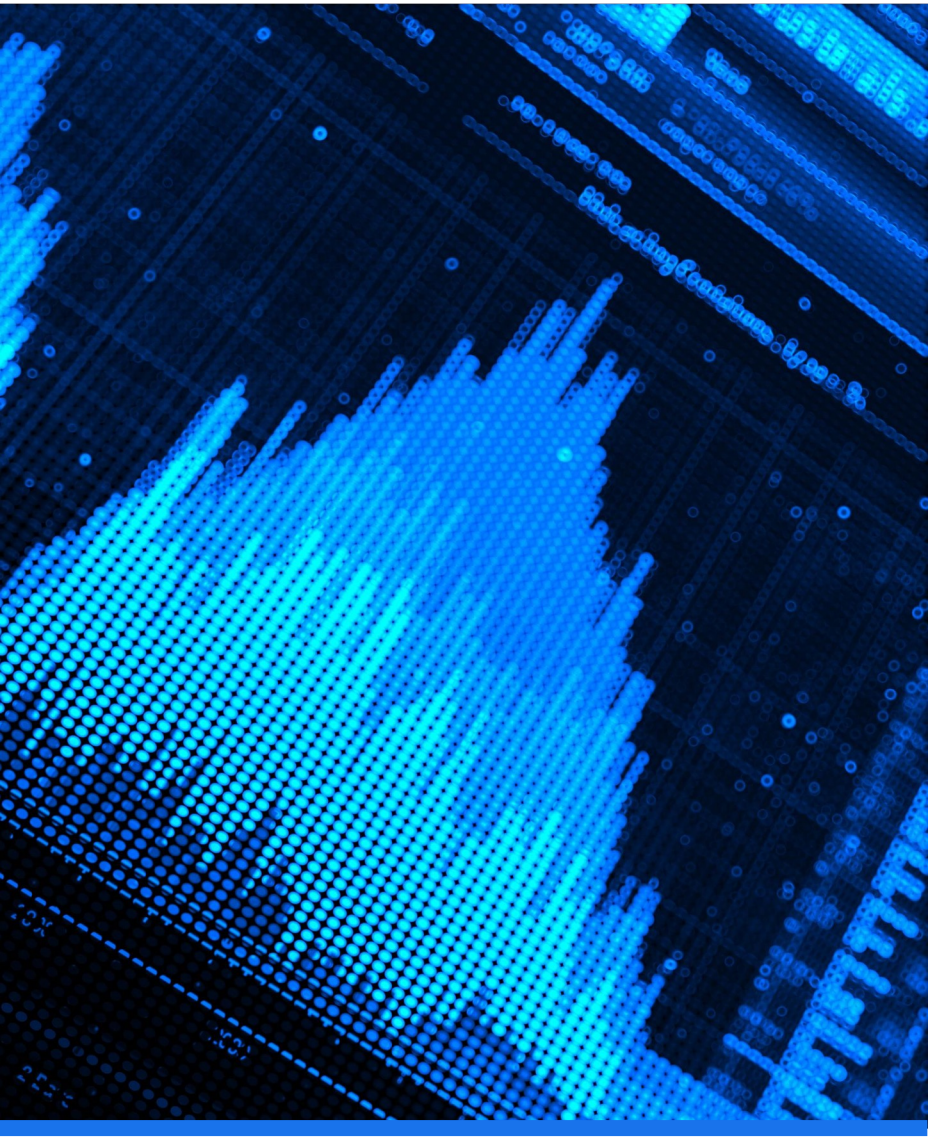
Classification Model for Lesion Images on the HAM10000 Dataset

CONOR HUH, MRIDUL JAIN, VISHAL SAXENA, LYNNE WANG

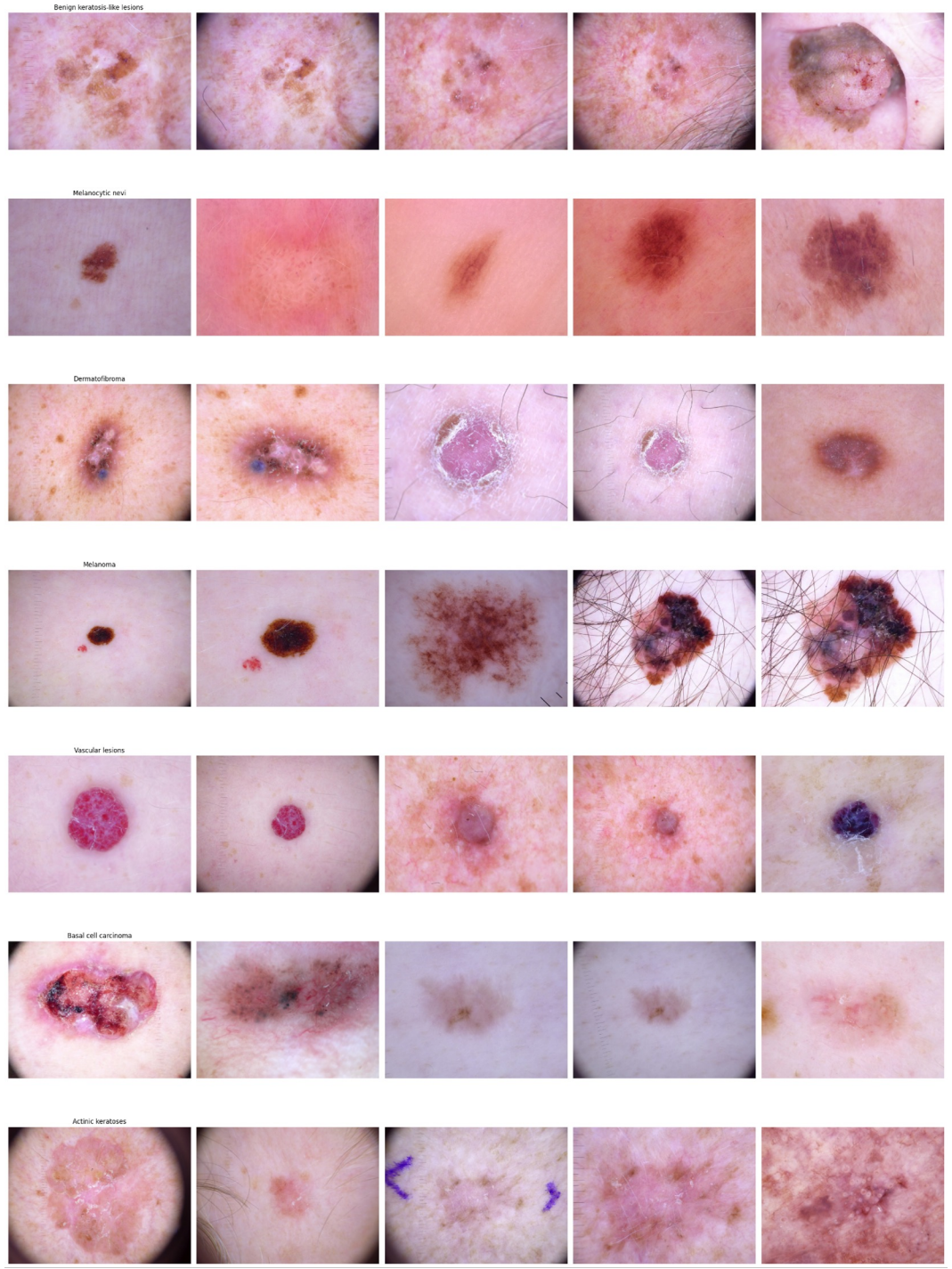
MIDS 281 COMPUTER VISION, UC BERKELEY



Presentation Roadmap

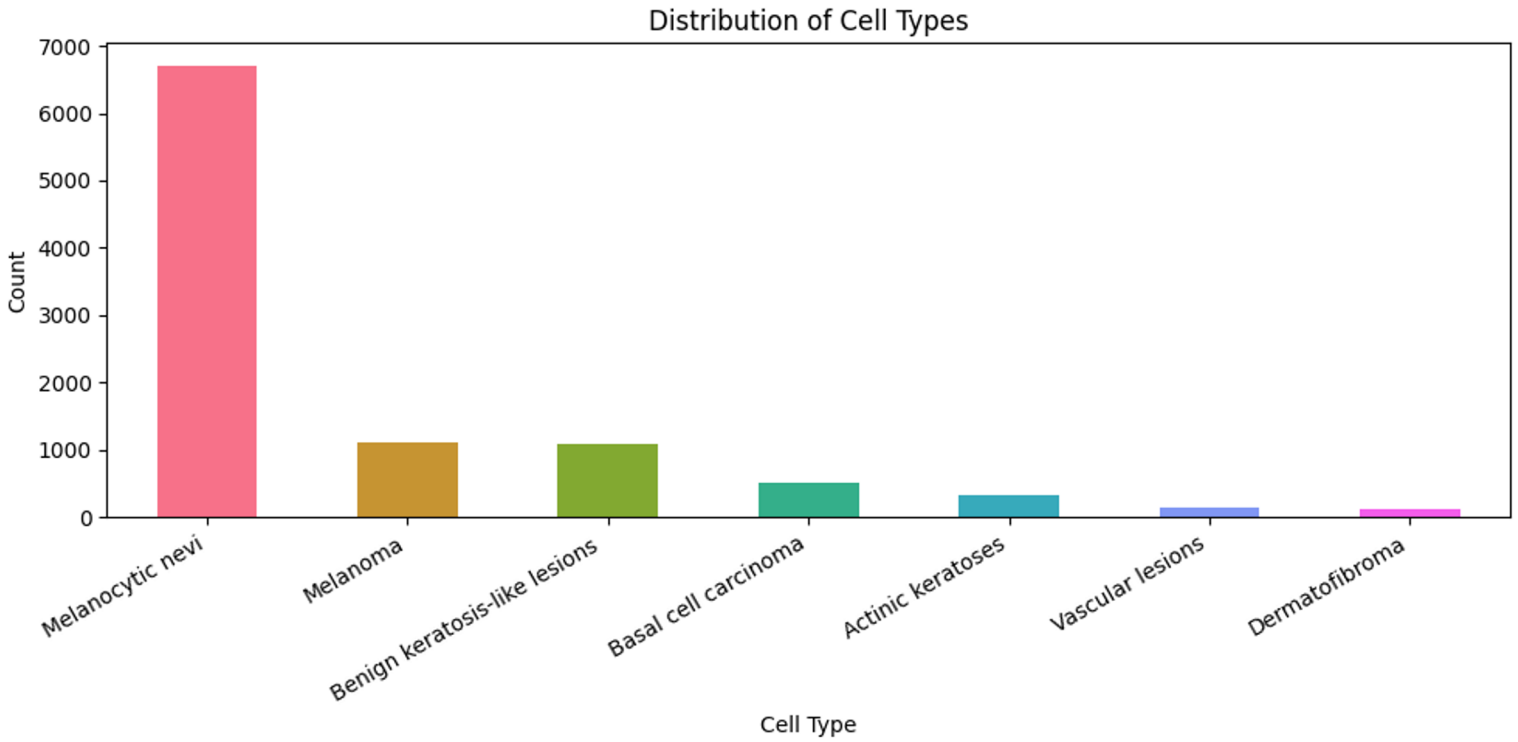
- Addressing Class Imbalance in the HAM Dataset: Upsampling Techniques
- Feature Extraction: Simple and Complex Approaches for Lesion Classification
- Feature Selection Rationale and Interpretation
- Classification Methods and Performance Analysis
- Ensuring Generalizability: Data Splitting, Hyperparameter Search, and Evaluation
- Balancing Efficiency and Accuracy: Comparative Analysis of Feature-Classifier Combinations

Overview of the HAM Lesion Dataset



Addressing Class Imbalance in the HAM Dataset:

1. Oversampling
2. Class Weighting



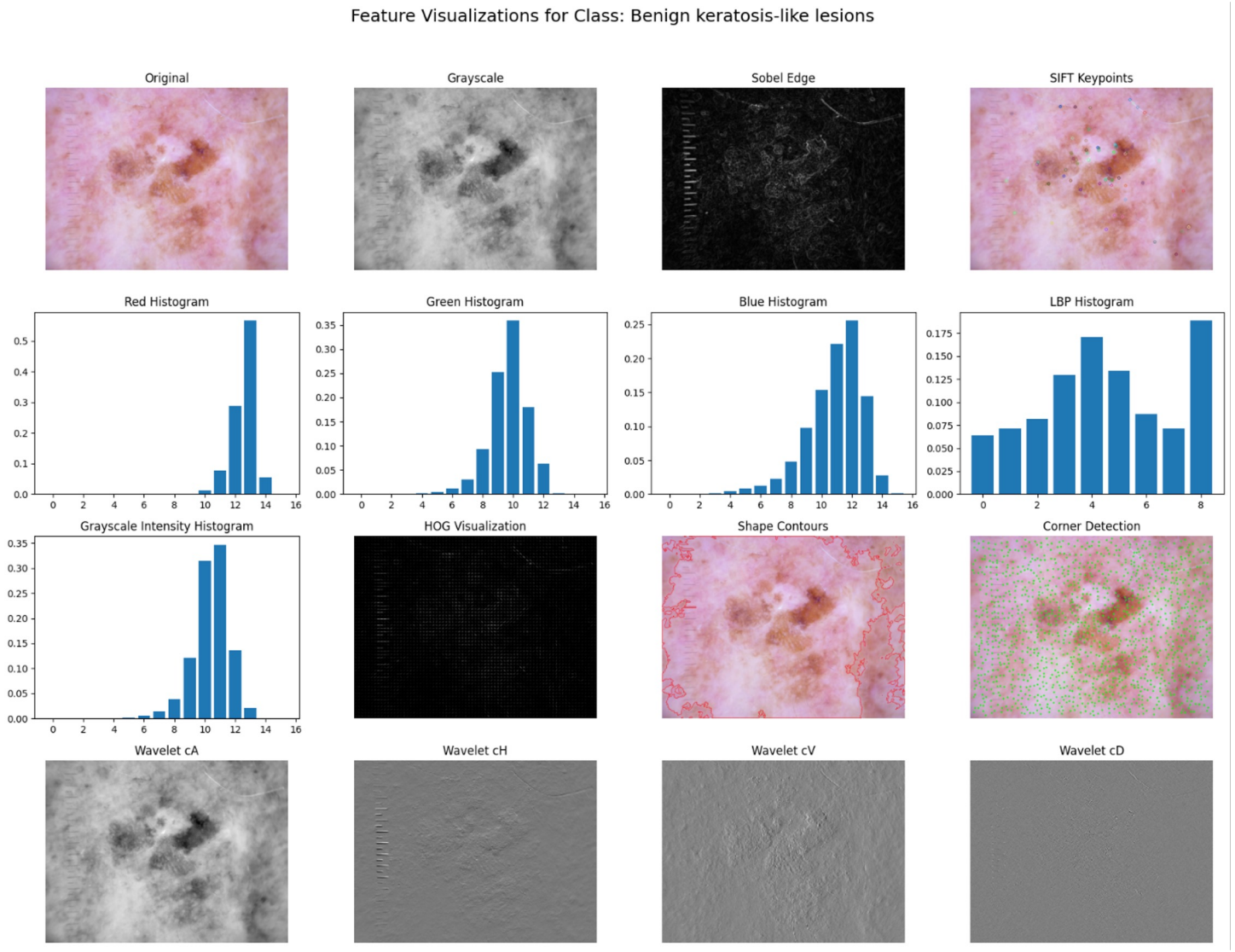
Baseline performance

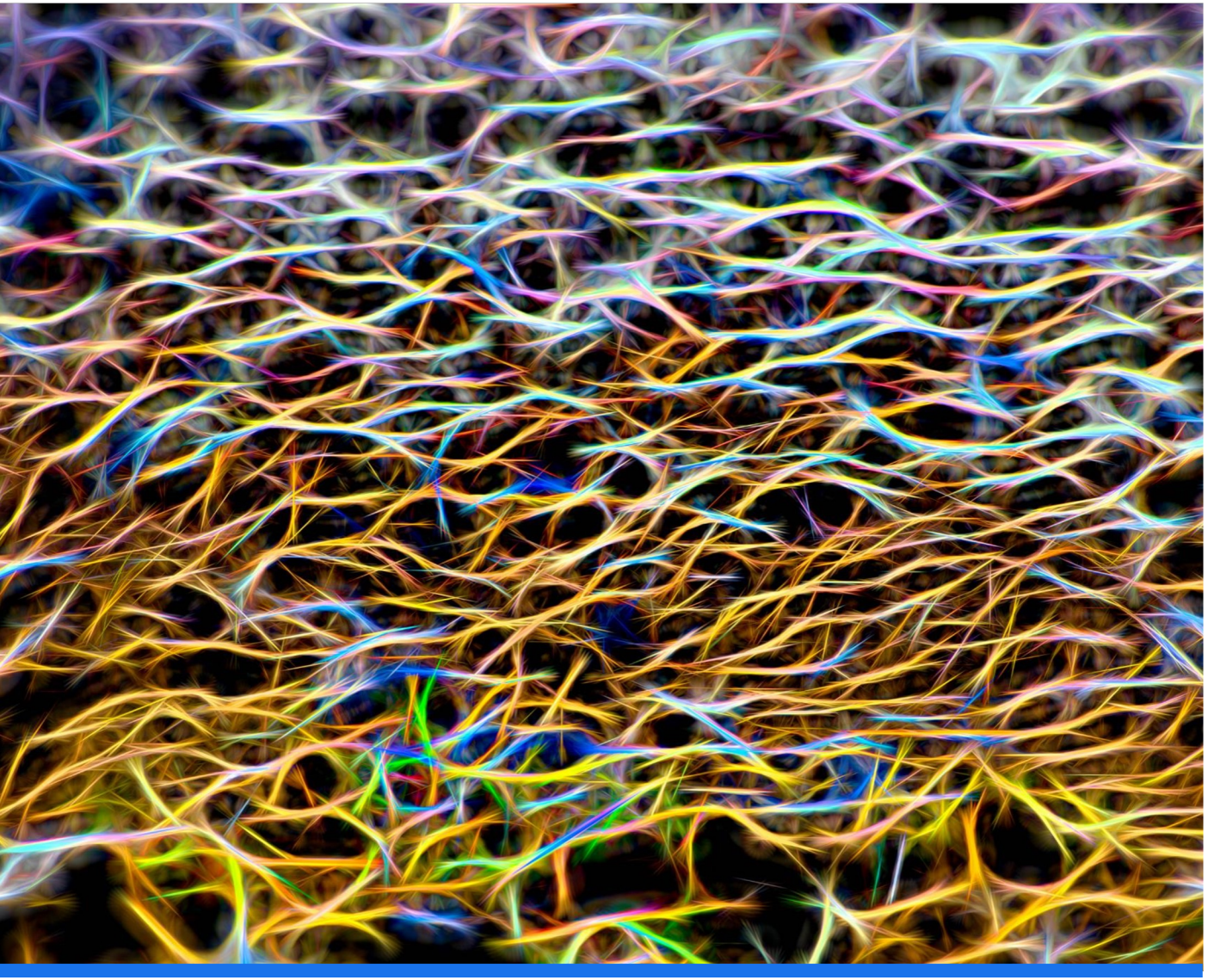
- using a majority class classifier is 67%
- Using LR for 8 x 8 downsampled pixel values is 69%

Simple Feature Extraction

Feature Type	Description	Feature Dimension
HSV	Normalized 3D histogram of hue, saturation, and value channels	1024
HOG	Gradient-based edge orientation features in grayscale	8100
LBP	Histogram of uniform local binary patterns (rotation invariant)	59
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
Corner	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

Visualization of Simple Features

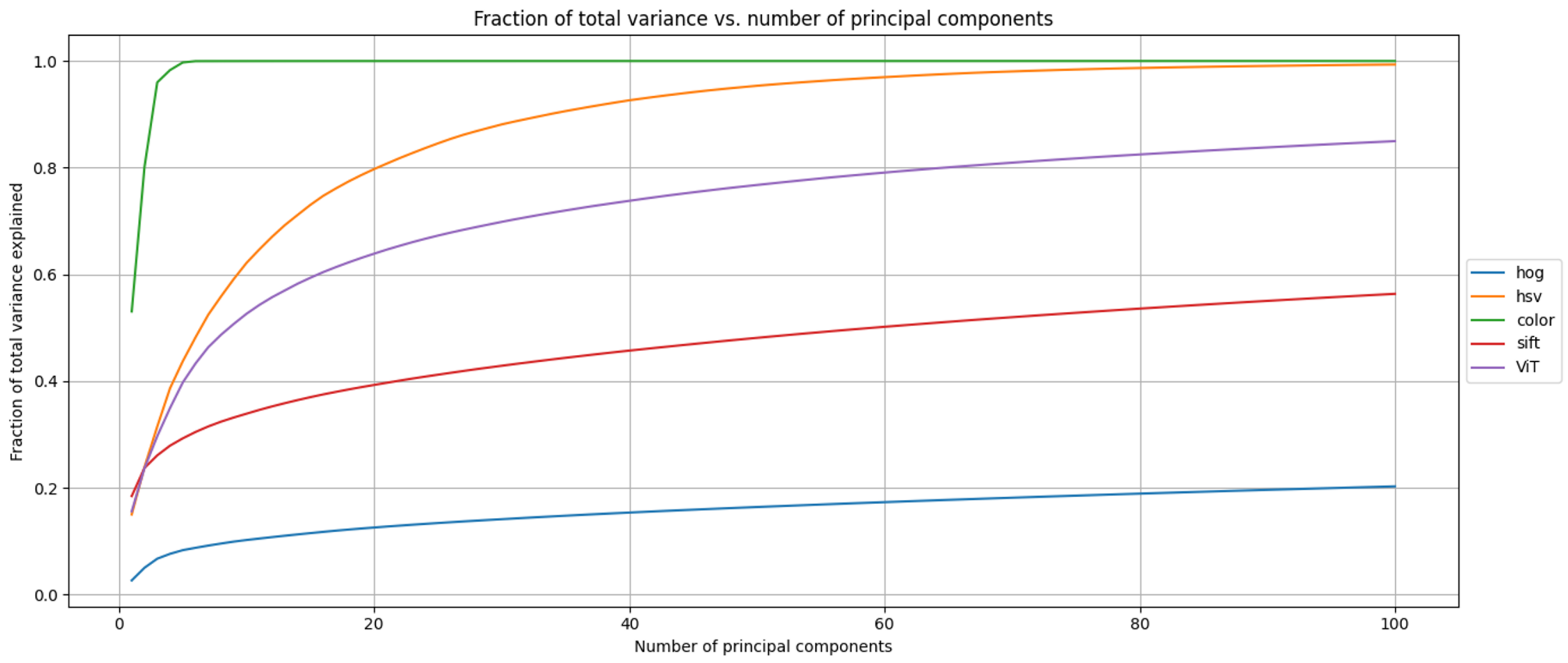




Complex Feature Extraction:

1. **Vit**
2. **Clip**
3. **Resnet**

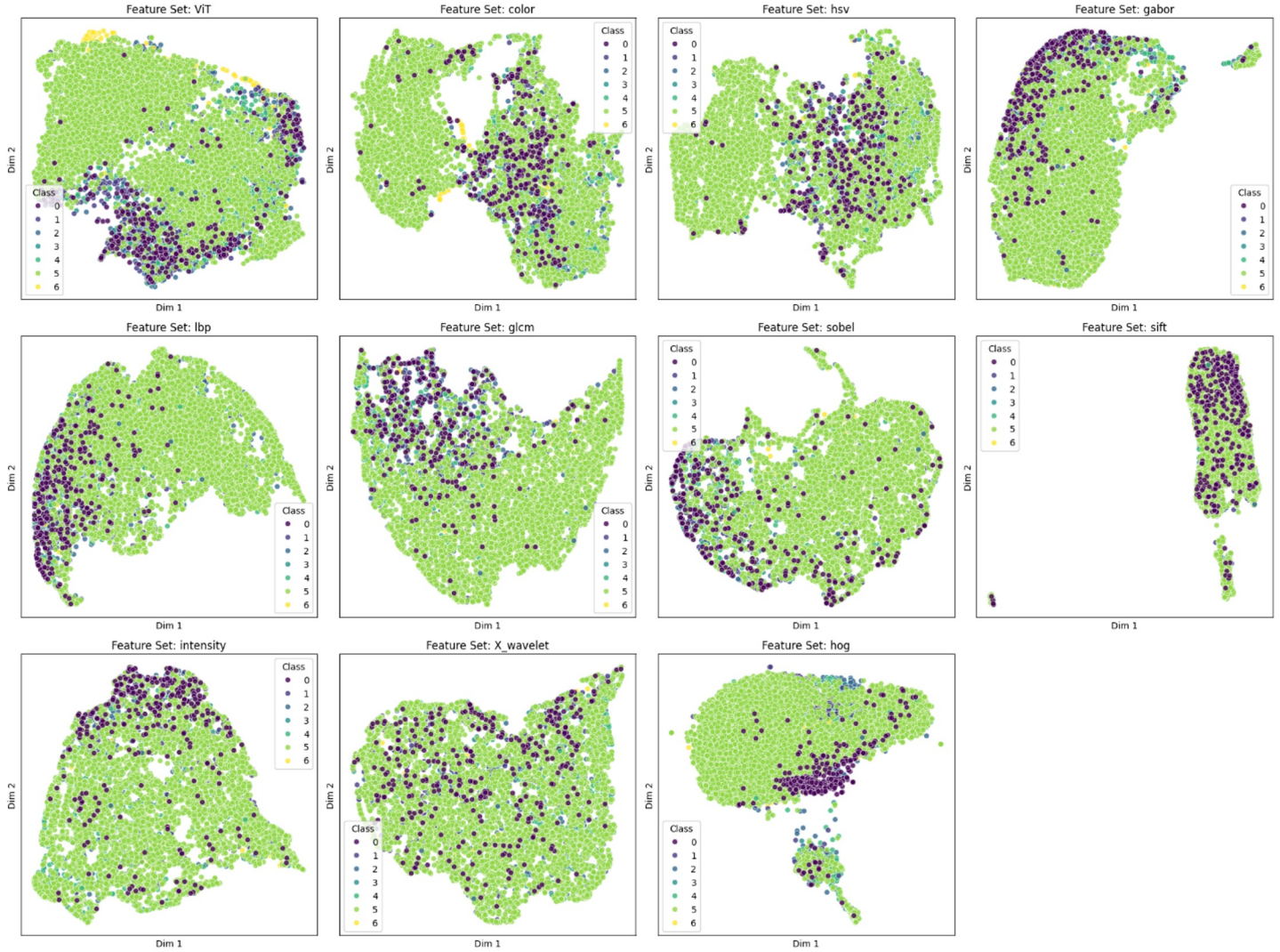
PCA of High Dimensional Features



AUC-ROC of Each Feature

	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
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

UMAP 2D Visualization of Features



Training and Evaluation Strategies

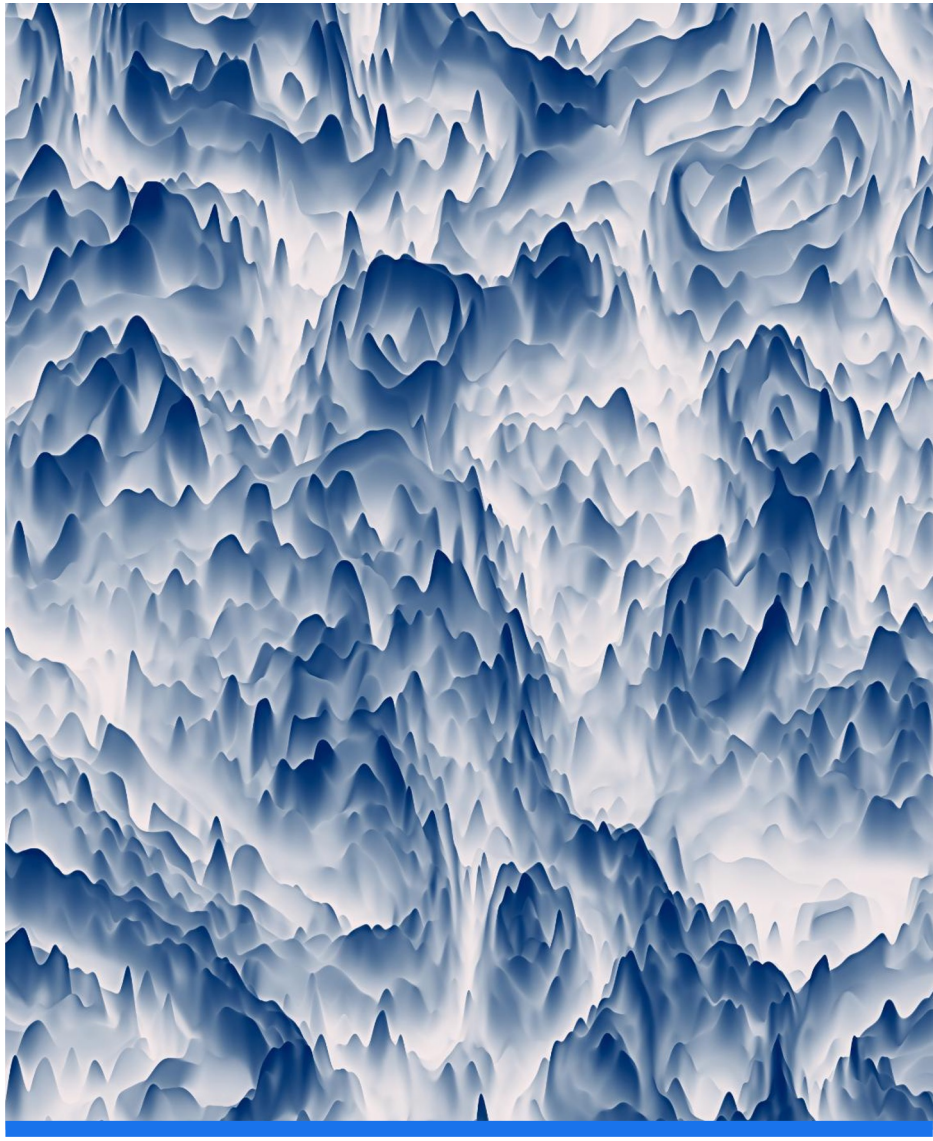
Approach 1

Approach 1: (Evaluate on whole test set)

- Produced a best model after optimizing for F1, and AUC-ROC using both Oversampling and Class Weighting approaches.
- Best fit over: XGBC, RCF, Logistic Regression, and KNN
- Identified top 5 features using feature trimming process.
- Threshold tuning for AUC-ROC experiments
- No Threshold Tuning for F1-Macro Experiments
- All elements in test set included in evaluation (test set) metrics

K-Fold
+
Oversampling or Class-Weighting

- 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 a separate validation set to improve class-wise decision boundaries and enhance macro F1 performance on the multiclass task for AUC-ROC experiments.
    - Using a separate validation set, we treat each class as an independent binary classification problem to find its unique F1-score-maximizing probability threshold. The result is a set of tailored, per-class decision cutoffs used to enhance classification performance.
- No threshold tuning was conducted on the F1 Optimized Experiments



Approach 1: Model Performance (1/2)

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

Approach 1: Model Performance (2/2)

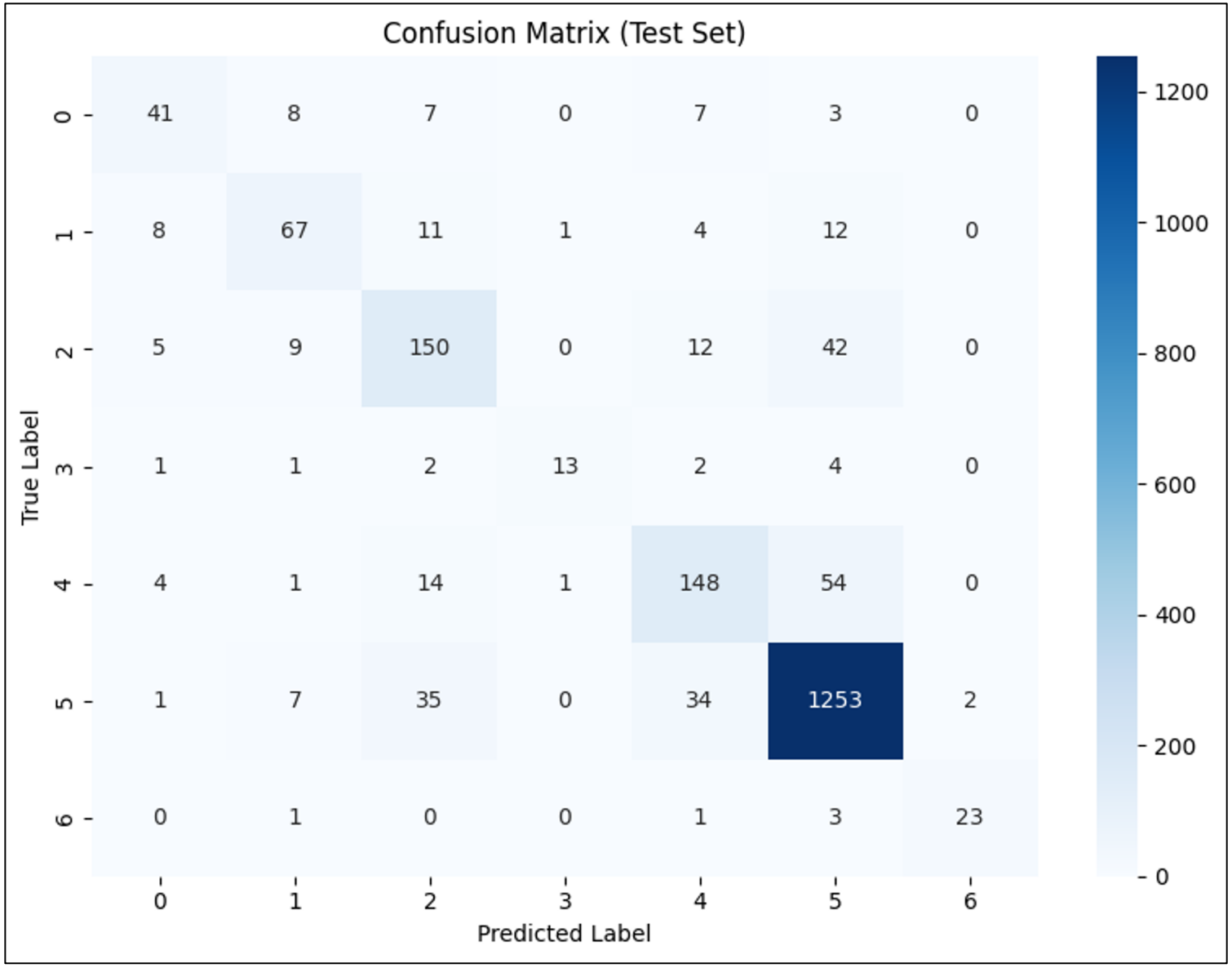
F1 Optimized Models w/ Class Balancing							
Model	Train Time (s)	Inference Time (s)	Best CV F1-Macro	Test Set AUC-ROC	Test Accuracy	Test Macro F1-Score	Best Hyperparameters
XGB	2038.62	0.05	0.6244	N/A	0.8333	0.6887	lr=0.16, max_depth=5, n_est=319
LR	1627.75	0.02	0.6369	N/A	0.8047	0.6742	C=62.71, penalty='l1'
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

Approach 1: Best Model

Class	Precision	Recall	F1-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	223
5 (Melanocytic nevi)	0.913931	0.940691	0.927118	1341
6 (Vascular lesions)	0.92	0.821429	0.867925	28
accuracy	0.850904	0.850904	0.850904	2003
macro avg	0.784738	0.707682	0.740732	2003
weighted avg	0.847812	0.850904	0.848619	2003

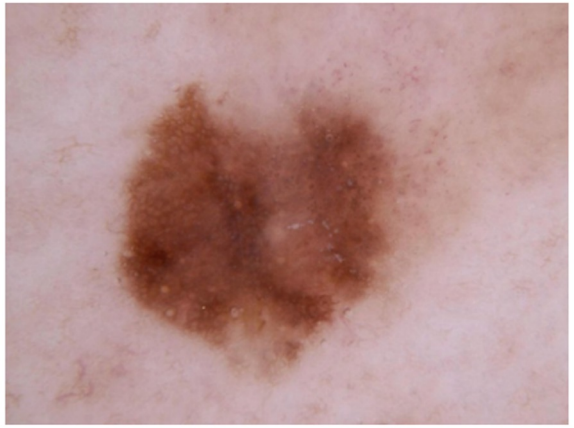
** Full test set (20% split)*

Approach 1: Best Model

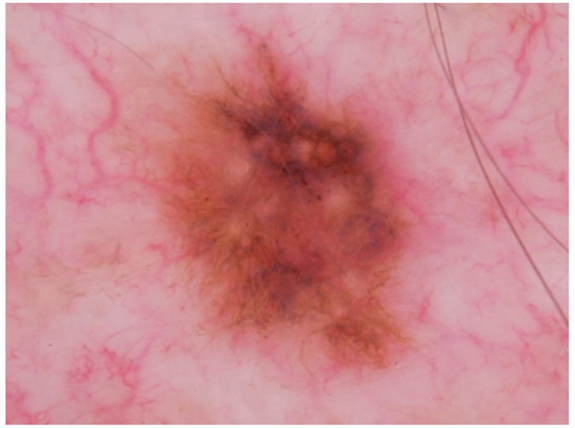


Approach 1

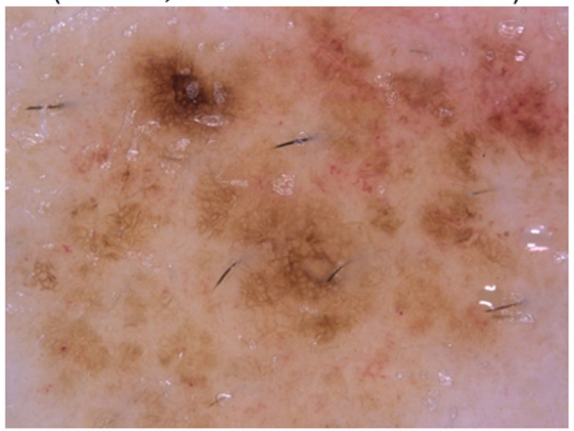
Case Analysis:
Misclassification Between Class 4 and Class 5



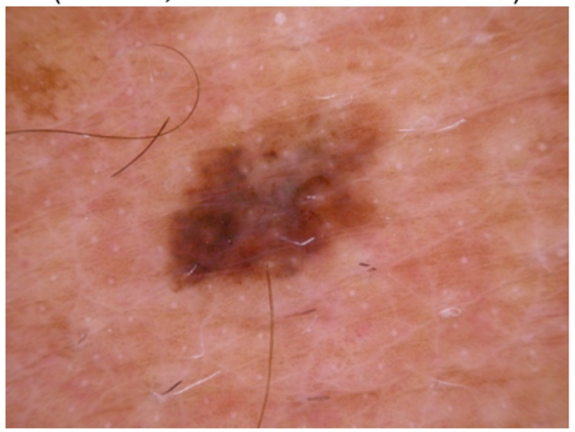
ISIC_0029272
(Class 4, misclassified as Class 5)



ISIC_0026930
(Class 4, misclassified as Class 5)



ISIC_0030771
(Class 5, misclassified as Class 4)



ISIC_0033266
(Class 5, misclassified as Class 4)

Approach 2

Approach 2: (Production Model, evaluate on subset of test set)

Using the best **model** (XGBoost Classifier) from Approach 1:

- Best Experiment Design (optimize for F1 using oversampling)
- Top 5 features

Then, applied **threshold tuning with low confidence masking** to evaluate inclusion of complex features from ViT, Resnet, and CLIP.

Note: ViT results are presented in both the runs (delta is inclusion / exclusion of threshold tuning with low confidence masking). Low confidence predictions, excluded from test set reported metrics

Approach 2: Best Model from previous iteration, plus threshold tuning with masking. But first, level setting on evaluation metrics…

Metric	Description
Best CV F1-Macro	• Measures **generalization performance during cross-validation**. • "How well model performs on unseen data — validation fold in each CV split." • F1-Macro scores from all k folds are averaged to get the final CV F1-Macro. • **Threshold-tuning dependent** — since final predictions (argmax or tuned thresholds) are used in CV scoring.
Test Set AUC-ROC	• Measures how well the model ranks or separates the positive class from the negative class using a one-vs-rest approach. Reflects the **model's discriminative ability**, not classification thresholds. - Drawback: AUC can be high even if the model is never correct in its top prediction — it measures ranking, not exact label prediction. • Evaluated on the held-out test set (20%). • **Not threshold-tuning dependent** — uses continuous prediction scores, not hard labels.
Test Accuracy	• In multi-class classification, measures how many predictions were correct, regardless of class. • Limited for imbalanced datasets — **accuracy can be dominated by majority classes.** • Computed on hold-out 20% test data. • Threshold-tuning dependent — final class labels are derived using threshold or argmax.
Test F1-Macro	• Compute F1 score for each class independently, then take the unweighted average across all classes. • **Advantage: Treats all classes equally, regardless of how rare they are.** • Ideal for imbalanced datasets like HAM10000. • Evaluated on hold-out 20% test set. • Threshold-tuning dependent — F1 is based on predicted labels after thresholding.

- All test metrics evaluated on masked hold out test set. CV F1-macro on valuation folds of training set (80%).
- F1-Macro is the **most appropriate metric** to assess performance of our imbalanced data set.
- Case: High Test-AUC-ROC, Low CV F1-Macro. May happen if no thresholding scheme was used or classification boundaries are close together .

Approach 2: Used Threshold Tuning (masking), but why is it important?

- It increases trustworthiness of results by *rejecting low confidence predictions.*
- It allows for preference for **no prediction** over a **bad prediction. That is:** “I don't know” is better than a wrong cancer diagnosis
  - "Only trust predictions if I'm at least T% confident"
- In our case when 500 low confidence predictions were masked out, F1 Macro score improved. The fine-tuned threshold probability was 0.9.

See Appendix for the Threshold Tuning Algorithm used by Approach 2

Approach 2: Best Model from previous iteration, plus threshold tuning

Approach 1: F1 optimization with random oversampling, does not use threshold tuning. Identified best model: XGBC (lr=0.29, max_depth=3, n_est=491)

Model	Best CV F1-Macro	Test Set AUC-ROC	Test Accuracy	Test Macro F1-Score	Best Hyperparameters
ViT	0.6752	0.9699	0.8509	0.7407	lr=0.29, max_depth=3, n_est=491

Approach 2: F1 optimization with random oversampling, uses threshold tuning and masking. Uses same best model from Approach 1: XGBC (lr=0.29, max_depth=3, n_est=491)

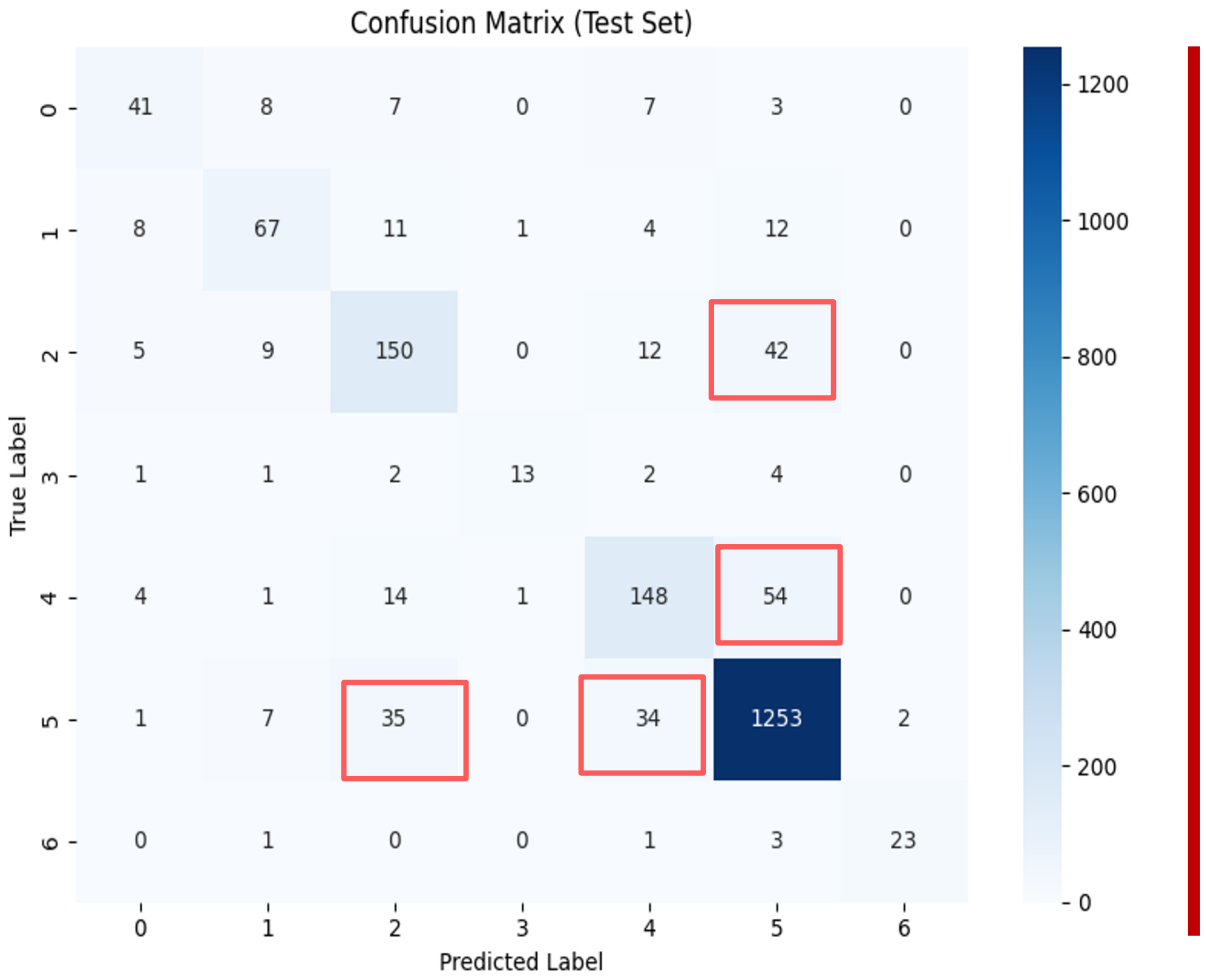
Model	Best CV F1-Macro	Test AUC-ROC	Test Accuracy	Test F1-Macro	Best Hyperparameters
ViT	**0.682016**	**0.982986**	**0.938124**	**0.857187**	lr=0.29, max_depth=3, n_est=491
ResNet	0.628186	0.974642	0.924983	0.799887	lr=0.29, max_depth=3, n_est=491
CLIP	0.617616	0.976536	0.923395	0.784727	lr=0.29, max_depth=3, n_est=491

- ViT with threshold tuning and low confidence masking (Approach 2) outperforms VIT run from Approach 1 (highest F1-Macro score) because approach 2 used threshold tuning with masking of low confidence classifications.
- Both Resnet and CLIP runs also outperform best model run from Approach 1 because of the same reason.
- Approach 1's high Test AUC-ROC but low CV F1 Macro we believe is due to the fact that no thresholding w/ masking was applied.

Approach 2 ViT: Effect of Masking Out Low Confidence Predictions Using Threshold Tuning

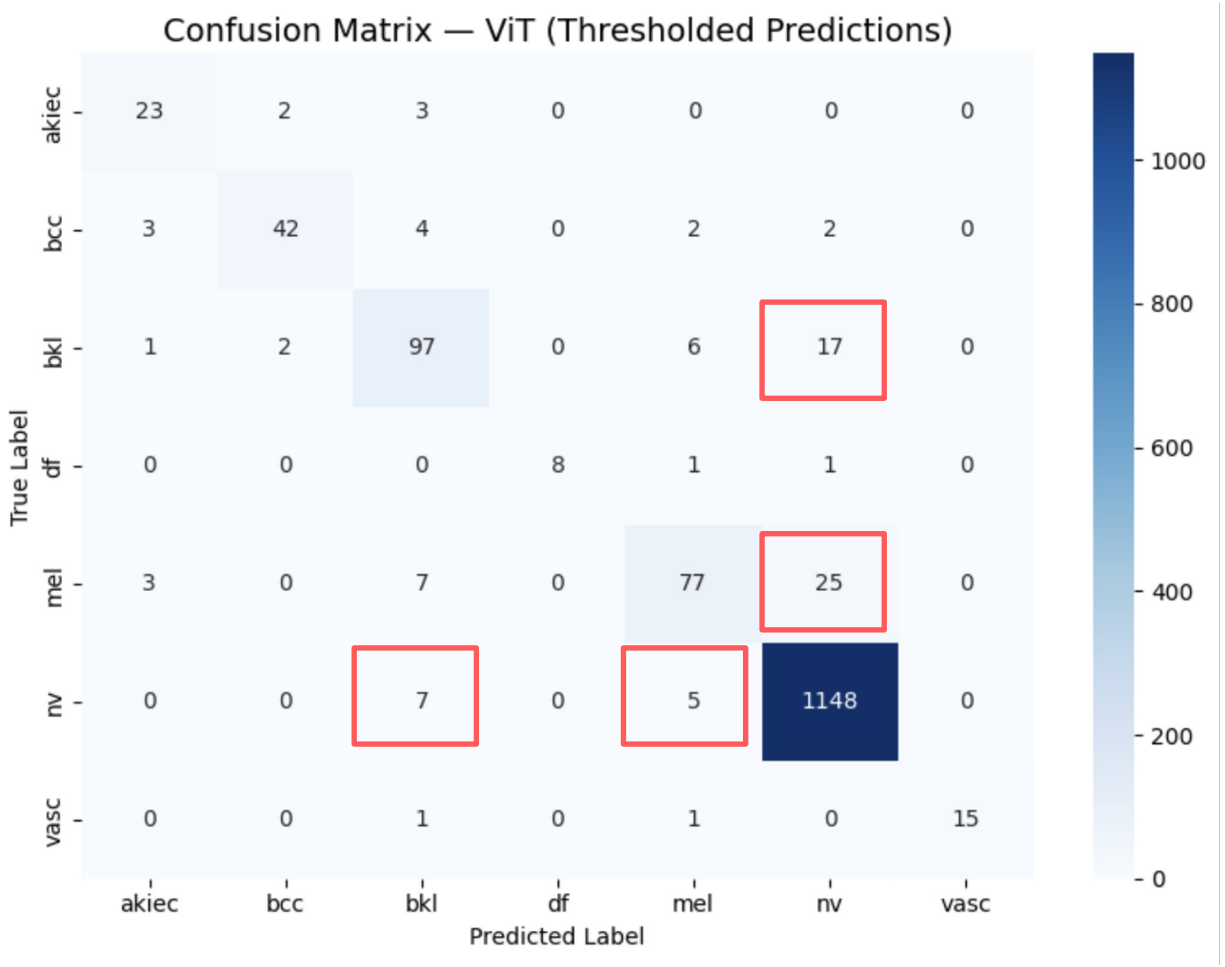
Approach 1 ViT

Confusion Matrix, No Threshold Tuning or masking



Approach 2 ViT

Confusion Matrix, Threshold Tuning with masking out of (500) Low Confidence Predictions*



* y_test shape = (2003,). y_test_masked.shape = (1503,)

Approach 1 vs 2 Test Set Differences

Class	Approach 1 Support	Approach 2 Support	Absolute Delta	Percentage Delta (A1 vs A2)
akiec	66	28	-38	- 80.85%
bcc	103	53	-50	- 64.10%
bkl	218	123	-95	- 55.72%
df	23	10	-13	- 78.79%
mel	223	112	-111	- 69.77%
nv	1341	1160	-181	- 14.47%
vasc	28	17	-11	- 48.89%
Total	2003	1503	-500	*- 58.94% avg reduction per class*

* *Average Minority class only % reduction: -66.35%*

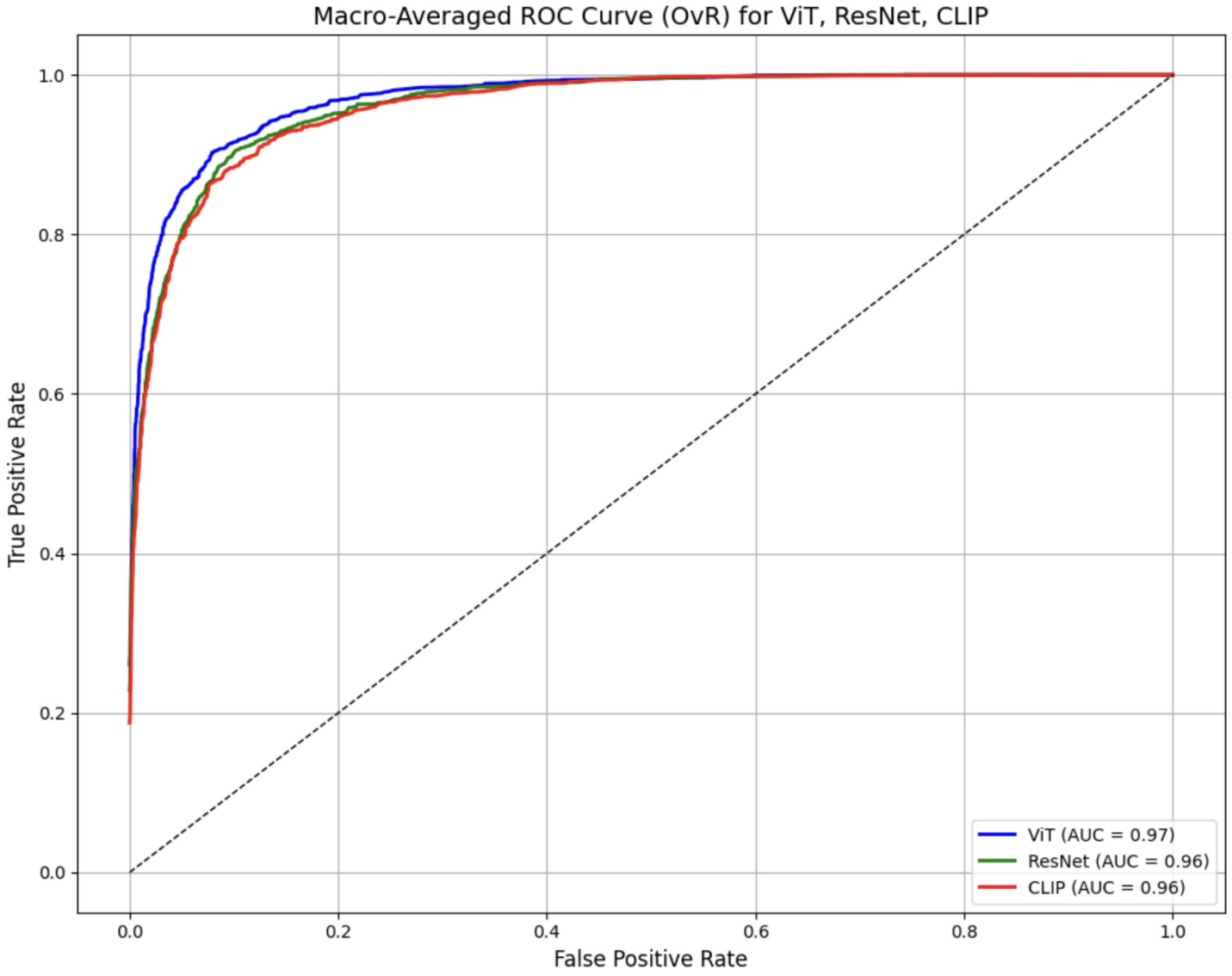
Approach 2 ViT: Per Class Metrics

Class	precision	recall	f1-score	support
akiec	0.7667	0.8214	0.7931	28
bcc	0.913	0.7925	0.8485	53
bkl	0.8151	0.7886	0.8017	123
df	1	0.8	0.8889	10
mel	0.837	0.6875	0.7549	112
nv	0.9623	0.9897	0.9758	1160
vasc	1	0.8824	0.9375	17

Majority class has the highest F1 score, but **even the most minority class (with only 10 test samples) has an F1 score of 0.8889**, suggesting a good – and high confidence – representation of minority classes in the model

* ***Full test set size: 2003***
* ***Masked test set size: 1503 (-500 low confidence predictions)***

Approach 2: Best Model from previous iteration, plus threshold tuning & low confidence masking



- ViT with threshold tuning is the best performer.
- Resnet matches CLIP's performance, and actually outperforms CLIP for low FPR-high TPR rates suggesting that ResNet may extract more domain-aligned mid-level features (e.g., edge patterns, texture) than CLIP, which is trained more generically on vision-language alignment.

Future Work:

1. **Fine-Tune Deep Learning Models** such as ViT with Contrastive Learning and QLoRA: Use contrastive learning, to better capture lesion-specific representations and improve separability across classes, and Quantized Low Rank Adaptation (QLoRA) to facilitate high dimension training on moderate compute.
2. **Simple Feature Refinement and Expansion**: revisit shape, boundary, color, and edge-based simple features—using domain expert guidance—to better capture lesion structure and improve class separability.
3. **Preprocessing Enhancements**: Use of lesion localization and contrast enhancement may help the model focus more directly on relevant lesion areas while reducing background noise.
4. **Model ensembling** improves predictions for underrepresented classes by averaging outputs from multiple models, reducing variance and enhancing stability. This aggregation helps capture weak signals, increasing the likelihood of correctly predicting rare classes.
5. **Real-World Validation** to evaluate model performance on out-of-distribution data and assess generalizability

Thank You!

Appendix Approach 2: Threshold Tuning Algorithm

1. **80/20 Train-Test Split**
   - Split the full dataset into training (80%) and test (20%) sets using stratification to preserve class distribution.
2. **Stratified K-Fold + Random Oversampling on Training Folds (excluding majority class)**

   a. For each fold:
   - Train the model on K−1K−1 folds (with oversampling).
   - Predict class probabilities on the held-out validation fold (this is the **out-of-fold (OOF)** prediction).

   b. Collect predicted class probabilities for each validation fold.

   c. At the end, you have a complete OOF probability array and corresponding ground-truth labels for the entire training set.
3. **Threshold Tuning**
   - Pass the OOF predicted probabilities and true labels to the threshold tuning function.

   a. The function tries a set of thresholds T∈[0.1,0.9]T∈[0.1,0.9] (using `np.linspace`).
     - For each threshold TT:
       - Reject (set label to `-1`) any prediction where **max class probability < T** (low confidence).
       - Compute the **macro F1 score** only on the retained (high-confidence) predictions.

       b. Return the optimal threshold T∗T∗ that **maximizes macro F1**.
4. **Retrain Final Model**
   - Train a final model MM on the full **oversampled** training data (`X_train_ROS`, `y_train_ROS`).
5. **Predict on Held-Out Test Set**
   - Use model MM to predict class probabilities on `X_test`.
6. **Convert Probabilities to Class Labels using T∗T∗**
   - For each prediction:
     - If `max(probability) >= T*`, keep `argmax(probability)` as predicted class.
     - Otherwise, reject prediction by assigning `-1`.
7. **Mask Out Low-Confidence Predictions**
   - Filter out predictions where label = `-1`.
8. **Compute Macro F1 Score**
   - Compute **macro F1** using only the **high-confidence predictions** and their true labels.