Projects in Data Science
Final Project Report Spring 2025

Andrea Cela - andce@itu.dk

Ákos Benjámin Döbörhegyi - akdo@itu.dk

Erik Aymerich Garcia - eray@itu.dk

Laura Cabral Costa Ascensão Tiago - lcab@itu.dk

Michal Grega - migre@itu.dk

`https://github.com/Erikzo03/2025-FYP-groupJellyfish.git`

Abstract

In this study, we developed a comprehensive system for classifying skin conditions using dermoscopic images, with a primary focus on detecting malignant lesions. The objective was to train machine learning models capable of distinguishing between benign and malignant skin lesions, primarily based on the well-established ABC features. We followed a flexible project framework provided by the course, which supported iterative development and experimentation. Our workflow included engineering both baseline and hair-related features, conducting classification experiments, and evaluating model performance. Additionally, we carried out a brief open-ended investigation to explore potential enhancements. However, our results indicated no significant performance improvement from incorporating hair-related features or additional features beyond the ABC set.

1 Introduction

The classification of skin lesions remains a significant challenge in medical image processing, particularly for the early detection of skin cancer. Early and accurate diagnosis is crucial, as skin cancer is among the most common malignancies worldwide, and its incidence continues to rise.[6] Advancements in machine learning have shown promise in improving diagnostic accuracy, especially when applied to dermoscopic images, which provide detailed visualization of skin lesions.

In this project, we developed a comprehensive framework for classifying skin lesions using dermoscopic images, with a primary focus on detecting malignant lesions. Unlike approaches that rely heavily on deep learning, our methodology emphasized the extraction and utilization of handcrafted features. This approach allowed us to explore the significance of various visual elements—such as color, shape, and texture—in differentiating between benign and malignant lesions. We employed classical machine learning classifiers, including Random Forest and logistic regression, to evaluate the effectiveness of these features.

Our workflow adhered to the structured project framework provided in the course template (2025-FYP-Final). The process involved several key steps: preprocessing raw images and corresponding masks, extracting relevant features, compiling an integrated dataset (dataset.csv), and conducting classification experiments. We performed two primary experiments—one utilizing baseline features and another incorporating additional features related to hair extraction and other lesion characteristics. Finally, we comprehensively evaluated our models and presented the findings in this report.

1.1 Purpose

We aim to determine whether adding extra features on top of the well-known ABC features in skin lesions makes a significant difference in the performance of the model. We do this by creating a main baseline with only the ABC features and an extended model with the extra features and comparing their performances, namely accuracy and recall. That leads us to the final question; Does adding extra features to the model yield significant changes in the performance? [8]

2 Dataset

The dataset for this project was obtained through the research PAD-UFES-20 dataset (Pacheco et al., 2020). It consists of 2,298 samples of six different types of skin lesions. Each sample consists of a clinical image and up to 22 clinical features including the patient's age, skin lesion location, Fitzpatrick skin type, and skin lesion diameter.[7] We selected 816 skin lesion images, each accompanied by segmentation masks and a filtered metadata CSV file. After downloading all necessary materials, we organized the files within a dedicated project directory to ensure streamlined access and reproducibility.

A copy of the metadata file, containing image filenames and corresponding labels, was placed directly in the main folder. Each team member maintained a local copy of the dataset on their own machines to enable independent development and testing. This approach also helped avoid challenges related to handling large image files on GitHub, where dermoscopic images and segmentation masks could complicate version control operations such as pushing or pulling updates.

Once everything was set up locally, we started by creating a Python script that matched each lesion with its corresponding masks based on the filenames. This contributed to the dataset organization and made it easier to proceed with the project.

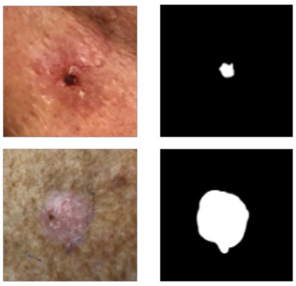


Figure 1: Skin lesion photo with the respective mask

To gain a deeper understanding of the dataset, we developed an R script that systematically examined and analyzed data by extracting key properties such as dimensions, color channels, and file formats. The script also parsed associated metadata to analyze class distribution across the dataset. Additionally, we generated visualizations by overlaying segmentation masks onto the corresponding images, providing intuitive insights into the structure and labeling quality of the dataset. [Fig.2]

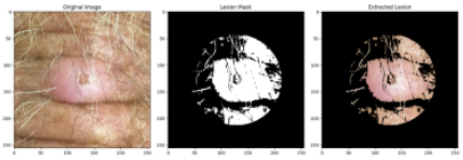


Figure 2: Example of overlaid segmentation

Most of the segmentation masks matched the lesion areas well, but a few appeared less accurate or noisy. [Fig. 3]

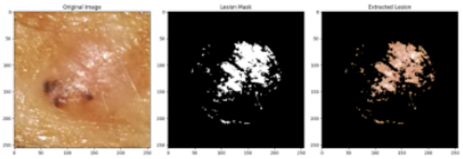


Figure 3: Example of overlaid segmentation

This early data exploration gave us valuable insight into the dataset's quality and helped us prepare for potential challenges in the feature extraction phase.

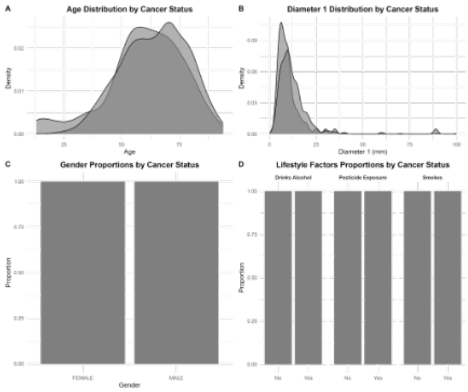


Figure 4: Distribution of metadata variables.

The top row shows the age and lesion diameter distributions. The bottom row includes categorical distributions for sex, alcohol consumption, pesticide exposure, and smoking. These visualizations provided a quick overview of potential biases or class imbalances in the dataset.

Description	Value
Number of images	863
Malignant cases	75%
Benign cases	25%

The proportion of malignant to benign was approximately 75/25. A balanced dataset would have been preferable to facilitate the model training.

3 Feature Extraction

As the first step of model creation, we need to extract data from the images that we received. This will represent the core of our model; the model will be built based on the extracted features data. For this process, we use Python modules such as cv2, sklearn, matplotlib, and others.

As a basis of image processing, the first step of extracting any feature is to implement K-means clustering to segment the image and therefore ignore the background of the image that does not tell us any valuable information. Following procedures differs from feature to feature.

For the main baseline of the project, we use the ABC features—asymmetry, border, and color. In the extended baseline, we introduce new features, namely Blue Veil and Contrast, and we implement a hair removal function, which processes the images before feature extraction.

3.1 Asymmetry

We extract the asymmetry features using three distinct methods.

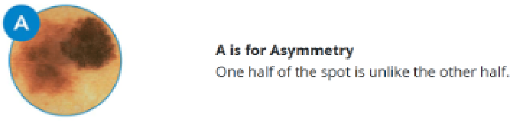


Figure 5: Example of Asymmetry [5]

3.1.1 Basic Mirror Asymmetry

Flip the masked images horizontally and vertically, and compute the pixel-wise absolute differences between the original and flipped masks. [8]

3.1.2 PCA-aligned rotational asymmetry

Flip the masked images horizontally and Perform principal component analysis (PCA) to align the lesion with its major axis. Compute the pixel-wise absolute difference as before.[1]

3.1.3 Boundary-weighted asymmetry

Isolate border pixels from the lesions and compute the Euclidean total distance transform from the boundary inward, emphasizing asymmetry near the border of the lesion. [1]

3.2 Border

Check if the contours and edges of the lesion are irregular or well-defined. We derive multiple subfeatures from the masked images.

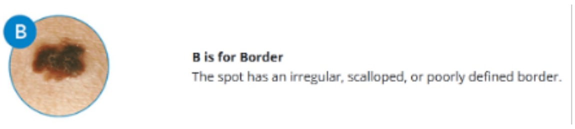

B
B is for Border
The spot has an irregular, scalloped, or poorly defined border.

Figure 6: Example of Border [5]

3.2.1 Sobel filter

A gradient-based method that looks for strong changes in the first derivative of an image.

First, a Gaussian filter is applied.

$$Gaussian_X = \left[\frac{1}{2\pi\sigma_x\sigma_y} \exp\left(-\frac{1}{2\sigma_x^2}\right) \quad \frac{1}{2\pi\sigma_x\sigma_y} \quad \frac{1}{2\pi\sigma_x\sigma_y} \exp\left(-\frac{1}{2\sigma_x^2}\right) \right]$$

The Sobel edge detector uses a pair of 3 × 3 convolution masks, one estimating the gradient in the x-direction and the other in the y-direction.[6][8]

G_x (x-direction)			G_y (y-direction)		
-1	0	$+1$	$+1$	$+2$	$+1$
-2	0	$+2$	0	0	0
-1	0	$+1$	-1	-2	-1

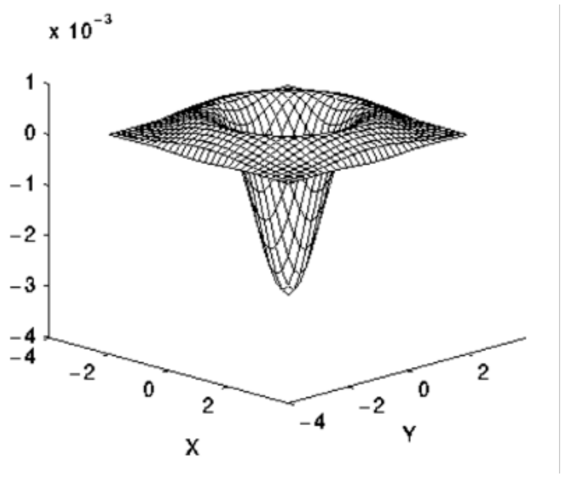
3.2.2 Laplacian of Gaussian

The Laplacian is a 2-D isotropic measure of the 2nd spatial derivative of an image.

The Laplacian L(x,y) of an image with pixel intensity values I(x,y) is given by

$$L(x, y) = \frac{\partial^2 I}{\partial x^2} + \frac{\partial^2 I}{\partial y^2}$$

A Gaussian filter is applied before as well to smooth the image because of the high sensitivity to noise of the second-degree partial derivatives. [2]



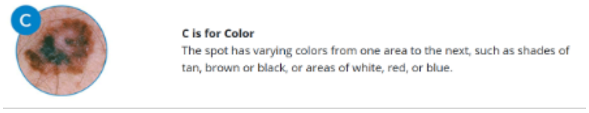
3.3 Color

For the color feature, we finally use the unprocessed images that we were given, unlike the A and B features, where we extract information using their masked counterparts—black and white images.

We extract 19 color features capturing the mean, variation, asymmetry, and ratios of color in both RGB and HSV(Hue, Saturation, Value) color spaces. These features help detect irregularities and patterns such as multi-colored lesions or uneven color distribution, which are often seen in malignant skin lesions. While not all high values are inherently dangerous, features like color variance, hue variation, and channel asymmetry are considered more predictive of malignancy.

Methods used in the segmentation of images for color extraction:

LAB color space conversion, SLIC superpixels, Circular central mask, and KMeans clustering. [5][8]



C is for Color

The spot has varying colors from one area to the next, such as shades of tan, brown or black, or areas of white, red, or blue.

Figure 7: Example of Color [5]

3.4 Blue veil

This feature is part of the extended baseline of our project and is used on the unprocessed images, similarly to the color feature. It also detects color, specifically the blue-white spectrum, which is prevalent in one of the skin lesion types, melanoma. The lesion pixels are converted to the HSV color space and specific thresholds are applied to each pixel for color, saturation, and brightness. Afterwards, we get a binary result of whether the blue veil was detected, together with the ratio as an output.

3.5 Contrast

Uses unprocessed images and converts them to grayscale. Computes the standard deviation of the grayscale values, which is the main metric of contrast in an image. Alongside the standard deviation, this function outputs the mean grayscale intensity as well. The contrast feature is also implemented only in the extended baseline of our project.

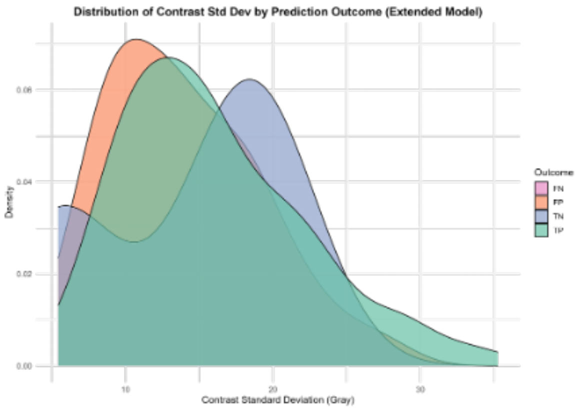


Figure 8: Distribution of contrast standard deviation by prediction outcome (TP-True Positive, FP-False Positive,TN-True Negative,FN-False Negative) for the extended model.

As shown, True Positives and False Positives tend to have higher contrast variability compared to True Negatives and False Negatives. This indicates that lesions with more intense grayscale variation (higher contrast) are more likely to be classified as malignant. It supports the relevance of contrast as a discriminative feature, although its influence may contribute to some misclassifications if texture alone is misleading.

3.6 Hair removal

To detect hair, we use techniques like blackhat filtering and thresholding, which allow us to estimate how much of the lesion is covered by hair. For example, we calculated the percentage of blackhat pixels inside the lesion area as one of our features. These features were designed to help the classifier handle cases where hair might obscure important lesion information.

The ratio is calculated as the proportion of image pixels covered by the detected hair mask. A ratio of 1 would mean the entire image is covered by the hair mask, indicating a lot of hair.

Once all features are ready, we combine them in the main baseline (ABC features) and extended baseline (ABC + Blue Veil, contrast + hair removal) scripts. These scripts loop through each image listed in the metadata file, load both the image and its mask, run all feature extraction functions, and store the results in a list. Ultimately, this list is converted into a Pandas DataFrame. This file now holds all the numerical or categorical feature values that represent the main input for all our later classification experiments.

4 Classification model

We began this project by approaching the problem as a non-binary classification task, aiming to distinguish between multiple types of skin lesions. To determine the best model, we evaluated four different classification algorithms: K-Nearest Neighbors (KNN)[10], Random Forest, Logistic Regression, and Decision Tree. Model selection was based primarily on accuracy, with recall also taken into consideration.

Initially, Logistic Regression showed the highest accuracy among the tested models. However, further analysis revealed that it consistently predicted basal cell carcinoma (BCC) with a recall of 1.0 for that class. Although this indicates perfect identification of BCC cases, it also reflects a strong bias toward this class, likely compromising the classification of other lesion types.

To address this issue, we transitioned to a binary classification framework, distinguishing only between cancerous and non-cancerous lesions. In this setting, Random Forest was selected as the preferred model due to its ability to generate more stable and generalizable predictions. This approach improved the model's discriminative power and mitigated bias, leading to more balanced and clinically relevant classification results.

At first, discriminative ability improved, and the bias toward a single disease class was reduced. In the context of medical applications, Random Forest is especially valuable due to its high interpretability, resilience to noisy data, and strong generalization capabilities. These traits make it a practical and trustworthy model in clinical decision support systems, where both accuracy and transparency are critical.

Reference	Focus Area	Key Insights
Predicting Essential Medicine Demand with Random Forests	Medicine demand prediction	RF models achieve high accuracy and optimize supply chain management
Federated Random Forests for Healthcare Applications	Federated learning in healthcare	FRF models improve generalizability and maintain privacy
Practical Applications of Random Forests in Biomedicine	Biomedical research	RFs excel in disease prediction, biomarker discovery, and image analysis
Random Forests in Precision Medicine Research	Precision medicine	RFs support classification, prognosis, and patient stratification
Random Forests for Medical Data Classification	Medical data classification	RFs identify important features and enhance diagnostic accuracy

While researching suitable models for our skin lesion classification task, we explored various ensemble methods and found a useful example dataset and code from an online source. The code uses synthetic data to compare how different classifiers perform, including Decision Tree, Random Forest, Extra Trees and AdaBoost.

The image we included in this section visually compares different classifiers. As seen in the Random Forest plots, the decision boundaries are smoother and more generalizable compared to a single Decision Tree. This makes Random Forest a great fit for our medical project, where the data can be noisy or inconsistent. It offers both accuracy and reliability, which are essential when working with healthcare-related data

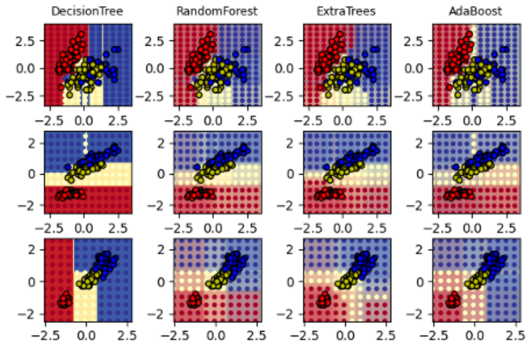


Figure 9: Classifiers on feature subsets [3]

In conclusion, using Random Forest in our project helped us build a model that can effectively handle a variety of features and still provide consistent results, something that's especially important in real-world medical applications, where decisions must be both trustworthy and interpretable.

5 Classification & Experiments

Baseline vs Extended Baseline Models

For the baseline experiment, the script main_baseline.py orchestrates a comprehensive image analysis and classification pipeline. It begins by extracting three distinct sets of features from input images: Asymmetry (Feature A), Border characteristics (Feature B), and Color distributions (Feature C), using custom utility functions. These extracted features are then merged with diagnostic labels loaded from a metadata CSV file, creating a consolidated feature dataset, which is subsequently saved. This dataset undergoes preprocessing, including handling missing values via imputation and one-hot encoding categorical features.

For model evaluation, the script employs K-fold Stratified Cross-Validation. Within each fold, a Random Forest Classifier is trained on an inner training portion (with an inner validation step) and then rigorously evaluated on the designated test portion of that fold. Detailed performance metrics, such as accuracy, precision, recall, F1-score, and confusion matrices, are calculated for each fold, aggregated, and then saved to various output CSV files, along with all individual test predictions.

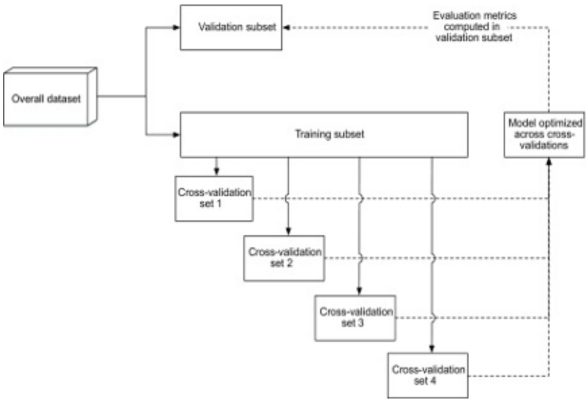


Figure 10: Cross Validation [4]

Validation

The script implements an inner validation step within each K-fold. The development data of a fold (`x_dev_fold`) is split into `x_train_inner` and `x_val_inner` using `train_test_split`. The Random Forest model is then trained solely on `x_train_inner`:

```
    model_fold.fit(x_train_inner,
    y_train_inner)
```

Immediately after, its performance is validated on the unseen `x_val_inner` before testing on the fold's main test set:

```
    val_acc = accuracy_score(y_val_inner,
    model_fold.predict(x_val_inner))
```

This `val_acc_inner_fold` provides an intermediate performance check.

Then we created `main_extended.py`, which ran a similar experiment but included the hair extraction together with 2 extra features—Blue-Veil and Contrast, along with the baseline ones. We used the same data split as in the baseline to ensure a fair comparison. The model was trained and evaluated again, and we compared the metrics to see if hair features made any difference.[9]

6 Results

First, we tested 4 different classification models to determine which one yields the best results. Afterwards, we implemented K-fold cross-validation with 5 folds on both of the models using the previously chosen classification model. The tests were done on our dataset of 863 images that were divided into train/validate/test splits using stratification. The results are shown below.[9]

$$\text{Recall}_{\text{non-cancer}} = \frac{\text{TP}_{\text{non-cancer}}}{\text{TP}_{\text{non-cancer}} + \text{FN}_{\text{non-cancer}}}$$

Formula for the non-cancer recall

$$\text{Accuracy} = \frac{\text{Correct Predictions}}{\text{Total Predictions}} = \frac{TP + TN}{TP + TN + FP + FN}$$

Formula for the overall accuracy score.

Class	Metric	Main Baseline	Extended Baseline
Non-cancer	Precision	0.6350 ± 0.20	0.36 – 1.00 (unstable)
	Recall	0.0934 ± 0.02	Even worse (often 0)
	F1-score	0.1613 ± 0.04	∼ 0.00–0.17
Cancer	Precision	0.7660 ± 0.01	∼ 0.75
	Recall	0.9784 ± 0.02	∼ 0.98
	F1-score	0.8592 ± 0.01	∼ 0.85

Wilcoxon's signed-rank test

The Wilcoxon signed-rank test was applied to determine whether there is a statistically significant difference in performance between the two models, under the null hypothesis that no such difference exists. The resulting p-values did not indicate statistical significance either in the model accuracy or in recall, suggesting insufficient evidence to reject the null hypothesis. This implies that any observed performance differences between the models may be due to random variation rather than a systematic effect.

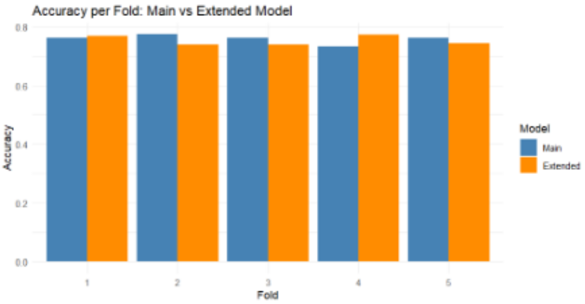


Figure 11: Accuracy on the Main Model vs Extended Model

P-value: 0.8125

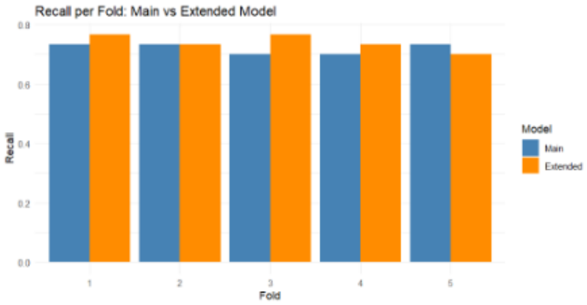


Figure 12: Recall on the Main Model vs Extended Model

P-value: 0.2693

7 Discussion

Through this project, we learned that carefully selected handcrafted features can be surprisingly effective for skin lesion classification. Shape-related features, such as symmetry and border irregularity, often provided valuable signals to the model. Color-based features also contributed, though their reliability was sometimes influenced by external factors like lighting variations or the presence of hair in the image.

Why are the results from the extended model not better than those from the main model?

There are 2 main possible reasons for this. The first and more likely one is that the inpainting features from hair extraction are subtly damaging the skin lesion features that are very important for feature extraction. The other possibility is that the new features, Blue-Veil and Contrast, are adding more noise than signal.

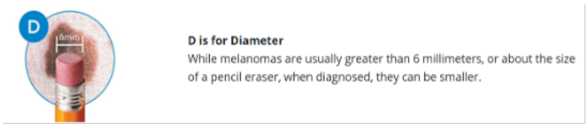
We experimented with adding hair detection features, which did offer a slight performance boost. However, the impact wasn't dramatic—possibly because the baseline features already captured much of the relevant information or because our method of detecting hair was not entirely robust. Nevertheless, this reinforced the idea that thoughtful feature engineering can still make a meaningful difference in machine learning pipelines.

There were also some challenges. Some segmentation masks were not very accurate, which directly affected the features we extracted, especially those related to shape and color symmetry. Additionally, the dataset size was limited, so we had to be cautious of overfitting and make sure to use reproducible train-test splits to maintain reliability.

When comparing our model to real-world dermatology practice, it's important to acknowledge certain limitations. Dermatologists often consider time-dependent factors such as lesion growth, diameter changes, and depth progression, which are not captured in static images. These dynamic features can provide critical diagnostic insights, especially for identifying malignant transformations. While our model analyzes a snapshot in time, real skin lesions can evolve, and tracking that evolution is key in clinical settings.

Possible Improvements:

7.1 Diameter: Limitations Due to Lack of Spatial Calibration



D is for Diameter

While melanomas are usually greater than 6 millimeters, or about the size of a pencil eraser, when diagnosed, they can be smaller.

Figure 13: Example of Diameter [5]

In the ABCD diagnostic framework, D represents diameter, which is a crucial parameter in clinical skin cancer evaluation. However, in our project, we were unable to incorporate this feature due to the absence of spatial calibration information in the dermoscopic images. Without a known pixel-to-millimeter ratio or reference scale, it was not possible to accurately determine the physical size of lesions. Measurements in pixels alone lack diagnostic relevance unless they can be translated into real-world units.

ABCD RULE

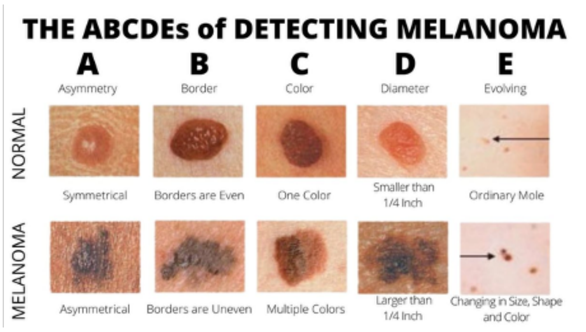


Figure 14: Correlations of all features with the benign/malignant label.

As an alternative, we concentrated on extracting morphological and visual features such as asymmetry, border irregularity, color variation, contrast, and the presence of blue veil patterns, features that serve as strong indicators of lesion abnormality.

Had we had access to precise scaling data, integrating diameter as a feature would have enhanced the clinical validity of our model and potentially improved its diagnostic performance.

However, given the critical importance of precision in medical contexts, we opted not to include diameter to maintain the integrity and reliability of our results.[5]

7.2 Evolving: Importance of Temporal Progression

Another key aspect we did not incorporate was the temporal evolution or progression of skin lesions, which plays a significant role in clinical diagnosis. In medical practice, dermatologists routinely monitor how a lesion changes over time—whether it increases in size, alters in pigmentation, or becomes more structurally irregular. However, our dataset provided only a single static image per lesion, limiting our analysis to a snapshot rather than a longitudinal perspective.

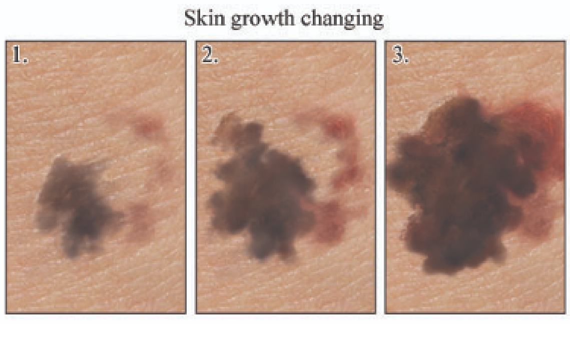
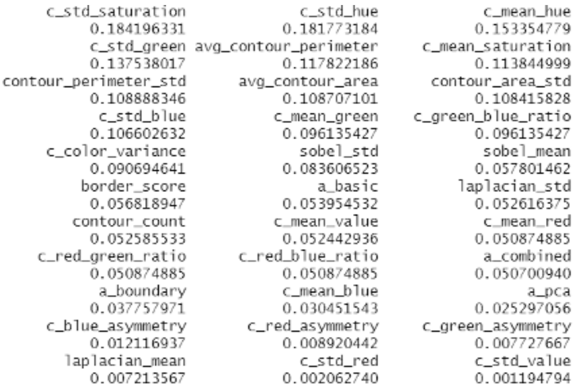


Figure 15: The E rule (Evolving)

Incorporating evolution would require time-series data, ideally consisting of multiple images of the same lesion captured at different time intervals. This would allow the model to detect dynamic patterns of morphological change, enhancing diagnostic accuracy and better aligning with real-world clinical assessment methods. Investigating lesion progression over time is a critical component of melanoma detection, and with the appropriate dataset, it is an area we would strongly consider for future development.

7.3 Statistical analysis of features

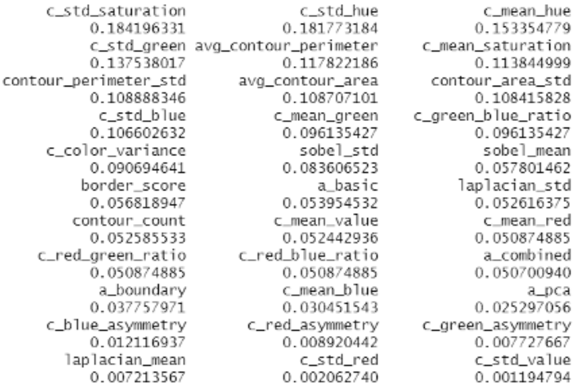
Numerous features are characteristic of certain types of lesions. However, when developing a classification model, it is crucial to avoid overfitting. Ideally, we aim for features with a strong correlation with the dependent variable. Below, we present a table showing the Pearson correlation coefficients between each feature and the diagnosis label (our dependent variable). In future iterations, we would consider removing features with very low correlation, as they contribute little to predictive performance and may introduce noise.



```
     c_std_saturation             c_std_hue           c_mean_hue
          0.184196331           0.181773184          0.153354779
          c_std_green avg_contour_perimeter    c_mean_saturation
          0.137538017           0.117822186          0.113844999
contour_perimeter_std      avg_contour_area     contour_area_std
          0.108888346           0.108707101          0.108415828
           c_std_blue          c_mean_green   c_green_blue_ratio
          0.106602632           0.096135427          0.096135427
     c_color_variance             sobel_std           sobel_mean
          0.090694641           0.083606523          0.057801462
         border_score               a_basic        laplacian_std
          0.056818947           0.053954532          0.052616375
        contour_count          c_mean_value           c_mean_red
          0.052585533           0.052442936          0.050874885
    c_red_green_ratio      c_red_blue_ratio           a_combined
          0.050874885           0.050874885          0.050700940
           a_boundary           c_mean_blue                a_pca
          0.037757971           0.030451543          0.025297056
     c_blue_asymmetry       c_red_asymmetry    c_green_asymmetry
          0.012116937           0.008920442          0.007727667
       laplacian_mean             c_std_red          c_std_value
          0.007213567           0.002062740          0.001194794
```

Figure 16: Correlations of all features with the benign/malignant label.

Another thing to consider when filtering features is multicollinearity. In our project, we use many features that might not be necessary. As an example, in the figure below, we see the pairwise Pearson correlation of features that have a substantially high multicollinearity.



```
     c_std_saturation             c_std_hue           c_mean_hue
          0.184196331           0.181773184          0.153354779
          c_std_green avg_contour_perimeter    c_mean_saturation
          0.137538017           0.117822186          0.113844999
contour_perimeter_std      avg_contour_area     contour_area_std
          0.108888346           0.108707101          0.108415828
           c_std_blue          c_mean_green   c_green_blue_ratio
          0.106602632           0.096135427          0.096135427
     c_color_variance             sobel_std           sobel_mean
          0.090694641           0.083606523          0.057801462
         border_score               a_basic        laplacian_std
          0.056818947           0.053954532          0.052616375
        contour_count          c_mean_value           c_mean_red
          0.052585533           0.052442936          0.050874885
    c_red_green_ratio      c_red_blue_ratio           a_combined
          0.050874885           0.050874885          0.050700940
           a_boundary           c_mean_blue                a_pca
          0.037757971           0.030451543          0.025297056
     c_blue_asymmetry       c_red_asymmetry    c_green_asymmetry
          0.012116937           0.008920442          0.007727667
       laplacian_mean             c_std_red          c_std_value
          0.007213567           0.002062740          0.001194794
```

Figure 17: Multicollinearity between features

It could therefore be beneficial to remove one feature of each feature pair in the figure above.

8 Conclusion

This project aimed to assess whether adding an extended feature set could significantly improve a model's ability to classify skin lesions as cancerous or non-cancerous. Due to the heavily skewed class distribution in the original multi-class dataset, binary classification was chosen to focus the task. Even so, the final dataset remained imbalanced, with roughly 75% cancer cases.

Two models were developed—one with a base set of features, and one with additional variables, including those derived from a hair-removal preprocessing step. Evaluation using k-fold cross-validation and metrics like precision, recall, and F1-score showed strong detection of cancer cases (low false negatives), but poor recall for non-cancerous ones (high false positives), likely a result of the class imbalance.

The Wilcoxon signed-rank test was applied to assess whether the extended model offered significant improvements. The resulting p-values showed no statistically significant difference, suggesting that the added features—including those potentially distorted by hair inpainting—did not meaningfully enhance performance.

Overall, while feature expansion was a valid direction, future efforts may benefit more from balancing the dataset or trying alternative modeling approaches than from adding further features.

9 References

1. Celebi, M. E., Kingravi, H. A., & Uddin, B. (2007). A methodological approach to the classification of dermoscopy images. *Computerized Medical Imaging and Graphics, 31*(6), 362–373. https://doi.org/10.1016/j.compmedimag.2007.01.003

2. Fisher, R., Perkins, S., Walker, A., & Wolfart, E. (2003). Spatial Filters - Laplacian/Laplacian of Gaussian. The University of Edinburgh. https://homepages.inf.ed.ac.uk/rbf/HIPR2/log.htm

3. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., & Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research, 12*, 2825–2830. https://scikit-learn.org/stable/modules/ensemble.html#random-forests

4. Basu, S., Kumbier, K., Brown, J. B., & Yu, B. (2020). A Tutorial for Medical and Public Health Researchers on Machine Learning Methods. *American Journal of Epidemiology, 189*(11), 1564–1575. https://pmc.ncbi.nlm.nih.gov/articles/PMC7138444/

5. American Academy of Dermatology. (n.d.). Melanoma: Signs and symptoms. https://www.aad.org/public/diseases/skin-cancer/types/common/melanoma/symptoms

6. Leiter, U., Keim, U., & Garbe, C. (2020). Epidemiology of Skin Cancer: Update 2019. https://link.springer.com/chapter/10.1007/978-3-030-46227-7_6

7. Pacheco, A. G. C., et al. PAD-UFES-20: a skin lesion dataset composed of patient data and clinical images collected from smartphones. https://data.mendeley.com/datasets/zr7vgbcyr2/1

8. DermNet NZ. (2008). Three-point checklist for dermoscopy. https://dermnetnz.org/cme/dermoscopy-course/three-point-checklist

9. Chauhan, N. S. (2023, November 29). K-Fold Cross-Validation Tutorial. DataCamp. Retrieved May 16, 2024, from https://www.datacamp.com/tutorial/k-fold-cross-validation

10. K-Nearest Neighbor (KNN) Algorithm. https://www.geeksforgeeks.org/k-nearest-neighbours/