Classification of skin lesions

INFO 290T Final Project

Yuqi Liu, Matias Achour, Can Yang, Aparna Roy

April 2024

Abstract

In this project, we examine the Dermamnist dataset from MedMNIST[1], which comprises various classes of dermatological images, facilitating the diagnosis of skin diseases. The project's central aim is to build and optimize an image classifier using both traditional and advanced machine learning techniques. Initially, we engaged in significant preprocessing and feature extraction phases, employing simple feature extraction methods such as Histogram of Oriented Gradients (HOG), Local Binary Patterns (LBP), and HSV color histograms alongside complex features extracted from the ResNet18 pretrained model. Subsequent to feature extraction, dimensionality reduction was performed using Principal Component Analysis (PCA) and visualized through t-SNE to enhance interpretability and generalizability. For the classification, we developed models using Support Vector Machines (SVM), Random Forest, and Logistic Regression, optimizing them through hyperparameter tuning. The project benchmarks these models based on their accuracy and computational efficiency, providing insights into the trade-offs between model complexity and performance. This report details each step of the process, from data acquisition and preprocessing to model training and evaluation, highlighting the challenges encountered and the methodologies employed to address them, aiming for a balance between accuracy and efficiency in the classifiers developed.

1 Introduction

This dataset is a collection of dermatological images, specifically designed to facilitate machine learning and computer vision applications in the medical field. It includes a total of 10,015 images, each resized to 224x224 pixels, and categorized into seven classes representing different skin conditions. The distribution of images across these classes is detailed in Table 1.

Class	**Images**
Actinic keratoses and intraepithelial carcinoma (akiec)	327
Basal cell carcinoma (bcc)	514
Benign keratosis-like lesions (bkl)	1099
Dermatofibroma (df)	115
Melanocytic nevi (nv)	1113
Vascular lesions (vasc)	6705
Melanoma (mel)	142

Table 1: Distribution of images across different classes in the Dermamnist dataset

For convenience, the abbreviations in parentheses will be used to refer to these classes throughout this report.
The dataset is strategically partitioned into training, validation, and testing sets, following a ratio of 70% for training, 10% for validation, and 20% for testing. This allocation ensures a robust

training regime while maintaining adequate data for validation and generalization testing. Below, Figure 1 presents a sample image from each class.

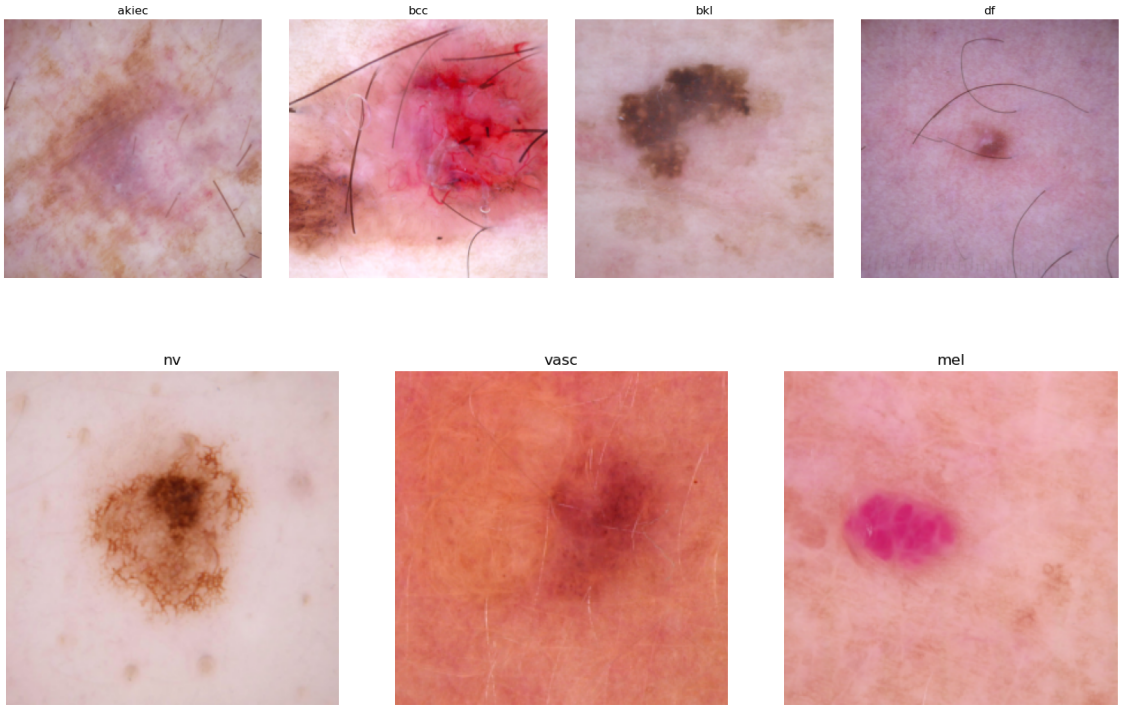


Figure 1: Example images from each category

In subsequent sections, we will delve into the detailed methodology of feature extraction, discuss the model training process, and present the results, including accuracy and efficiency metrics of the different models. This will provide insights into the trade-offs involved in the classifier design and the potential improvements for future iterations of the project.

2 Data Preprocessing

2.1 From RGB to Grayscale

In many of the feature extraction techniques that will be presented, like FFT and LBP, the images were transformed form RGB to grayscale using the in-built python function, which converts using

$$gray = 0.2989 \cdot red + 0.5870 \cdot green + 0.1140 \cdot blue \tag{1}$$

2.2 Image Augmentation

To enhance the robustness of our classifiers and mitigate the challenges posed by the limited availability of data for certain lesion types, we implemented a strategic image augmentation protocol. All lesion categories, with the exception of category 5—which already had a substantial representation in the dataset—underwent augmentation. This process involved horizontally and vertically flipping the images, as well as adjusting their brightness levels. Such transformations not only diversify the training data but also simulate various photographic conditions, thus enabling our models to learn more generalized features from the enriched dataset.

2.3 Image Denoise

We use the method of total variation denoising, which means minimizing the total variation while the denoised image remain similar to the original image, which can be expressed by the Rudin-Osher-Fatemi(ROF) minimization problem: $\min_u \sum_{i=0}^{N-1} \left(|\nabla u_i| + \frac{\lambda}{2}(f_i - u_i)^2\right)$, where λ is a positive parameter.The first term of the cost function is the total variation; where the second term represents data fidelity. As $\lambda \to 0$, the total variation term dominates, forcing the solution to have smaller variation and looking less like the input data and see Figure 2 to check the effect when $\lambda = 10$

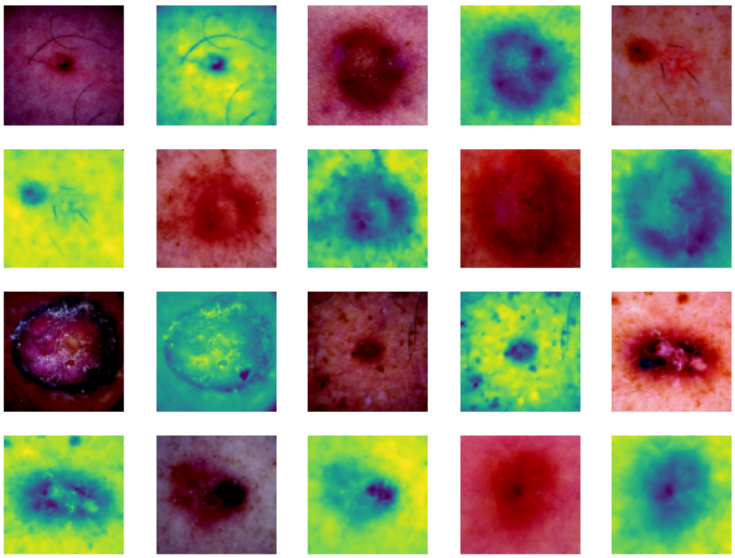


Figure 2: Before and after preprocess examples from images in class 3

3 Feature extraction

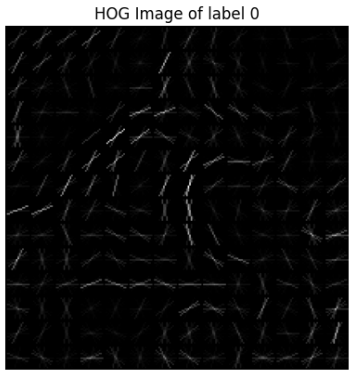
3.1 Histogram of Gradient (HOG) feature extraction

HOG (Histogram of Gradient) is good at extracting the intensity and direction of local gradients in an image and literature shows that it performs good in skin diseases classification problem [2]. HOG captures both data about edges, as well as direction of the edges, by splitting the gradients into bins. In this project, we have used six orientations, as well as block sizes of 16x16 pixels. For each of the sample images in figure 1, we have performed HOG visualization, which is shown in figure 3.
With the given parameters, we get a total of 1176 features.

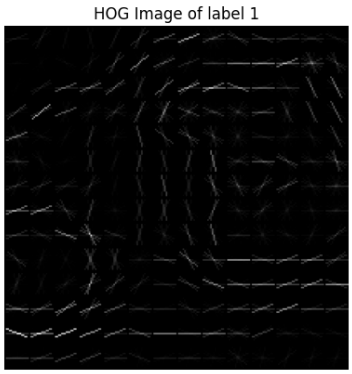
3.2 FFT Feature

FFT(Fast Fourier Transform) converts a medical image from its original spatial domain to the frequency domains.[3]The 2-D discrete Fourier Transform could be calculated as

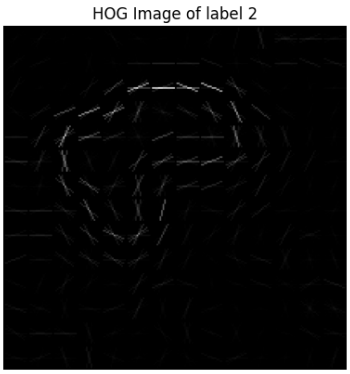
$$X_k = \sum_{n=0}^{N-1} e^{-2\pi i k \cdot (n/N)} x_n$$



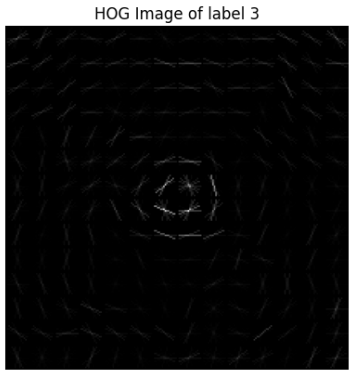
(a) HOG visualization for akiec



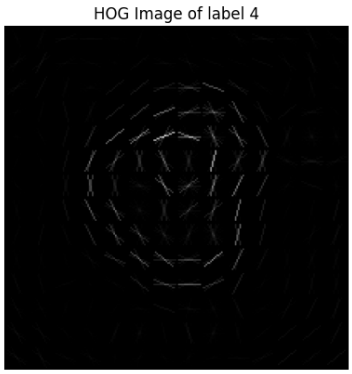
(b) HOG visualization for bcc



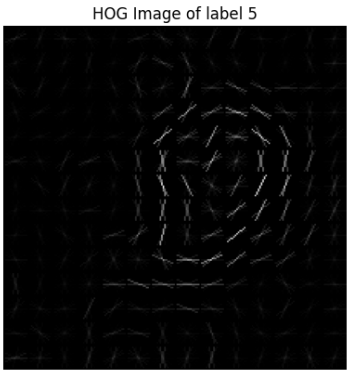
(c) HOG visualization for bkl



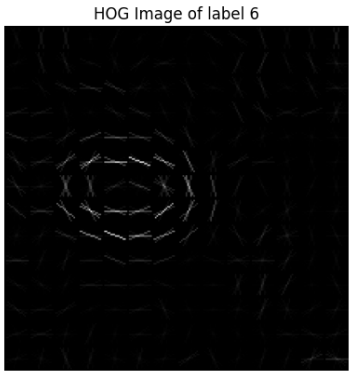
(d) HOG visualization for df



(e) HOG visualization for nv



(f) HOG visualization for vasc



(g) HOG visualization for mel

Figure 3: HOG feature extraction

3.2.1 Spectrum

The spectrum is just displaying the intensity of different frequency components within the image. Before that, first we need to shift the zero frequency component of the result to the center for easier analysis. An example of it is as follows, see Figure 4. This illustrates the magnitude spectrum on a random sample image, transformed to grayscale.

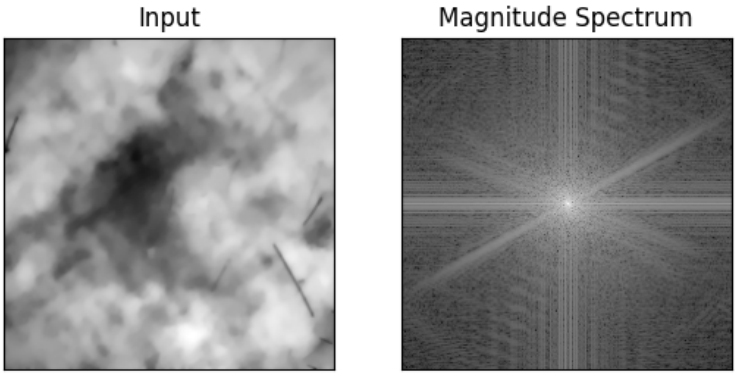


Figure 4: Before and after fft

3.2.2 Spectral entropy

Spectral entropy is used to quantify how much information the spectral distribution holds.And it could be calculated as follows:

$$entropy = -\sum[(spectrum/\sum specturm) \cdot log(spectrum/\sum specturm)]$$

A high spectral entropy means that the frequency distribution is relatively uniform, indicating that the image contains a rich texture and details; low spectral entropy suggests that certain frequency components are dominant, which may signify that the image is smoother or contains a single texture or shape. And for our experiments, we got those three features where Spectrum is averaged over all pixels in the image.

3.3 GLCM Feature

The Gray Level Co-occurrence Matrix (GLCM) is based on grayscale image used to extract texture feature based on spatial intensity relationship of pixels.And a GLCM is built where the number of rows and columns is equal to the number of gray levels in the image. The matrix element $P = (i, j, d)$ is the relative frequency with which two pixels, separated by a pixel distance d, occur within a given neighborhood, one with intensity i and the other with intensity j. The angle θ is the direction of the pixel pair.And here we set offset=10,angles=90 degree. Multiple properties can be defined for GLCM, some of the described below.

3.3.1 Contrast

Contrast measures the local variations in the GLCM. High contrast values indicate a high variation in intensity values within an image. And it is calculated as

$$\sum_{i,j=0}^{levels-1} GLCM_{i,j}(i-j)^2$$

3.3.2 Energy

Energy provides the sum of squared elements in the GLCM which measures textural uniformity. Higher energy values indicate more uniformity or less textural variation.

3.3.3 Homogeneity

Homogeneiry measures the closeness of the distribution of elements in the GLCM to the GLCM diagonal. Higher values indicate more homogeneity or similarity in the intensity values.

$$\sum_{i,j=0}^{levels-1} GLCM_{i,j}/(1+(i-j)^2)$$

3.3.4 Correlation

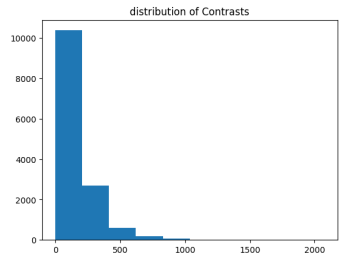
Correlation measures how correlated a pixel is to its neighbor over the whole image. It ranges from -1 to 1, where a value close to 1 implies a strong correlation and value close to -1 implies a weak correlation.

3.3.5 Dissmilarity

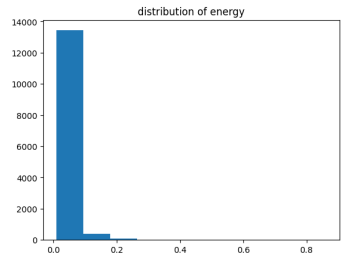
Measures the average difference in intensity between neighboring pixelsin GLCM. High dissimilarity values indicate greater heterogeneity in texture.The calculation is

$$\sum_{i,j=0}^{levels-1} GLCM_{i,j}\|i-j\|$$

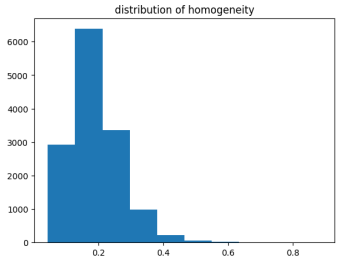
Using the presented properties, the plots below show the distribution of each property in the dataset.



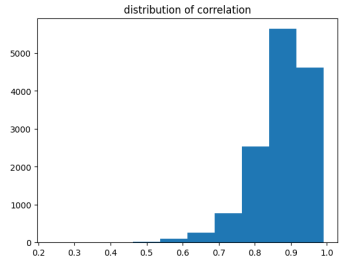
(a) distribution of feature contrasts



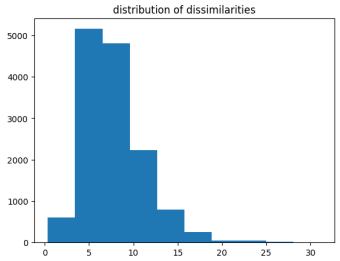
(b) distribution of feature energy



(c) distribution of feature homogeneity



(d) distribution of feature correlation



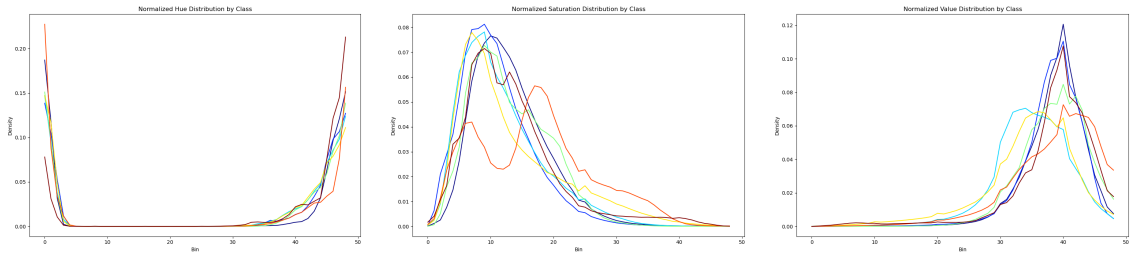
(e) distribution of feature dissimilarity

Figure 5: GLCM Feature Distribution

3.4 HSV color histogram

Understanding the distribution of color within dermatological images is essential for differentiating skin conditions, which often manifest distinct color patterns corresponding to various disease

states. The HSV (Hue, Saturation, Value) color model provides a more intuitive color space for this purpose compared to RGB, as it separates color description (hue) from intensity (value), with saturation describing the depth of the pigment.
In our feature extraction process, we employ HSV color histograms to capture these three components across the dataset. Hue can indicate the type of lesion (e.g., redness associated with inflammatory conditions), saturation can reflect the depth or severity of the lesion, and value may highlight the brightness or prominence of an image feature.
For each image, we calculate the HSV histograms by aggregating the pixels' values into bins, each with a range of 50. The cumulative count of pixels within these bins furnishes us with a 150-dimensional feature vector that encapsulates the color distribution of the image. The ensuing histogram plots afford a visual representation of the color distribution across different classes, as depicted below.



(a) Hue Distribution (b) Saturation Distribution (c) Value Distribution

Figure 6: Histograms of Hue, Saturation, and Value

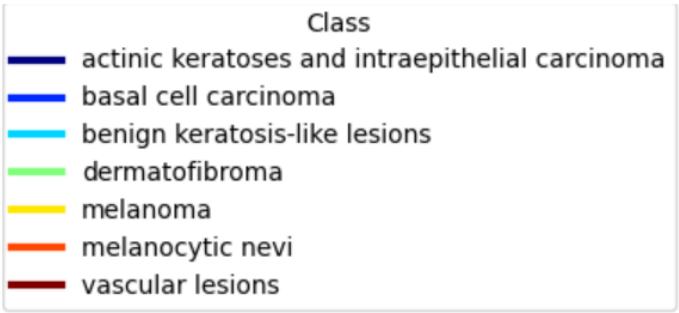


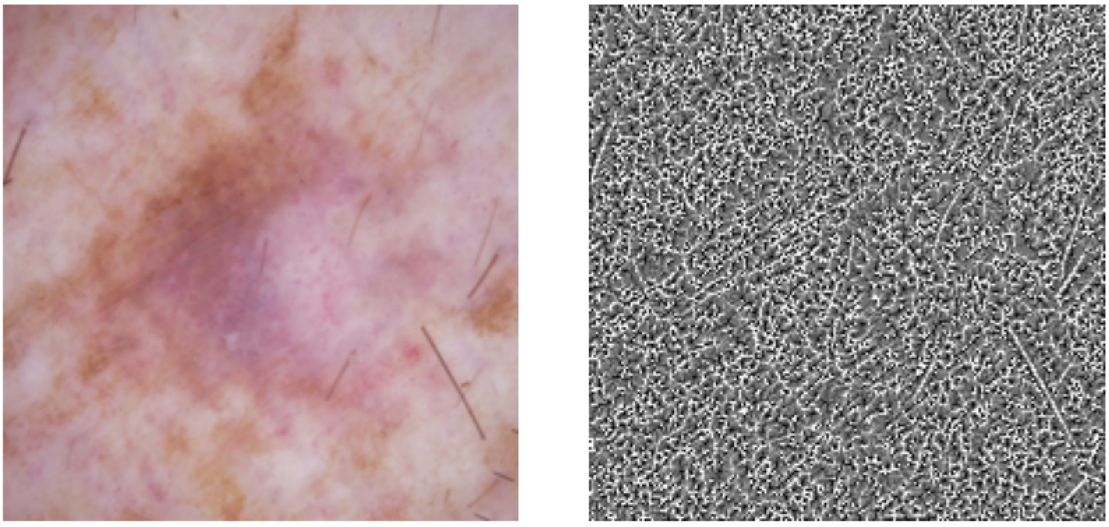
Figure 7: Legend of Classes

Distinct variations in color properties are evident among the classes. Notably, the class representing *melanocytic nevi* is characterized by a unique saturation and value distribution, highlighting the utility of HSV features in differentiating this class. Although hue values appear to be uniformly distributed across classes, this observation correlates with the visual similarities in coloration when examining image samples for each category (see figure 1). This uniformity may also reflect variations in skin tone that are unrelated to pathological conditions, thus introducing a form of 'noise' into the data. Nevertheless, the HSV color space remains a valuable asset in distinguishing lesions with distinct chromatic attributes.

3.5 Local Binary Pattern (LBP)

Local binary pattern (LBP) captures the local texture information by comparing each pixel with its neighbors. A high LBP value suggests a high contrast between a pixel and its surroundings, while a low value indicates low contrast. Thus, areas with rapid intensity change, like edges or corners, will typically have higher LBP values.

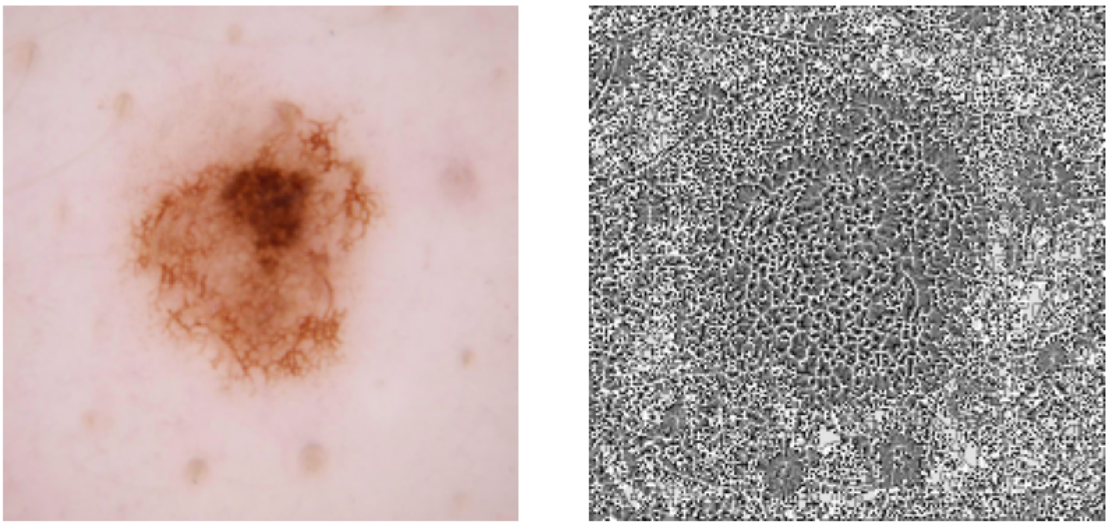
This textural feature is particularly pertinent in dermatological imaging, where the contrast in skin lesions can vary significantly due to the inherent color differences among various skin conditions. For instance, lesions that are inherently darker may yield distinct LBP signatures in comparison to lighter lesions. To illustrate the concept, Figure 8 and 9 presents images from the *akiec* and *nv* classes along with their respective LBP feature visualizations.



(a) Original image form the *akiec* class

(b) LBP feature of the image

Figure 8: Original image from class *akiec*, with corresponding LBP feature

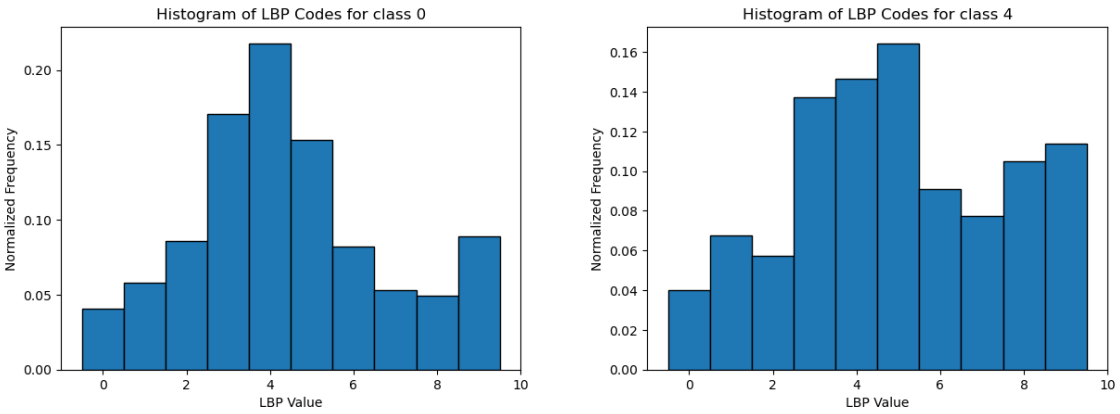


(a) Original image form the *nv* class

(b) LBP feature of the image

Figure 9: Original image from class *nv*, with corresponding LBP feature

To further discern the texture information encoded by LBP, we analyze the histogram of the LBP codes, which aggregates the frequency distribution of the binary patterns throughout an image. The histograms for the *akiec* and *nv* classes are depicted in Figure 10.
A comparative examination of these histograms concludes that the *nv* class exhibits higher LBP values, indicative of greater textural contrast within the lesions. This observation aligns with



(a) LBP values of class *akiec*

(b) LBP values of class *nv*

Figure 10: Histograms of Hue, Saturation, and Value

visual analyses; lesions classified as *nv* typically possess darker hues and thus, higher contrast.

3.6 Feature Extraction from Pre-trained Neural Network

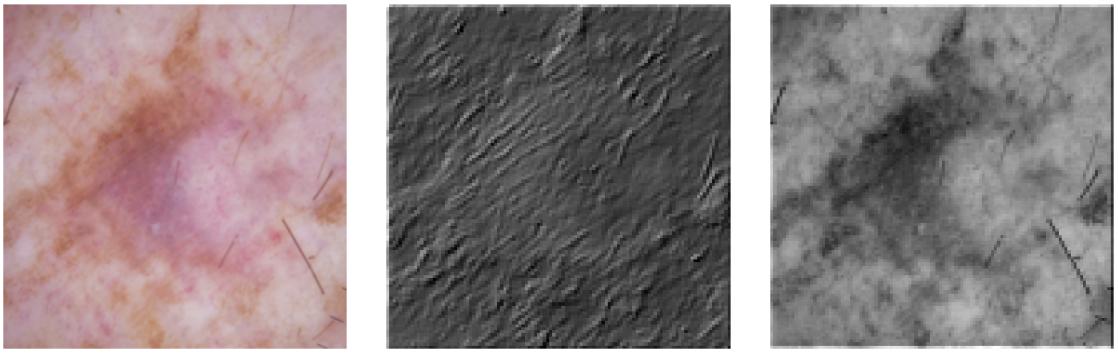
The advent of deep learning has introduced numerous pre-trained neural networks, each meticulously optimized for various tasks, including image classification. Within the domain of medical imaging, ResNet architectures have consistently demonstrated exemplary performance. While ResNet-50 offers a deeper and more complex structure, ResNet-18 is recognized for its computational efficiency. Given our constraints on computational resources, ResNet-18 was selected for feature extraction from our dermatological image dataset.
ResNet-18 is a convolutional neural network (CNN) with an architecture designed to facilitate the learning of deep representations without succumbing to the challenges of training very deep networks. The detailed structure of ResNet-18 is represented in Table 2.

Layer Name	Output Size	ResNet-18
conv1	$112 \times 112 \times 64$	7×7, 64, stride 2
conv2_x	$56 \times 56 \times 64$	3×3 max pool, stride 2 $\begin{bmatrix} 3 \times 3, & 64 \\ 3 \times 3, & 64 \end{bmatrix} \times 2$
conv3_x	$28 \times 28 \times 128$	$\begin{bmatrix} 3 \times 3, & 128 \\ 3 \times 3, & 128 \end{bmatrix} \times 2$
conv4_x	$14 \times 14 \times 256$	$\begin{bmatrix} 3 \times 3, & 256 \\ 3 \times 3, & 256 \end{bmatrix} \times 2$
conv5_x	$7 \times 7 \times 512$	$\begin{bmatrix} 3 \times 3, & 512 \\ 3 \times 3, & 512 \end{bmatrix} \times 2$
average pool	$1 \times 1 \times 512$	7×7 average pool
fully connected	1000	512×1000 fully connections
softmax	1000	

Table 2: ResNet-18

The architecture culminates in a layer outputting a 512-dimensional vector. This vector, which is

flattened from the final convolutional layer, encompasses a wealth of features extracted from the input images. It is this comprehensive feature set that we extract, thus obtaining the activations from the penultimate layer to yield a 512-feature vector for each image.
To illustrate the type of features that the CNN captures, one can visualize the feature maps from the convolutional layers. Below, we display an original dermatological image alongside two representative feature maps extracted from the initial convolutional layer.



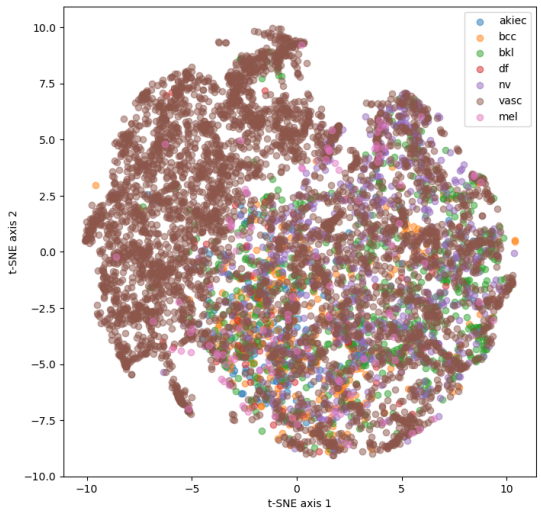
(a) Original image (b) Feature map from first layer (c) Another map from first layer

Figure 11: Random image from dataset, together with two corresponding feature maps

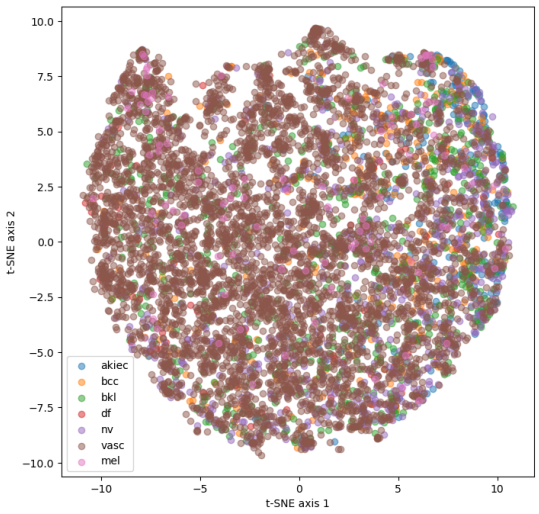
Selected arbitrarily from the 64 potential maps at the first layer, these feature maps reveal the CNN's ability to extract various aspects of the image data. For instance, one map may emphasize the presence of edges—both diagonal and horizontal—while another may capture the distribution of specific color textures, such as the brown regions in the depicted image.

3.7 T-distributed Stochastic Neighbor Embedding (t-SNE) Visualization

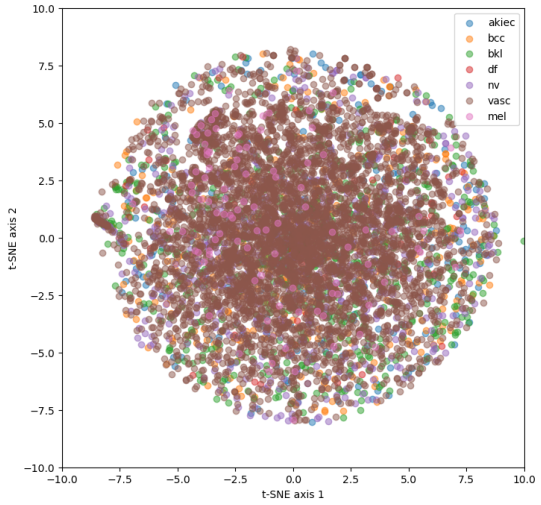
For each of these features, we have done a t-SNE visualization. t-SNE is used to visualize high-dimensional data in lower dimensions, and can be used to visualize how good a feature can distinguish between the varios categories.
In our exploration of the dataset, we have leveraged several feature extraction techniques, including Histogram of Oriented Gradients (HOG), Hue-Saturation-Value (HSV) histograms, Local Binary Pattern (LBP), GLCM, FFT and features derived from the pre-trained ResNet-18 network. By testing, as well as looking at the most powerful feature extractions for our dataset, we decided to do feature extraction with HOG, HSV, LBP and the pre-trained Resnet-18 network. To assess the discriminative power of each of the chosen feature sets, we have employed t-distributed Stochastic Neighbor Embedding (t-SNE), a non-linear dimensionality reduction technique that is particularly well-suited for visualizing high-dimensional data in two or three dimensions.
Figure 12a, 12b, 13a, 13b and 14 shows the t-SNE visualization for each of the dense feature sets we use in classification, as well as t-SNE visualization of our final, concatenated, feature vector.
Upon examination of these visualizations, it is apparent that while some features do not exhibit a clear separation between classes, there are notable exceptions. The t-SNE plot for LBP, for instance, indicates a degree of segregation for the *akiec* class, echoing the previous discussions on LBP's sensitivity to texture differences within this category. Similarly, the features extracted via ResNet-18 demonstrate a discernible separation of the *mel* class, colored pink, from other classes. It also provides a reasonable delineation of the *vasc* class, which is significant given the class imbalance present within the dataset.
Despite initial appearances suggesting limited class separation in feature space, the subsequent classification results, as detailed in Section 4, suggest that these features can indeed provide a surprisingly effective basis for classifying the seemingly overlapping class distributions.



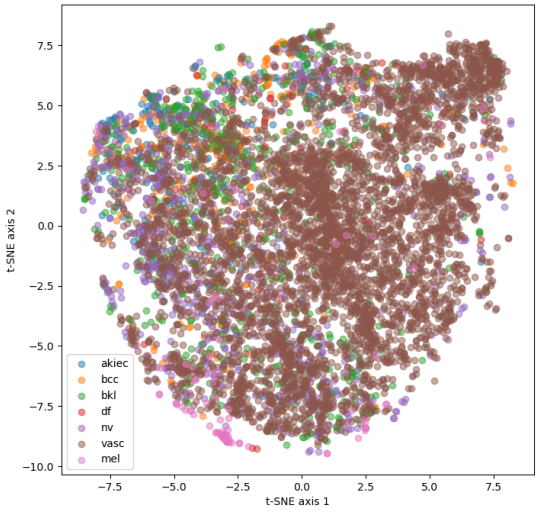
(a) t-SNE for HSV



(b) t-SNE for LBP



(a) t-SNE for HOG



(b) t-SNE for ResNet-18

3.8 Principal Component Analysis (PCA)

Principal Component Analysis (PCA) serves as a cornerstone technique in the dimensionality reduction of datasets, streamlining the feature set while preserving the essence of the original data. By mapping the data to a new coordinate system, PCA accentuates the variance along principal components, thereby simplifying the complexity of high-dimensional spaces.

Figure 15 encapsulates the efficacy of PCA across the various feature extraction methodologies employed in the classification. Each curve signifies the proportion of the dataset's variance maintained against the accruing number of principal components.

The HSV histogram trajectory on the PCA plot ascends sharply before plateauing, a signature indicating that the color space is well-represented by a compact subset of components. The HOG features follow a similar path, however with a more gradual ascent, suggesting that while the primary components capture a considerable amount of information, the complexity of the textural patterns calls for a broader spectrum of components to achieve comprehensive variance coverage. In the case of LBP, the plot takes a swift and short-lived rise, reflective of the method's limitation to a mere 10 features. Such a concise feature set conveys that the majority of textural information

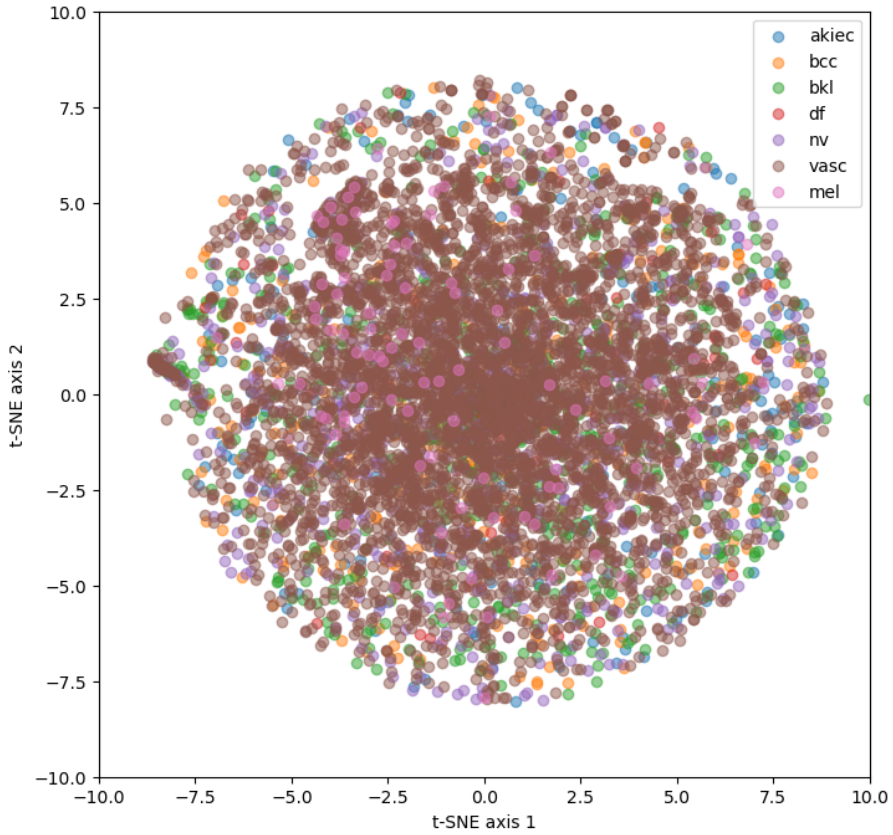


Figure 14: t-SNE for all features concatenated

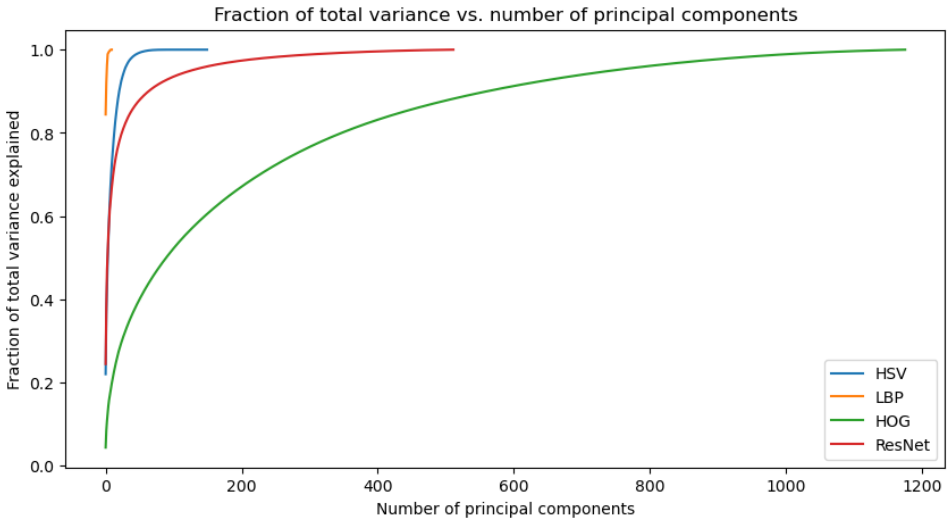


Figure 15: Fraction of total variance versus the number of principal components for each feature extraction method and their aggregation.

is concentrated within a narrow scope, and while the steep curve might suggest high variance within those features, the shape of the curve also highlights the method's inability to encapsulate a wider range of textural variance.

The PCA curve for ResNet-18 features starts with a rapid rise, showing that the network captures a lot of information with just a few components. But as more components are added, the rate of new information gained slows down. This suggests that after a certain point, each new component adds less and less to our understanding of the data.
Furthermore, figure 16 combines all the features to show the overall variance.

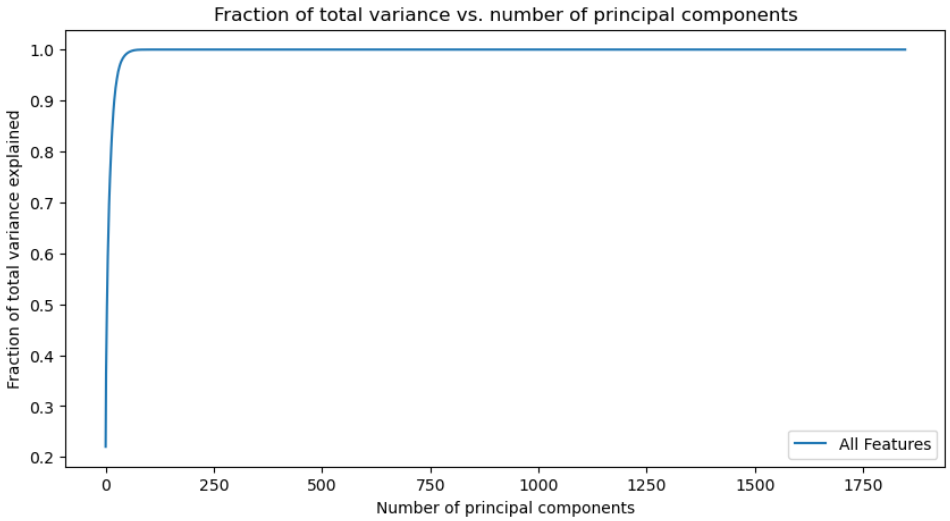


Figure 16: PCA for all features concatenated

In conclusion, selecting the appropriate number of principal components is crucial for preserving the integrity of the dataset's information while optimizing for computational efficiency. A careful balance must be struck to retain enough components to encapsulate the majority of the data's variance, yet few enough to maintain a streamlined feature set for effective model training. In section 5, we will go more in detail to find values of PCA which balances this trade-off.

4 Classification

In this section, we will train the extracted features from HSV, LBP, HOG and ResNet-18 with three different classifiers. We will look at classification using logistic regression, random forest and SVM. Before all of the classification, we concatenate all the features into a single feature matrix. This gives a total of 1848 features. To prevent some features from dominating and getting problems with big differences in absolute values, we standarize our feature matrix. After this, we will look at the differences in results before and after hyperparameter tuning. Finally, we will look at how PCA gives a trade-off between accuracy and efficiency.

4.1 Results before hyperparameter tuning

Logistic Regression is a statistical model that, despite its name, is used for classification tasks. It predicts the probability of the target variable being one of two classes by applying a logistic function to a linear combination of the input features. This method is particularly effective when the relationship between the feature set and the classification outcome is expected to be linear. The performance is shown in table 3.
Random Forest is an ensemble learning method that operates by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes predicted by individual trees. Due to its ensemble nature, random forests are less likely to overfit to the training data compared to individual decision trees and are adept at handling datasets with a high dimensionality. The performance of this classifier is shown in table 4.

Hyperparameter	Train Accuracy	Val Accuracy	time spent
C=10, max iterations=1000	1.0	0.63	46.8s

Table 3: Logistic regression performance

Hyperparameter	Train Accuracy	Val Accuracy	Time spent
max depth=20	1.0	0.57	32s

Table 4: Random Forest performance

Support Vector Machine (SVM) is a powerful and versatile classification technique that finds the optimal hyperplane that maximizes the margin between different classes in the feature space. SVMs are particularly well-suited for classification tasks where the classes are clearly separable and the datasets are not too large. They are also capable of handling non-linear relationships through the use of kernel functions.

Hyperparameter	Train Accuracy	Valid Accuracy	Time spent
C=1,, kernel='poly'	0.86	0.74	49.3s

Table 5: SVM Performance

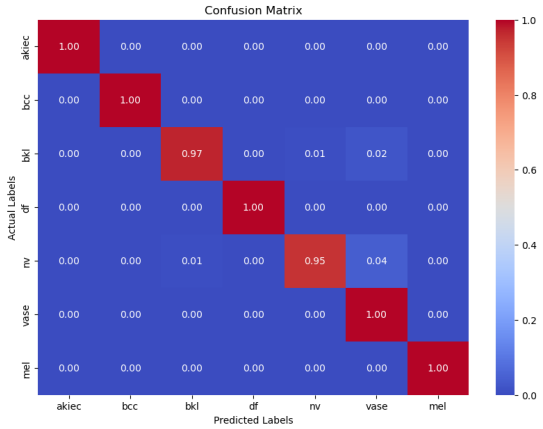
4.2 Generalizability

In the previous section, all our results came from no tuning of hyperparameters. To optimize generalizability, as well as avoiding potential overfitting, we did hyperparameter tuning to find the optimal hyperparameters for each of the classifiers. Table 6 shows the results for best parameters and corresponding accuracies for each of the classifiers.

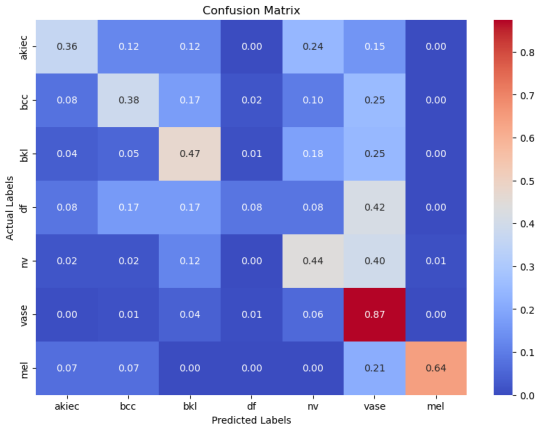
Model	**Parameter**	**Search Values**	**Best Value**	**Best Accuracy**
Logistic Regression	C solver	0.01, 0.1, 1, 10 'liblinear', 'lbfgs'	0.1 'liblinear'	0.73
Random Forest	n_estimators max_depth min_samples_split	10, 50, 100 None, 10, 20, 30 2, 5, 10	100 None 2	0.699
SVM	C kernel	0.001, 0.01, 0.1, 1 'linear', 'rbf'	1 'rbf'	0.77

Table 6: Hyperparameter search results

These were the best possible accuracies we could achieve. According to the scientific research paper Nature, the benchmark for accuracy for this dataset lies between 0.71-0.73, with the best neural network achieving an accuracy of 0.76 using the pre-trained model of Google AutoML Vision [1]. Given this, both the logistic regression and the SVM achieves relatively high accuracies after hyperparameter tuning, and the performance is definetly improved compared to the results before tuning. To better understand what goes wrong during the classification, we can look at confusion matrix for all classifiers, shown in figure 17, 18 and 19.

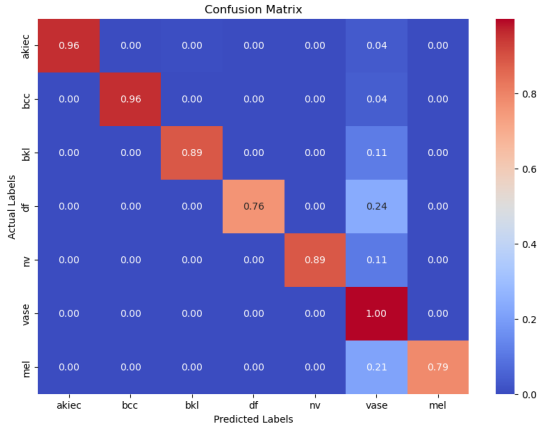


(a) Confusion matrix of logistic regression training accuracy

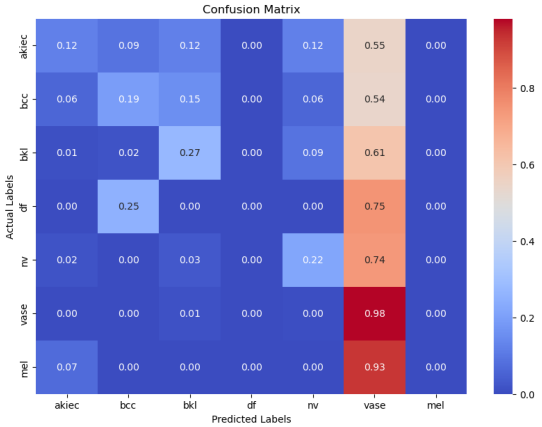


(b) Confusion matrix of logistic regression validation accuracy

Figure 17: Confusion matrix of logistic regression

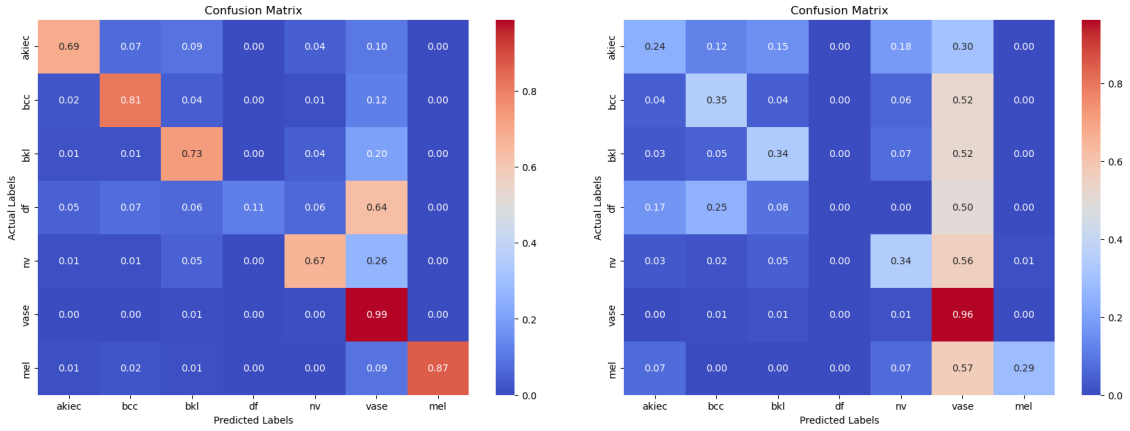


(a) Confusion matrix of random forest training accuracy



(b) Confusion matrix of random forest validation accuracy

Figure 18: Confusion matrix of random forest

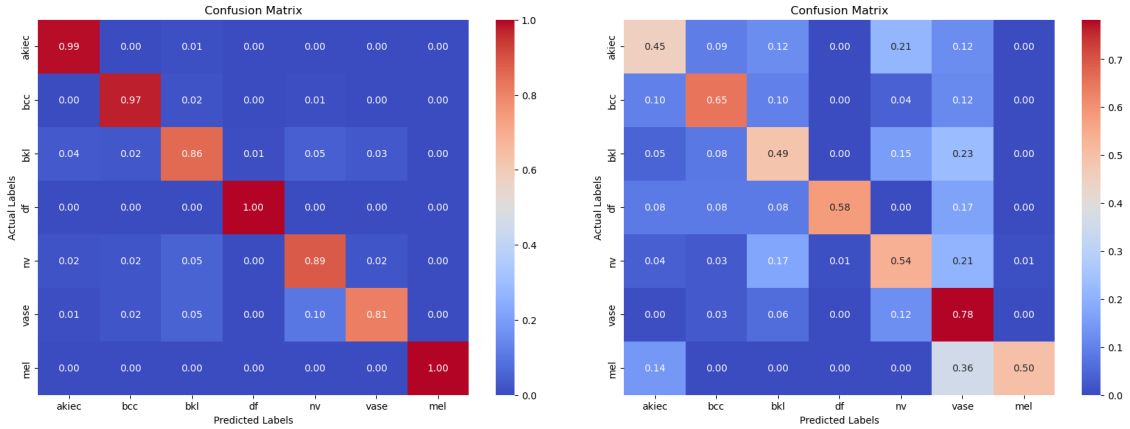


(a) Confusion matrix of SVM training accuracy

(b) Confusion matrix of SVM validation accuracy

Figure 19: Confusion matrix of SVM

We see that for the training accuracy, all the models performs well, but then struggles with overfitting for validation. Furthermore, *vasc* has a much higher validation accuracy than the rest of the classes, and many classes are heavily misclassified as *vasc*. By looking at table 1, we can assume that this is due to the big imbalance in training points for the classes. The majority of the data is from class *vasc*, thus giving origin to misclassifications.
A way of conquering this issue is by balancing the classes by assigning weights to each class. Only looking at the SVM, that gives the confusion matrices as shown in figure 20



(a) Confusion matrix of SVM training accuracy after weighting

(b) Confusion matrix of SVM validation accuracy after weighting

Figure 20: Confusion matrix of SVM after weighting

What is clear is that all classes has gone up in accuracy, while *vasc* has gone down. This will decrease the total performance, but give a more stable classifier. This is more desirable, as we achieve higher generalizability.

5 Efficiency

In section 3 we discussed PCA, and its pros and cons. As mentioned then, PCA can be used as a trade-off between efficiency and performance, and in this last section we will try to optimize one

solution for efficiency and one for accuracy. Table 7 gives results from running the classifiers with their optimal hyperparameters for different values of PCA.

Model	PCA: 1200		PCA: 500		PCA: 50	
	Accuracy	Run Time	Accuracy	Run Time	Accuracy	Run Time
Logistic Regression	0.705	40.2 s	0.699	18.5s	0.51	11.12s
Random Forest	0.697	38.79s	0.689	12.35s	0.42	9.88s
SVM	0.755	51.8s	0.694	42.8s	0.67	18.8 s

Table 7: Model accuracy and run time comparison for different values of PCA

As expected, the accuracies drops when the amount of PCA components falls too low. But in between, for 500 components, we still get a reasonably good accuracy, while in two of three cases reducing the computaton time by over half. We can conclude that the models which optimizes efficiency, without dropping too low in accuracy, lies in the range of around 500 PCA components.

6 Conclusion

This project embarked on the ambitious task of classifying medical images of skin lesions, harnessing the power of feature extraction and machine learning classifiers. By utilizing a diverse array of techniques—HSV histograms, Local Binary Patterns (LBP), Histogram of Oriented Gradients (HOG), and features extracted from the pre-trained ResNet-18 convolutional neural network—a comprehensive feature set encompassing 1848 distinct attributes was developed.

Through the application of logistic regression, random forest, and Support Vector Machine (SVM) classifiers, the project not only achieved notable classification results but also highlighted the strength of SVM, which culminated in a validation accuracy of 0.77 after hyperparameter optimization. Logistic regression followed with a best score of 0.73, while the random forest model trailed slightly at 0.699.

Key strategies were explored to enhance the generalizability of these models and to address the prevalent issue of class imbalance, even at the expense of marginal accuracy reduction. These discussions provided valuable insights into building robust and reliable classifiers.

Moreover, the study delved into the dimensionality reduction through Principal Component Analysis (PCA), effectively doubling the efficiency of the models. By trimming the feature set down to 500 principal components, we maintained a rich representation of the data while significantly enhancing computational efficiency.

In essence, the project not only demonstrated the viability of sophisticated feature extraction methods in medical image classification but also underscored the critical importance of model tuning and generalizability considerations, making the way for future advancements in this vital area of healthcare analytics.

7 References

[1]Jiancheng Yang, Rui Shi, Donglai Wei, Zequan Liu, Lin Zhao, Bilian Ke, Hanspeter Pfister & Bingbing Ni. edMNIST v2 - A large-scale lightweight benchmark for 2D and 3D biomedical image classification. 19 January 2023 https://www.nature.com/articles/s41597-022-01721-8

[2]Naeem A, Anees T (2024) DVFNet: A deep feature fusion-based model for the multiclassification of skin cancer utilizing dermoscopy images. PLOS ONE 19(3): e0297667. https://doi.org/10.1371/journal.pone.0297667

[3]Damian, F.A.; Moldovanu, S.; Dey, N.; Ashour, A.S.; Moraru, L. Feature Selection of Non-Dermoscopic Skin Lesion Images for Nevus and Melanoma Classification. Computation 2020, 8, 41. https://doi.org/10.3390/computation8020041