Skin Lesion Classification with Vision Transformers and Handcrafted Features

By Lynne Wang, Conor Huh, Mridul Jain, Vishal Saxena

MIDS 281 Computer Vision, UC Berkeley

Abstract

Skin cancer is among the most common and deadly forms of cancer worldwide, making early and accurate diagnosis critical. In this project, we investigate the HAM10000 dataset to explore the efficacy of automated classification of dermatoscopic images for accurate and timely classification of potentially deadline skin lesions. Our hybrid approach combines handcrafted simple features (such as color histograms, texture descriptors (LBP, Gabor), and edge-based metrics) and deep learning embeddings from a pretrained Vision Transformer (ViT). These features are then evaluated across a suite of classifiers (XGBoost, Logistic Regression, and K-Nearest Neighbors) using a custom cross-validation approach and bayesian hyperparameter tuning. Our best-performing model—XGBoost with oversampling optimized using Macro F1 optimization—achieved an 85.1% accuracy, 0.7407 Macro F1-score, and 0.9699 AUC-ROC on our held-out test set.

1. Introduction

HAM10000 dataset ("Human Against Machine with 10,000 training images) consists of dermatoscopic images in 600 × 450 resolution, collected from two primary sources:

- The Department of Dermatology at the Medical University of Vienna in Austria, and
- Dr. Cliff Rosendahl in Queensland, Australia.

Fig.1 (below) illustrates representative dermatoscopic images for each of the seven skin lesion categories in the HAM10000 dataset: Benign Keratosis-like Lesions, Melanocytic Nevi, Dermatofibroma, Melanoma, Vascular lesions, Basal Cell Carcinoma, and Actinic Keratoses.

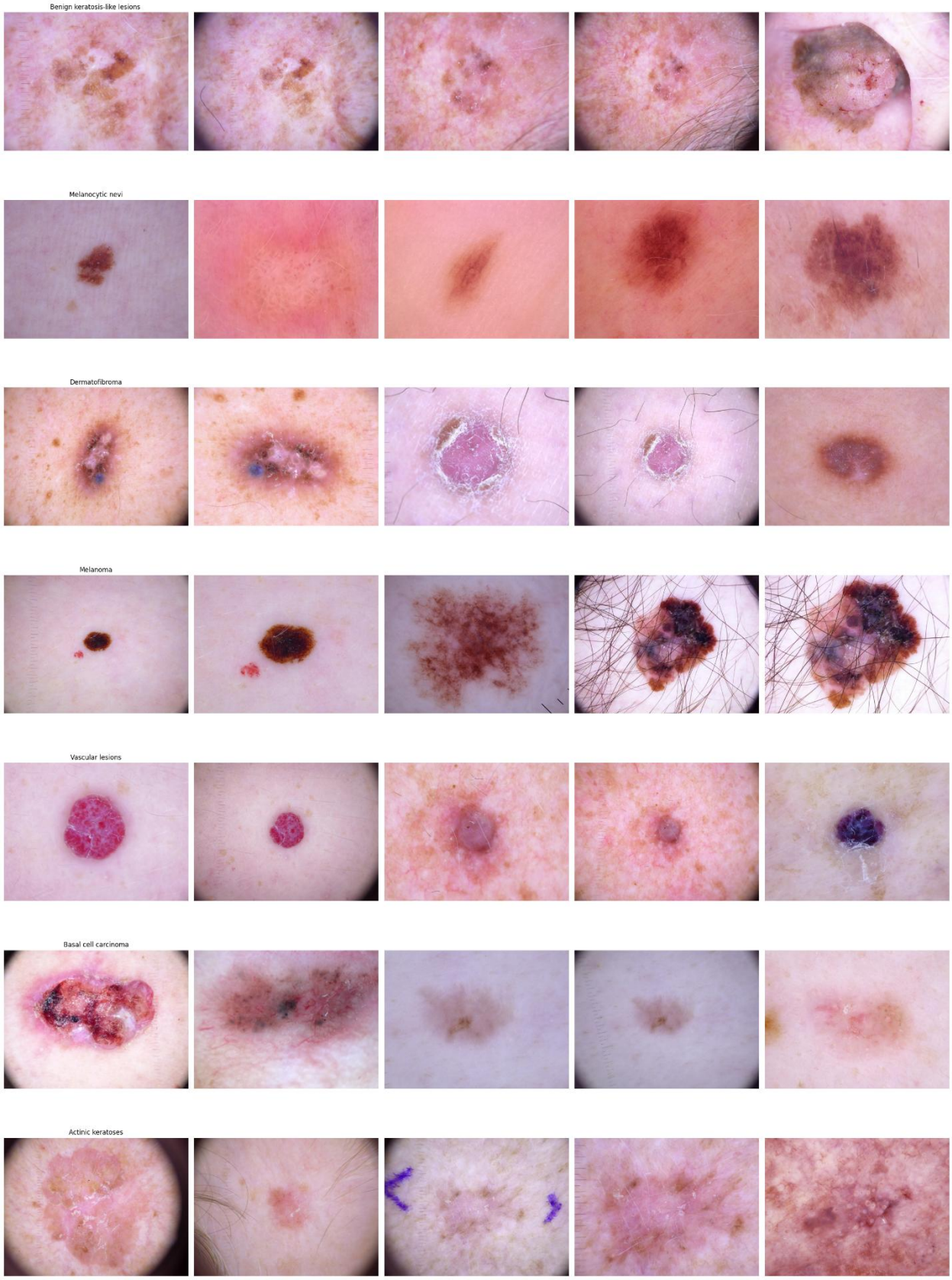

Benign keratosis-like lesions
Melanocytic nevi
Dermatofibroma
Melanoma
Vascular lesions
Basal cell carcinoma
Actinic keratoses

Fig. 1

Fig. 2 (below) shows the distribution of cell types in the dataset. As shown, the class distribution of the seven classes is highly imbalanced, with Melanocytic nevi making up the vast majority of samples (> 6k), while dermatofibroma and vascular lesions are minimally represented with a couple of hundreds of samples.

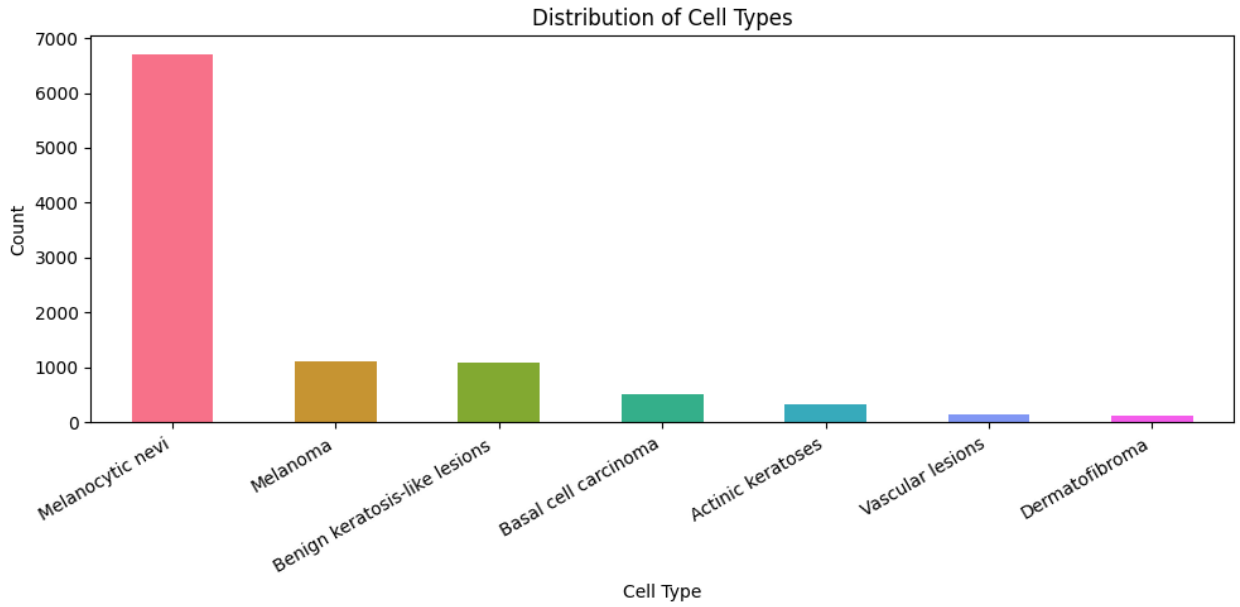


Fig. 2

The objective of this project is to design and evaluate an image classification pipeline that can accurately classify a given dermatoscopic image into one of these seven skin lesion types. This is a challenging multiclass classification task, due to both intra-class variability (e.g., different appearances of the same lesion type) and class imbalance (some categories are heavily underrepresented).

We approach this problem by combining simple features and modern deep learning features with multiple classification models. The core focus is on effective feature extraction, using both simple features (such as HSV histograms, Gabor filters, Local Binary Patterns, among others) and complex features derived from a pretrained Vision Transformer (ViT). These features are used to train a diverse set of classifiers, including Logistic Regression, Random Forest, K-Nearest Neighbors, and XGBoost, allowing us to compare both accuracy and computational efficiency across different modeling strategies.

This report details our full methodology, results, and insights from building end-to-end machine-learning models trained in skin lesion classification on this dataset.

2. Base Model

To establish a baseline, we trained a model by downsampling each dermoscopic image to just 8×8 pixels, resulting in a 64-dimensional feature vector of grayscale pixel intensities. We applied 5-fold cross-validation with upsampling via duplication of minority classes to mitigate class imbalance. In each fold, 80% of the data was used for training and 20% for validation. The trained models were then evaluated on the test set, and their performance was reported in terms of Accuracy and Macro F1-score (Table 1). This baseline model provides a reference point for assessing the effectiveness of more advanced methods, which are described in the following sections.

Fold	Accuracy	Macro F1
1	0.6912	0.3284
2	0.7068	0.3473
3	0.6973	0.3557
4	0.7185	0.3363
5	0.6973	0.3434

Table 1

3. Features

3.1 Simple Feature Selection

We implemented several classical computer vision techniques to extract simple, interpretable features from the dermatoscopic images. These features, along with their corresponding vector dimensions, are summarized in Table 2 below.

Feature Type	Description	Feature Dimension
HSV	Normalized 3D histogram of hue, saturation, and value channels	1024
HOG	Gradient-based edge orientation features in grayscale	26244
LBP	Histogram of uniform local binary patterns (rotation invariant)	10
Sobel	Descriptive stats on edge magnitude (mean, std, skew, kurtosis, percentiles, entropy)	8
Wavelet	Approximation mean + energy from detail coefficients	4
Color	Mean & std of RGB + normalized 3D RGB histogram	518
GLCM	Texture features via gray-level co-occurrence matrix	6
Gabor	Mean & std dev from multiple filtered responses (6 orientations × 4 scales)	48
Shape	Area, perimeter, compactness of the largest contour	3
Corners	Number of corners detected using Shi-Tomasi method	1
SIFT	SIFT keypoint descriptors flattened to fixed length vector (50 x128)	6400
Intensity	Statistical moments: mean, std, skewness, kurtosis of grayscale intensity	4

Table 2

3.2 Simple Feature Visualization

Fig. 3 presents representative feature visualizations for an image in the dataset. The visualizations include the original image, a grayscale conversion, Sobel edge detection, and keypoint detection using the Scale-Invariant Feature Transform (SIFT). We also include per-channel color histograms for the red, green, and blue components, along with a Local Binary Pattern (LBP) histogram to capture texture features. Additional visualizations include a grayscale intensity histogram, Histogram of Oriented Gradients (HOG) representation for edge orientation, shape contours overlaid on the original image, and corner detection results. Multiscale texture analysis is represented through discrete wavelet decomposition components: approximation (cA), horizontal detail (cH), vertical detail (cV), and diagonal detail (cD).

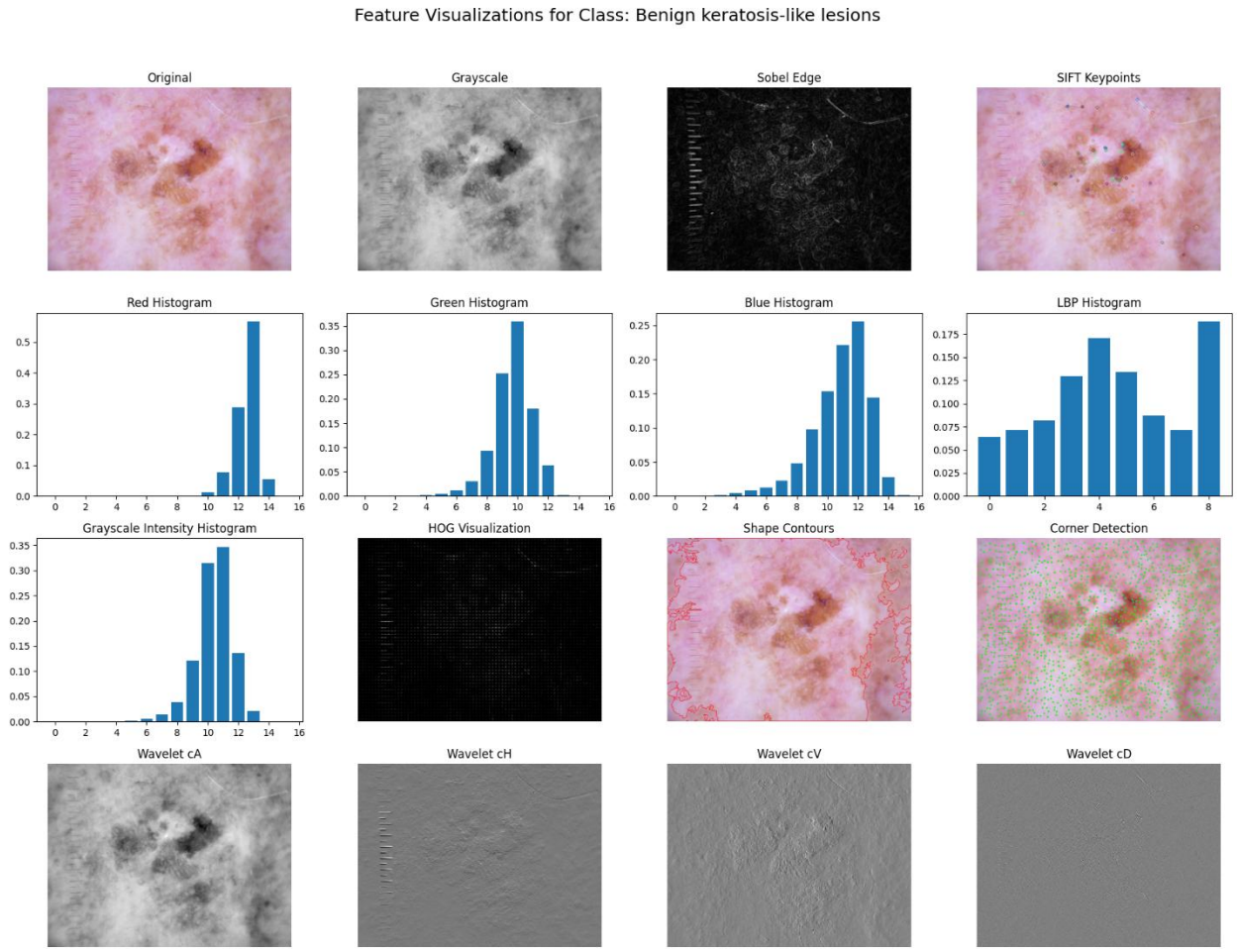


Fig. 3[1]

[1] We observed that some dermatoscopic images contain non-lesion artifacts, such as ruler markings (typically along the left edge) and body hair occlusions. To mitigate potential bias introduced by these elements, we conducted preprocessing experiments to remove them. First, we cropped all images to exclude the ruler region; however, model performance showed no measurable improvement, suggesting that either the model inherently learned to disregard the ruler or that cropping inadvertently eliminated useful lesion context.

3.3 Complex Features – Vision Transformer (ViT)

For complex features, we used a pretrained Vision Transformer (ViT) (google/vit-base-patch16-224-in21k) from the HuggingFace Transformers library to convert each image into a 768-dimensional feature vector from the token output.

3.4 Principal Component analysis (PCA)

Fig. 4 illustrates the cumulative fraction of total variance explained as a function of the number of principal components for each high-dimensional feature. The color feature reached over 99% variance explained within the first 10 components. This made it a strong candidate for aggressive dimensionality reduction. HSV features exhibited a more gradual accumulation of explained variance, with about 95% captured by the first 100 components.

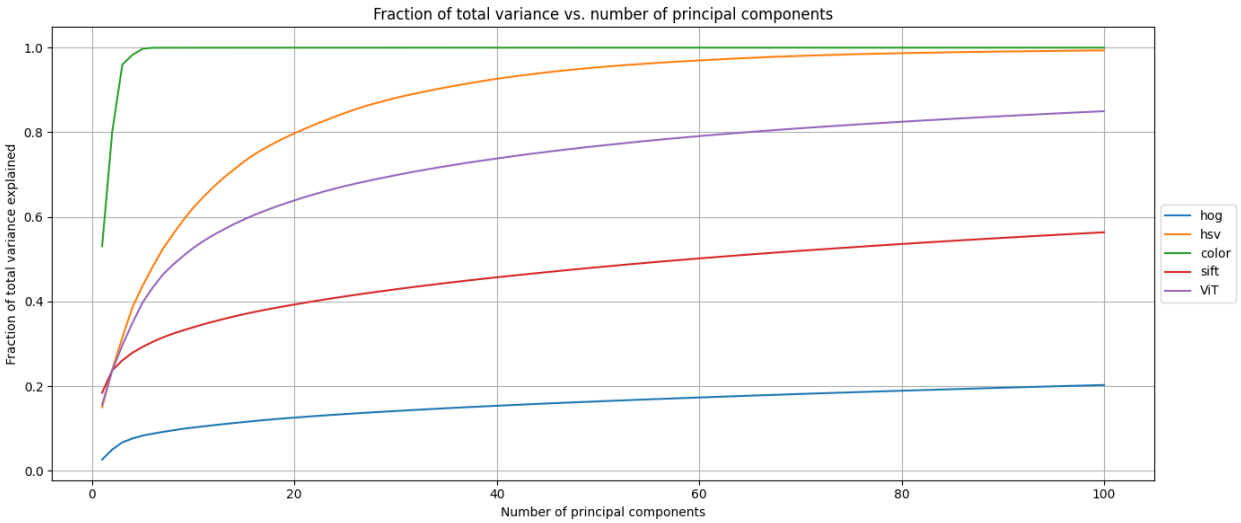


Fig. 4

Based on this analysis, we could reduce the color feature dimension to around 10 and the HSV feature dimension to around 150, as their PCA curves show that most of the variance is captured within those ranges. However, we ultimately chose not to reduce their dimensionality for the following reasons. Computational efficiency was not a major constraint in our pipeline, and preserving the full feature representation ensured maximal information retention. More importantly, we observed that reducing the dimensions of these features resulted in lower AUC-ROC scores during model evaluation, indicating a loss of discriminative power. Given these findings, we opted to retain the original high-dimensional versions of the color and HSV features to maintain classification performance.

Similarly, we applied hair removal filters to suppress dark hair overlaying some lesions, but again, no consistent performance gain was observed. Given these results, we chose to retain the original images for the final classification pipeline, as artifact removal introduced complexity without demonstrable benefit.

3.5 UMAP Visualization

To better understand the discriminative power and internal structure of the feature spaces, we applied Uniform Manifold Approximation and Projection (UMAP) to project high-dimensional feature vectors into two dimensions for qualitative visualization, as shown in FIG. 5. The ViT (Vision Transformer) features exhibited clear and distinct clustering by lesion category, indicating strong semantic separation. Simple features such as HSV histograms and LBP demonstrated only partial class separation, with substantial overlap between different lesion types.

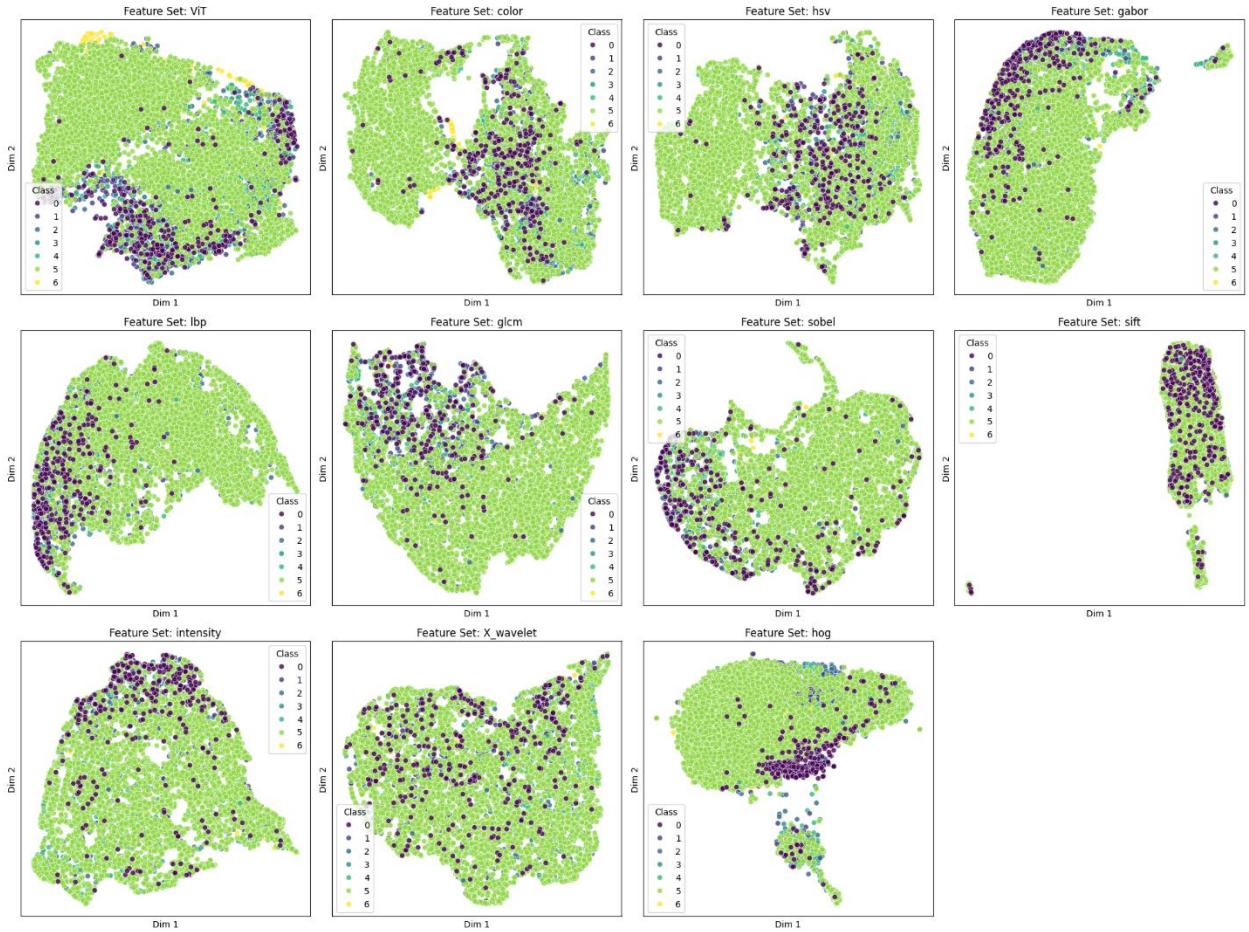


Fig. 5

3.6 Feature Screening Results

To quantitatively compare the discriminative power of each feature set, we conducted GPU-accelerated feature screening using RAPIDS cuML for each feature one at a time. To evaluate each feature independently, we trained a RandomForest model (n_estimators=100, max_depth=16, random_state=42) and then calculated the mean AUC-ROC for each model on a class-by-class basis (mean AUC-ROC one-versus-all). This process effectively converted our multiclass classification problem into a series of binary classification problems. To ensure that our feature screening matched our experiment evaluation process as closely as possible, we wrote custom oversampling code to

mirror our experiment pipeline. The purpose of this was to serve as a quick discriminative activity to reduce the number of features we considered in our experiment evaluation pipeline. Table 3 (below) presents the AUC-ROC scores for each feature set across the seven skin lesion classes, showing that ViT, HSV, and color histograms achieved the highest overall discriminative power, while traditional texture and shape-based features demonstrated limited standalone effectiveness.

	mean_auc_roc	akiec	bcc	bkl	df	mel	nv	vasc
ViT	0.918807	0.940296	0.946014	0.887043	0.913504	0.85162	0.908344	0.984825
color	0.905113	0.903213	0.915098	0.86108	0.885808	0.881939	0.899282	0.989368
hsv	0.900603	0.896002	0.910165	0.853191	0.867202	0.887739	0.897905	0.992014
gabor	0.782414	0.881289	0.810262	0.734217	0.734684	0.777433	0.814387	0.724627
lbp	0.741303	0.886965	0.730321	0.660882	0.672998	0.739452	0.801442	0.697058
glcm	0.72045	0.829348	0.74523	0.680216	0.706693	0.730609	0.797681	0.553369
sobel	0.709296	0.79191	0.722999	0.631536	0.59653	0.717012	0.76058	0.744508
sift	0.701086	0.805895	0.732431	0.676648	0.618868	0.761287	0.754935	0.557536
intensity	0.684763	0.7843	0.750177	0.642779	0.616066	0.689608	0.733427	0.576982
X_wavelet	0.631446	0.746668	0.615292	0.554721	0.553793	0.676672	0.673983	0.598994
hog	0.618517	0.747637	0.566517	0.599901	0.592902	0.557032	0.624516	0.641112
shape	0.614213	0.708403	0.657218	0.614838	0.527211	0.541862	0.677848	0.572112
corners	0.507599	0.500864	0.503842	0.501728	0.514527	0.5002	0.500283	0.531749

Table 3

4. Model Evaluation: Performance, Generalization, and Efficiency

We trained and evaluated a variety of GPU-accelerated classifiers using both simple features and complex ViT-derived features, with a focus on optimizing for both accuracy and computational efficiency. To ensure robust and fair evaluation, we incorporated stratified data splitting, per-fold oversampling or class based weighting during training, and Bayesian hyperparameter tuning within a cross-validation framework.

4.1 Training and Evaluation Strategy

Our evaluation process followed the following strategy:

- We first performed a stratified 80/20 train-test split, ensuring class distribution was preserved. The 20% test set was held out entirely for final evaluation.
- The 80% training set was then further split (80/20) into training and validation subsets, and used with 5-fold Stratified K-Fold Cross-Validation for robust hyperparameter tuning.
- We used Bayesian optimization (BayesSearchCV) with 5-fold cross-validation to identify optimal hyperparameters. For each hyperparameter candidate, five models were trained and evaluated across folds, and the average performance (Macro-F1 or AUC-ROC) was used to guide optimization.

- After selecting the best hyperparameters, a final model was retrained on the entire 80% training set, incorporating the same oversampling or class-weighting strategy.
- Threshold tuning was performed on the validation set to improve class-wise decision boundaries and enhance Macro-F1 performance on the multiclass task.

To address class imbalance during both cross-validation and final model training, we experimented with:

- Oversampling, where examples from minority classes were duplicated to increase their representation in the training folds; or
- Class balancing via loss weighting, where higher penalties were assigned to errors on minority classes.

4.1.1 Threshold Tuning (AUC-ROC Optimized Experiments)

In the case of models optimized for AUC-ROC, we observed that our AUC-ROC scores were consistently high (>95%) for our highest performing models as compared to relatively low Macro-F1 scores. To address this, we wrote code to find the best classification thresholds for each class using another, separate held out validation set. The training, hyperparameter validation, threshold validation and test set size breakdowns were as follows:

1. Training Set Size: 60%
2. Hyperparameter Validation Set Size: 12% (per fold)
3. Threshold Validation Set Size: 20%
4. Test Set Size: 20%

The thresholding strategy was as follows:

- For each class, we test a range of potential confidence thresholds (from 1% to 99%, step size 0.01)
- At each threshold, we evaluate how well the model distinguishes that single class from all others, as measured using F1-Score.
- The confidence threshold that produced the highest F1-Score was selected as the final optimal threshold for that class.

When making final predictions on the test set, our approach was as follows:

- The model identifies all classes whose classification probability scores meet or exceed their unique threshold
- If one or more classes meet the threshold, we select the highest probability class, in the event that no class meets its minimum threshold, we default to predicting the class with the highest overall probability score, ensuring that for each example in our test set, we receive a classification.

Note that for experiments directly optimized for F1-Macro as opposed to AUC-ROC, this additional threshold tuning was not conducted and maintains the original training, validation, test set size described above.

4.2 Model Performance

As discussed above, to address class imbalance, we experimented with two strategies: (1) oversampling minority classes, which increases their representation in the training data, and (2) loss weighting, which assigns higher loss penalties to errors on minority classes during training.

To identify the best model configurations, we also explored two optimization objectives: (1) Macro F1-Score, and (2) AUC-ROC score with custom per class thresholding.

Table 3 below summarizes model performance across these different combinations of class balancing strategies and optimization objectives.

F1 Optimized Models w/ Oversampling							
Model	**Train Time (s)**	**Inference Time (s)**	**Best CV F1-Macro**	**Test Set AUC-ROC**	**Test Accuracy**	**Test Macro F1-Score**	**Best Hyperparameters**
XGB*	2364.96	0.11	0.6752	0.9699	0.8509	0.7407	lr=0.29, max_depth=3, n_est=491
LS	347.99	0.03	0.6556	0.9534	0.7902	0.6712	C=2.07, penalty='l1'
RF	347.47	0.31	0.5585	0.9347	0.6983	0.6119	max_depth=10, n_bins=19, n_est=500
KNN	105.28	0.17	0.3423	0.6654	0.613	0.3496	n_neighbors=3, p=2
AUC-ROC Optimized Models w/ Oversampling							
Model	**Train Time (s)**	**Inference Time (s)**	**Best CV AUC-ROC**	**Test Set AUC-ROC**	**Test Accuracy**	**Test Macro F1-Score**	**Best Hyperparameters**
XGB	707.4	0.11	0.9566	0.9649	0.8268	0.6948	lr=0.30, max_depth=3, n_est=412
RF	86.32	0.3	0.9435	0.9514	0.7907	0.6229	max_depth=45, n_bins=18, n_est=500
LR	347.99	0.03	0.9385	0.9534	0.7987	0.6761	C=0.69, penalty='l1'
KNN	6.96	0.07	0.7335	0.7221	0.5452	0.3142	n_neighbors=30, p=2
F1 Optimized Models w/ Class Balancing							
Model	**Train Time (s)**	**Inference Time (s)**	**Best CV F1-Macro**	**Test Set AUC-ROC**	**Test Accuracy**	**Test Macro F1-Score**	**Best Hyperparameters**
XGB	2038.62	0.05	0.6244	N/A	0.8333	0.6887	lr=0.16, max_depth=5, n_est=319

LR	1627.75	0.02	0.6369	N/A	0.8047	0.6742	C=62.71, penalty='l1'
RF	302.21	0.43	0.4007	N/A	0.7585	0.4048	max_depth=33, n_bins=101, n_est=212
KNN	84.33	0.09	0.3142	N/A	0.7083	0.3421	n_neighbors=15, p=2
AUC-ROC Optimized Models w/ Class Balancing							
Model	**Train Time (s)**	**Inference Time (s)**	**Best CV AUC-ROC**	**Test Set AUC-ROC**	**Test Accuracy**	**Test Macro F1-Score**	**Best Hyperparameters**
XGB	151.56	0.05	0.9499	0.9587	0.8228	0.6638	lr=0.17, max_depth=14, n_est=187
LS***	76.91	0.02	0.9457	0.9472	0.7942	0.5898	C=105.76, penalty='l2'
RF	21.97	0.27	0.9352	0.9504	0.7952	0.5842	max_depth=23, n_bins=95, n_est=470
KNN**	1.14	0.04	0.827	0.8194	0.7063	0.3261	n_neighbors=26, p=2

*XGBoost with oversampling and Macro-F1 optimization provided the best accuracy.

**KNN with class balancing and AUC-ROC optimization provided the fastest train time and inference time.

***Logistic Regression with class balancing and AUC-ROC optimization provided a good tradeoff between accuracy and train time.

Table 3

3.3 Generalization and Robustness

Table 4 presents the detailed test set classification report for the best-performing model—XGBoost with oversampling and Macro F1-Score optimization. Compared to the baseline model, which achieved approximately 70% accuracy and a Macro F1-Score of 0.34, the XGBoost model demonstrates a substantial performance gain, reaching 85% accuracy and a Macro F1-Score of 0.74. This significant improvement highlights the effectiveness of richer feature representations. As shown in Figure 4, the confusion matrix indicates especially strong performance on the majority class (Melanocytic Nevi) and reasonable generalization to minority classes such as Vascular Lesions and Dermatofibroma. These results emphasize the importance of feature quality in achieving robust and balanced classification performance.

Class	Precision	Recall	F1-Macro Score	Support
0 (Actinic keratoses)	0.683333	0.621212	0.650794	66
1 (Basal cell carcinoma)	0.712766	0.650485	0.680203	103

2 (Benign keratosis-like lesions)	0.684932	0.688073	0.686499	218
3 (Dermatofibroma)	0.866667	0.565217	0.684211	23
4 (Melanoma)	0.711538	0.666667	0.688372	222
5 (Melanocytic nevi)	0.913931	0.940691	0.927118	1332
6 (Vascular lesions)	0.92	0.821429	0.867925	28
accuracy	0.850904	0.850904	0.850904	
macro avg	0.784738	0.707682	0.740732	1992
weighted avg	0.847812	0.850904	0.848619	1992

Table 4

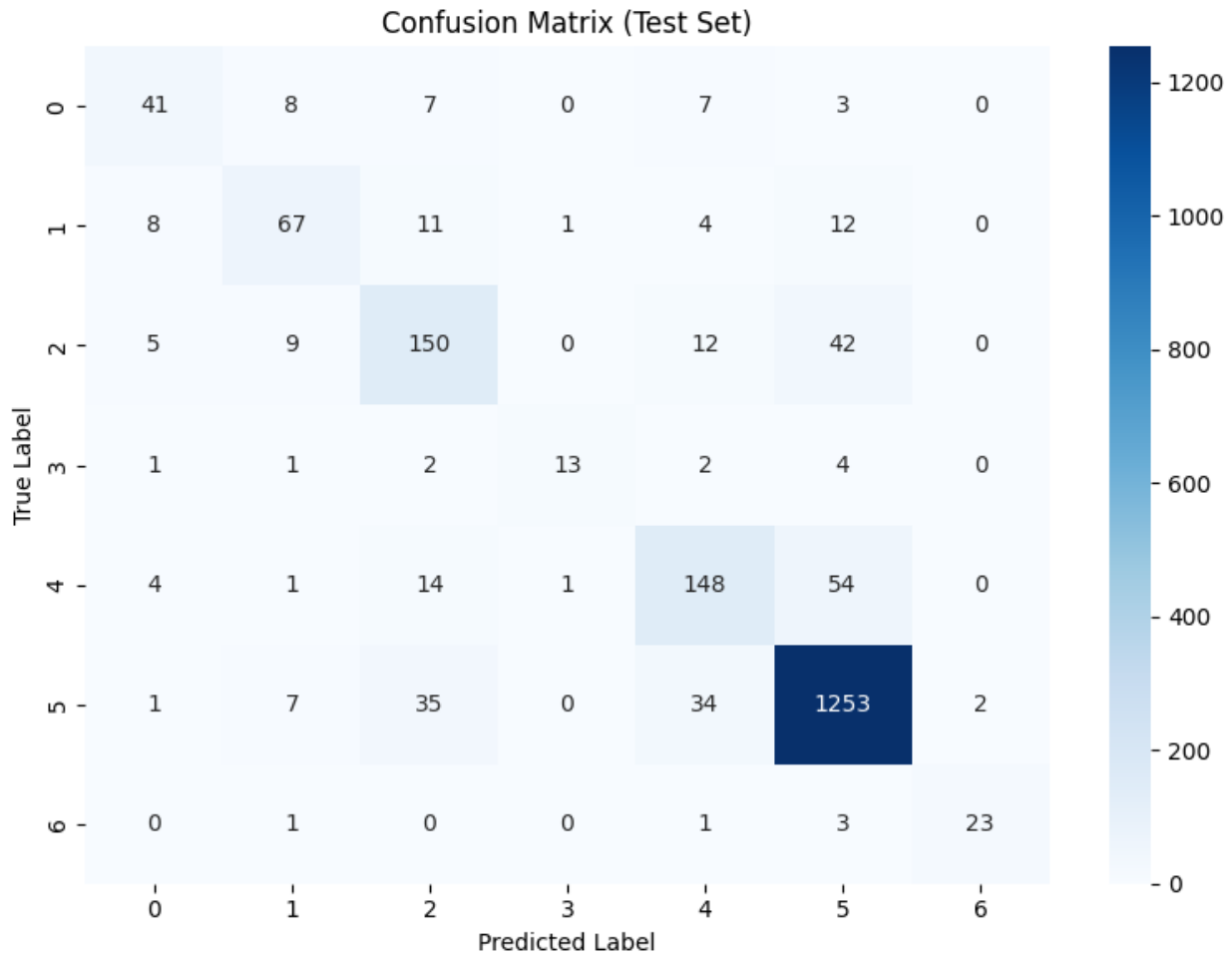


Fig. 6

As shown above, our most accurate model achieved consistent performance on unseen test data, with Macro-F1 reaching 0.7407 and AUC-ROC up to 0.97.

However, classification of underrepresented classes like vascular lesions and dermatofibroma remained challenging, even with oversampling. These findings suggest the need for additional

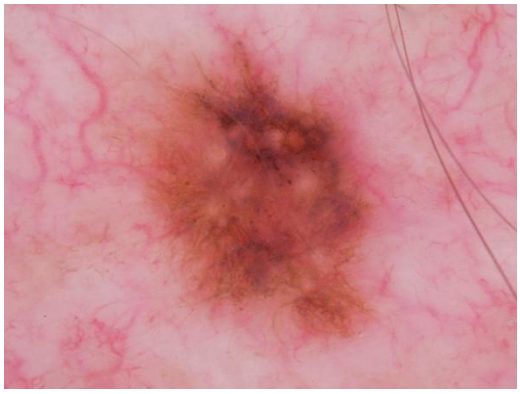
strategies, such as augmentation of training samples or contrastive fine-tuning of the embedding model.

3.4 Case Analysis: Misclassification Between Class 4 and Class 5

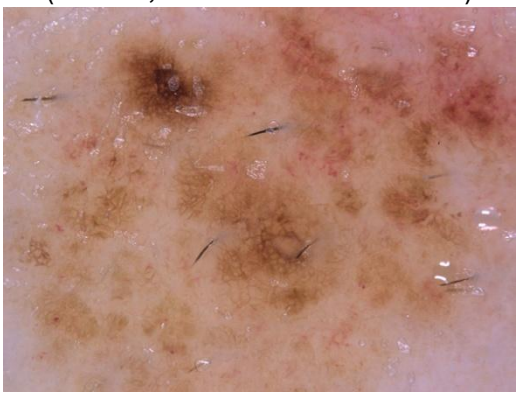
To gain further insight into the model's behavior, we examined four misclassified images (Fig. 7) where the confusion occurred between Class 4 and Class 5. Specifically:

- Top row: Images from Class 4 (true label) were misclassified as Class 5.
- Bottom row: Images from Class 5 (true label) were misclassified as Class 4.

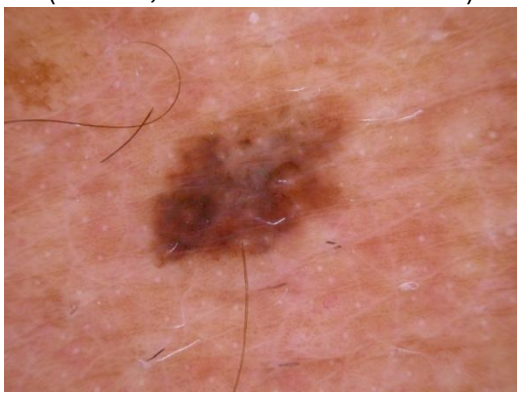
ISIC_0029272
(Class 4, misclassified as Class 5)



ISIC_0026930
(Class 4, misclassified as Class 5)



ISIC_0030771
(Class 5, misclassified as Class 4)



ISIC_0033266
(Class 5, misclassified as Class 4)

Fig. 7

Notably, the four images exhibit substantial visual similarity, making them difficult to distinguish—even for non-expert humans. Improving the model's ability to differentiate such cases may benefit from input by domain experts, such as dermatologists. Expert guidance could inform the development of more targeted edge detection and boundary-aware features, such as combining or reweighting HOG, Sobel, and corner detectors. Additionally, incorporating preprocessed images may help the model focus on clinically relevant regions and reduce distracting visual noise, better aligning with what the classifier is actually "seeing."

4. Conclusion

We found that more complex models (e.g., XGBoost) required significantly longer training times but yielded the highest accuracy and generalization. K-Nearest Neighbors (KNN) was the fastest in both training and inference, but consistently underperformed in accuracy—particularly on underrepresented lesion types. In contrast, Logistic Regression offered a trade-off, achieving strong efficiency and reasonable predictive performance.

There are several promising directions for future work:

- **Feature Refinement and Expansion:** revisiting shape, boundary, color, and edge-based features—possibly with expert guidance—to better capture lesion structure and improve class separability.
- **Preprocessing Enhancements:** Using lesion localization and contrast enhancement may help the model focus more directly on relevant lesion areas while reducing background noise.
- **Fine-Tuning Embeddings with Contrastive Learning:** using contrastive learning, to better capture lesion-specific representations and improve separability across classes.
- **Model Ensembling**, which may enhance robustness and per-class recall, especially for rare lesion categories.
- **Real-World Validation**, to evaluate model performance on out-of-distribution data and assess generalizability beyond the HAM10000 dataset.

Contribution Statement

All team members actively contributed to every phase of the project. This includes collaborative involvement in data preprocessing, simple feature extraction (e.g., HSV, LBP, Sobel, Gabor, HOG, among others), and complex feature extraction using the Vision Transformer (ViT). Each member also participated in training and evaluating machine learning models, conducting experiments, and drafting and revising the final report. The project reflects equal and shared responsibility across all core tasks.