learning model for early skin cancer diagnosis," *Biomedical Signal Processing and Control*, vol. 104, p. 107627, 2025.

[4] S. Takahashi, Y. Sakaguchi, N. Kouno, K. Takasawa, K. Ishizu, Y. Akagi, R. Aoyama, N. Teraya, A. Bolatkan, N. Shinkai *et al.*, "Comparison of vision transformers and convolutional neural networks in medical image analysis: A systematic review," *Journal of Medical Systems*, vol. 48, no. 1, p. 84, 2024.

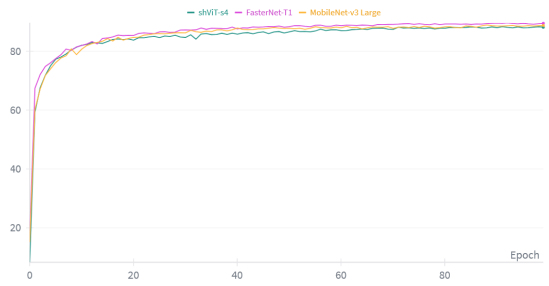

Fig. 4: Validation accuracy of three lightweight base models on ISIC 2019.

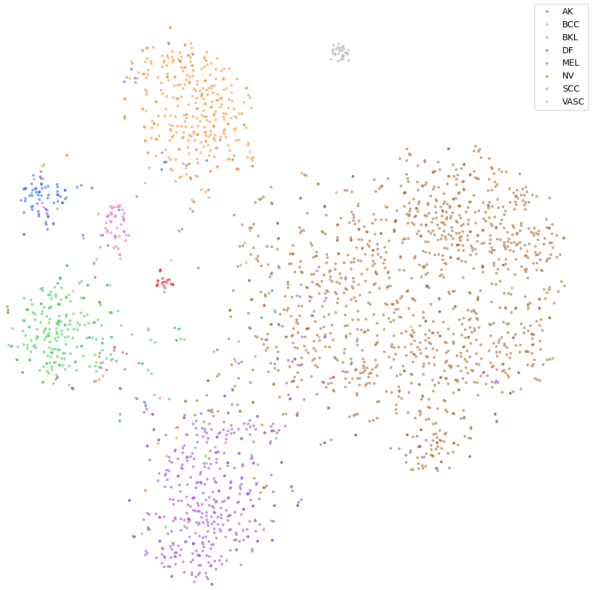

Fig. 5: t-SNE visualization using Fusion + Logistic classifier (LR) model on ISIC 2019.

[5] B. Cassidy, C. Kendrick, A. Brodzicki, J. Jaworek-Korjakowska, and M. H. Yap, "Analysis of the isic image datasets: Usage, benchmarks and recommendations," *Medical image analysis*, vol. 75, p. 102305, 2022.

[6] I. Aruk, I. Pacal, and A. N. Toprak, "A novel hybrid convnext-based approach for enhanced skin lesion classification," *Expert Systems with Applications*, p. 127721, 2025.

[7] P. Tschandl, C. Rosendahl, and H. Kittler, "The ham10000 dataset, a large collection of multi-source dermatoscopic images of common pigmented skin lesions," *Scientific data*, vol. 5, no. 1, pp. 1–9, 2018.

[8] S. Ayas, "Multiclass skin lesion classification in dermoscopic images using swin transformer model," *Neural Computing and Applications*, vol. 35, no. 9, pp. 6713–6722, 2023.

[9] A. Howard, M. Sandler, G. Chu, L.-C. Chen, B. Chen, M. Tan, W. Wang, Y. Zhu, R. Pang, V. Vasudevan *et al.*, "Searching for mobilenetv3," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 1314–1324.

[10] S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon, "Cbam: Convolutional block attention module," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 3–19.

[11] A. R. Priambodo and C. Fatichah, "Leveraging convolutional block attention module (cbam) for enhanced performance in mobilenetv3-based skin cancer classification," *Jurnal Teknik Informatika (Jutif)*, vol. 6, no. 3, pp. 1389–1404, 2025.

[12] N. R. Kurtansky, B. M. D'Alessandro, M. C. Gillis, B. Betz-Stablein, S. E. Cerminara, R. Garcia, M. A. Girundi, E. V. Goessinger, P. Gottfrois, P. Guitera *et al.*, "The slice-3d dataset: 400,000 skin lesion image crops extracted from 3d tbp for skin cancer detection," *Scientific Data*, vol. 11, no. 1, p. 884, 2024.

[13] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE conference on computer vision and pattern recognition*. Ieee, 2009, pp. 248–255.

[14] P. Ly, A. Verma, and D. Bein, "Skin cancer recognition with novel deep learning methodology on mobile platform," *International Journal of Computational Vision and Robotics*, 2024, (accepted) DOI: 10.1504/IJCVR.2024.10064682.

[15] P. Ly, D. Bein, and A. Verma, "New compact deep learning model for skin cancer recognition," in *2018 9th IEEE Annual Ubiquitous Computing, Electronics Mobile Communication Conference (UEMCON)*, 2018, pp. 255–261.

[16] A. R. Baig, Q. Abbas, R. Almakki, M. E. Ibrahim, L. AlSuwaidan, and A. E. Ahmed, "Light-dermo: A lightweight pretrained convolution neural network for the diagnosis of multiclass skin lesions," *Diagnostics*, vol. 13, no. 3, p. 385, 2023.

[17] T. Philipp, A. N. Bengü, R. Cliff, R. Veronica, T. Verche, W. Jochen, W. A. Katharina, M. Christoph, K. Nicholas, H. Allan *et al.*, "Milk10k: A hierarchical multimodal imaging learning toolkit for diagnosing pigmented and non-pigmented skin cancer and its simulators," *Journal of Investigative Dermatology*, 2025.

[18] N. C. Codella, D. Gutman, M. E. Celebi, B. Helba, M. A. Marchetti, S. W. Dusza, A. Kalloo, K. Liopyris, N. Mishra, H. Kittler *et al.*, "Skin lesion analysis toward melanoma detection: A challenge at the 2017 international symposium on biomedical imaging (isbi), hosted by the international skin imaging collaboration (isic)," in *2018 IEEE 15th international symposium on biomedical imaging (ISBI 2018)*. IEEE, 2018, pp. 168–172.

[19] C. Hernández-Pérez, M. Combalia, S. Podlipnik, N. C. Codella, V. Rotemberg, A. C. Halpern, O. Reiter, C. Carrera, A. Barreiro, B. Helba *et al.*, "Bcn20000: Dermoscopic lesions in the wild," *Scientific data*, vol. 11, no. 1, p. 641, 2024.

[20] H. Cheng, J. Lian, and W. Jiao, "Enhanced mobilenet for skin cancer image classification with fused spatial channel attention mechanism," *Scientific Reports*, vol. 14, no. 1, p. 28850, 2024.

[21] J. Chen, S.-h. Kao, H. He, W. Zhuo, S. Wen, C.-H. Lee, and S.-H. G. Chan, "Run, don't walk: chasing higher flops for faster neural networks," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2023, pp. 12 021–12 031.

[22] S. Yun and Y. Ro, "Shvit: Single-head vision transformer with memory efficient macro design," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 5756–5767.

[23] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," in *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, 2016, pp. 785–794.

[24] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T.-Y. Liu, "Lightgbm: A highly efficient gradient boosting decision tree," *Advances in neural information processing systems*, vol. 30, 2017.

[25] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[26] C.-H. Lin, C. Kaushik, E. L. Dyer, and V. Muthukumar, "The good, the bad and the ugly sides of data augmentation: An implicit spectral regularization perspective," *Journal of Machine Learning Research*, vol. 25, no. 91, pp. 1–85, 2024.

[27] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga *et al.*, "Pytorch: An imperative style, high-performance deep learning library," *Advances in neural information processing systems*, vol. 32, 2019.

[28] R. Wightman, "PyTorch Image Models." [Online]. Available: https://github.com/huggingface/pytorch-image-models

[29] J. D. Hunter, "Matplotlib: A 2d graphics environment," *Computing in Science & Engineering*, vol. 9, no. 3, pp. 90–95, 2007.

[30] M. L. Waskom, "seaborn: statistical data visualization," *Journal of Open Source Software*, vol. 6, no. 60, p. 3021, 2021. [Online]. Available: https://doi.org/10.21105/joss.03021

[31] L. v. d. Maaten and G. Hinton, "Visualizing data using t-sne," *Journal of machine learning research*, vol. 9, no. Nov, pp. 2579–2605, 2008.

[32] C. R. Harris, K. J. Millman, S. J. van der Walt, R. Gommers, and et al., "Array programming with NumPy," *Nature*, vol. 585, no. 7825, pp. 357–362, Sep. 2020.

[33] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," *arXiv preprint arXiv:1711.05101*, 2017.

[34] ——, "Sgdr: Stochastic gradient descent with warm restarts," *arXiv preprint arXiv:1608.03983*, 2016.

[35] A. Mao, M. Mohri, and Y. Zhong, "Cross-entropy loss functions: Theoretical analysis and applications," in *International conference on Machine learning*. pmlr, 2023, pp. 23 803–23 828.

[36] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2818–2826.