Skin Lesion Prediction

First Year Project

IT UNIVERSITY OF CPH

BSc in Data Science

Group 6

Costel Gutu
cogu@itu.dk

Inzamam Ul Haque
inzh@itu.dk

Matthew Jorge Paramo
mjpa@itu.dk

Oleksandr Adamov
olea@itu.dk

Vladislav Konjusenko
vlko@itu.dk

BSFIYEP1KU

GitHub: https://github.com/matthewparamo/fyp2023

June 2, 2023

Introduction

Skin cancer is a significant public health concern worldwide, with melanoma being the most aggressive and potentially fatal form. Early detection and accurate diagnosis of skin lesions are crucial for effective treatment and improved patient outcomes. According to the World Health Organization (WHO), the incidence of skin cancer has been steadily increasing over the past few decades, making it a pressing issue in dermatology [1]. In recent years, there has been growing interest in developing automated systems that can analyze images of skin lesions and assist in diagnosis. This project aims to contribute to this field by exploring the measurement of image features and utilizing machine learning techniques to predict the diagnosis of skin lesions.

The motivation for this project stems from the need for reliable and efficient methods to aid dermatologists in diagnosing skin lesions. While visual inspection by dermatologists remains the gold standard, it is a subjective and time-consuming process. Furthermore, the shortage of dermatologists in certain regions hinders timely diagnoses, leading to potential delays in treatment. Developing an automated system that can accurately analyze skin lesion images has the potential to assist dermatologists in their decision-making process, improve diagnostic accuracy, and facilitate early intervention when necessary.

Extensive research has been conducted in the field of computer-aided diagnosis of skin lesions using image analysis and machine learning techniques. Studies have focused on various aspects, including feature extraction, classification algorithms, and ensemble methods. Feature extraction techniques have explored color variations, texture patterns, shape characteristics, and asymmetry in skin lesion images [2][3]. Machine learning classifiers such as Support Vector Machines (SVM), Random Forests, and Convolutional Neural Networks (CNN) have been employed to classify skin lesions based on these features [4][5].

While existing research has achieved promising results, there are still several open research questions that this project aims to address. Specifically, this project focuses on the measurement of image features and their utilization in machine learning-based diagnosis. Some of the key research questions include:

1. Which image features are most informative for accurately predicting the diagnosis of skin lesions?
2. How can these image features be effectively measured and extracted from skin lesion images?
3. Which machine learning classifiers demonstrate the highest predictive performance for skin lesion diagnosis?

Addressing these research questions will contribute to the development of an automated system that can assist dermatologists in diagnosing skin lesions accurately and efficiently. The findings and insights gained from this research can potentially lead to improved patient outcomes, reduced diagnostic delays, and better utilization of dermatologists' expertise.

In the following sections of this report, we will describe the dataset used, present the methodology employed for feature measurement and classification, discuss the experimental results, and provide a comprehensive analysis of the findings. The ultimate goal is to provide valuable insights and recommendations for the further advancement of automated skin lesion diagnosis systems.

Data description and pre-processing

For this project we used PAD-UFES-20 dataset[6]. The PAD-UFES-20 dataset was collected along with the Dermatological and Surgical Assistance Program at the Federal University of Espírito Santo (UFES-Brazil), which is a nonprofit program that provides free skin lesion treatment, in particular, to low-income people who cannot afford private treatment.

The dataset includes patient-specific information, such as age, gender, and clinical details, which provide contextual information about the individuals and their skin lesions. The dataset 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(Figure 1), skin lesion location(Figure 2), Fitzpatrick skin type, and skin lesion diameter. The metadata associated with each skin lesion is composed of up to 26 features. All features were available in a CSV document in which each line represents a skin lesion and each column a metadata feature. In total, there are 1,373 patients, 1,641 skin lesions, and 2,298 images present in the dataset. Each image/sample has a reference to the patient and the skin lesion in the metadata. This information can be used to explore potential correlations between patient characteristics and the diagnosis of skin lesions.

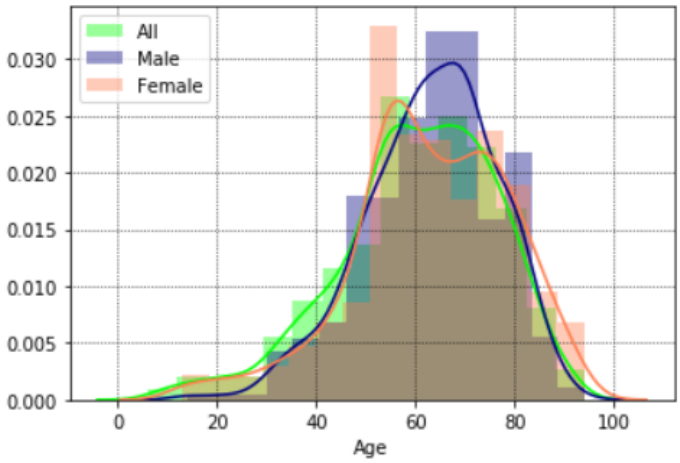

Figure 1: Age distribution in the dataset

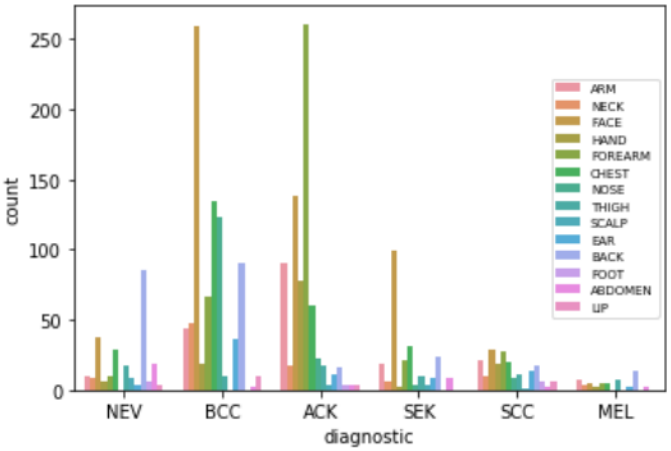

Figure 2: Anatomical region

The dataset comprises a diverse range of skin lesions, including various types of skin cancer (e.g., melanoma, basal cell carcinoma) as well as benign lesions. This diversity ensures that the dataset represents a broad spectrum of dermatological conditions, enabling the development of robust and generalizable models. The clinical images in the PAD-UFES-20 dataset are of high quality, ensuring accurate and detailed representation of the skin lesions. This quality is crucial for extracting meaningful features and patterns that can aid in diagnosis.

In summary, the PAD-UFES-20 dataset is a rich resource that combines patient data and clinical images collected from smartphones. It offers a diverse collection of skin lesions, facilitating research and development in the field of automated skin lesion diagnosis. The dataset's composition and quality make it a valuable asset for exploring image features, developing machine learning models, and advancing the field of computer-aided diagnosis of skin lesions.

To determine the health status of each image, we relied on the provided labels. Images labeled as "NEV" were considered healthy, as these typically represent benign skin lesions. On the other hand, all other images, which were not labeled as "NEV," were categorized as unhealthy, encompassing various types of skin abnormalities and potentially malignant lesions. To ensure a balanced representation of both healthy and unhealthy images within our dataset, we employed a stratified random sampling technique. This approach allowed us to create a subset of images that accurately reflects an equal distribution between the two classes. Specifically, we carefully selected a total of 140 healthy images and 140 unhealthy images for segmentation, model training, and testing purposes. By adopting this balanced sampling strategy, we aimed to mitigate any potential bias towards either category, thereby ensuring fair and unbiased evaluation of our segmentation model.

Following the random selection of 280 images encompassing both healthy and unhealthy samples, a meticulous evaluation of the dataset was conducted to identify potential irregularities. Particular attention was given to the identification of blurry images that might undermine subsequent analyses. Gratifyingly, no instances of blurry images were encountered during this assessment. Moreover, a comprehensive examination aimed at identifying any duplicate images was carried out, and it is noteworthy that no duplicates were detected. Additionally, a diligent scrutiny for missing values was undertaken, and it is with satisfaction that we report the absence of any missing values within the dataset. Furthermore, a rigorous assessment was conducted to ascertain the consistency and accuracy of image labeling, yielding the affirmative finding that no inconsistencies or inaccuracies were detected.

Image Segmentation

Segmentation plays a crucial role in image recognition systems as it identifies and extracts the specific objects we are interested in for further analysis or recognition purposes. By classifying the pixels in an image, segmentation techniques are employed to separate the desired object from the background, enabling detailed analysis. For instance, image segmentation can effectively isolate a tumor, cancerous region, or an obstruction in blood flow, facilitating their examination.[7]

In our image segmentation process, the first step involves resizing the images to a standardized dimension, typically 200x200 pixels. This resizing step holds significant importance for machine learning applications. By ensuring that all images have the same size, we create a consistent input format for our models. This standardization simplifies the computational complexity and allows for efficient processing of the images. Moreover, image resizing reduces the overall dimensionality of the data, which can be beneficial in terms of reducing memory usage and computation time

[8]. Additionally, resizing helps in maintaining the aspect ratio of the objects within the images, preserving their relative proportions and preventing distortion. By resizing the images, we establish a uniform foundation for subsequent analysis and segmentation tasks, enabling our machine learning algorithms to learn effectively and accurately interpret the visual information contained within the images. [9]

The grayscale version of the image is obtained by setting the as_gray parameter to True when reading the image. Converting images to grayscale simplifies the data by removing color information, reducing dimensionality, and eliminating potential variations caused by color differences. This preprocessing step improves the efficiency and accuracy of image segmentation and subsequent analysis tasks.[10]

Gaussian filtering is a technique used in image processing for smoothing and reducing noise in images. It involves applying a Gaussian filter, which is a weighted average, to each pixel in the image. The purpose of Gaussian filtering in image segmentation is to enhance the quality of the image by reducing noise and creating a more homogeneous appearance. By blurring the image slightly, Gaussian filtering helps to suppress small details and fluctuations, making it easier to distinguish and identify larger objects or regions during the segmentation process. This filtering technique helps improve the accuracy and robustness of image segmentation algorithms by providing cleaner and more reliable image data for analysis.[11]

The image is then thresholded using the threshold_multiotsu function from the *skimage* library, which automatically determines the optimal threshold values based on the specified number of classes. The resulting thresholds are used to classify the pixels into different regions. Multi-otsu thresholding is used in image segmentation to separate an image into multiple regions based on different intensity thresholds. It enables the identification and segmentation of multiple objects or regions with distinct intensity levels, improving the accuracy and detail of the segmentation process. This technique enhances the effectiveness of image analysis by providing finer segmentation and enabling a more comprehensive understanding of complex images.

A circular shape is defined using the np.linspace function, and an initial snake contour is created using segmentation.active_contour. The snake is iteratively deformed to fit the edges of the segmented region. The coordinates inside the polygon defined by the snake are extracted using polygon, and a binary mask is created.

The mask is applied to the grayscale image to obtain the segmented lesion using element-wise multiplication. Dilation is performed on the binary mask using a disk-shaped structuring element with a radius of 4 pixels to expand the segmented region.

Finally, the resulting images at various stages of the segmentation process are plotted

using Matplotlib, including the grayscale image, the segmented output, the binary mask, the dilated mask, the regions after thresholding, and the final segmented image.

In summary, our image segmentation method combines techniques such as resizing, grayscale conversion, Gaussian filtering, thresholding, active contour modeling, and binary mask manipulation to accurately extract a specific region of interest from an input image. The resulting segmented image can be used for further analysis, classification, or feature extraction tasks in medical imaging or other image processing applications.

Feature Extraction

Feature extraction is a crucial step in training machine learning models, particularly for tasks like image classification. It involves transforming raw input data, such as images, into a set of representative features that capture relevant information.[12] The goal of feature extraction is twofold. Firstly, it reduces the dimensionality of the data, making it more manageable and computationally efficient. High-dimensional raw data, such as pixel values in an image, can lead to the curse of dimensionality and hinder model performance. By extracting a smaller set of meaningful features, we can overcome this challenge and improve the efficiency of our models.[13]

In order to extract the color features of the already segmented images We developed a Python code to extract color features from an image utilizing the SLIC (Simple Linear Iterative Clustering) algorithm. A foreground mask is created by checking if all channels are not equal to [0, 0, 0], which identifies the non-black pixels in the image. By considering this condition, we can distinguish and isolate the pixels that contain color information and exclude those that correspond to the absence of color or black regions. The SLIC algorithm is applied to the modified image multiplied by the foreground mask. This segments the image into regions based on color similarity, with the specified parameters for the number of segments and compactness.

The SLIC (Simple Linear Iterative Clustering) algorithm is a popular image segmentation technique. It combines aspects of both superpixels and clustering algorithms to partition an image into regions or segments based on color similarity.The SLIC algorithm works by first initializing a set of cluster centers or seeds evenly distributed across the image. These seeds serve as representatives of the segments. Next, for each seed, a local search is performed within a predefined region around the seed to find the pixel with the lowest gradient or color difference. This pixel is then assigned to the corresponding segment. In an iterative process, the cluster centers are updated based on the average color and position of the pixels assigned to each segment. The process is repeated until convergence, ensuring that the segments

align with the color boundaries and that the cluster centers stabilize. [14]

The SLIC algorithm offers a balance between efficiency and accuracy, as it provides compact and visually meaningful segments while being computationally efficient. It is commonly used in computer vision tasks such as image analysis, object recognition, and image-based rendering.[14]

Choosing asymmetry as a feature for extraction in machine learning holds significant potential for enhancing the performance and interpretability of models across various domains. Asymmetry serves as an essential visual characteristic that can convey valuable information about underlying patterns and structures within images.

We divided a method to assess image asymmetry by calculating the correlation between vertical and horizontal projections of a masked image. The process involves several steps: normalization of the histograms to represent probability distributions, computation of the Bhattacharyya coefficient as a measure of similarity between the distributions, and the subsequent calculation of the Bhattacharyya distance as an indicator of image asymmetry.

By employing this method, we aim to evaluate the degree of symmetry present in an image. Symmetry is a fundamental visual characteristic and can provide valuable insights in various fields, including machine learning. In the context of machine learning, asymmetry analysis serves as a feature extraction technique that can aid in understanding and capturing significant patterns within images. [13]

In many real-world scenarios, asymmetry can be indicative of underlying characteristics or conditions. For instance, in medical imaging, asymmetry in anatomical structures or lesions can be a key diagnostic criterion. By quantifying and analyzing asymmetry, machine learning models can learn to recognize and classify patterns associated with specific conditions, assisting in accurate disease detection and diagnosis.[19]

K-Fold Cross Validation

K-fold cross-validation is a technique used in machine learning to evaluate and validate models. Common choices for the value of k are 5 or 10, but the value can vary depending on the dataset size and specific requirements. It involves dividing the dataset into k equal-sized subsets or folds, where the model is trained and evaluated k times using a different fold as the validation set each time. By averaging the performance metrics obtained from each iteration, it provides a strong estimation of the model's performance and helps assess its generalisation ability. We use this technique for skin lesion classification in all our models to soften the impact of data variability and obtain a more reliable evaluation of the models effectiveness. We set

the k-fold value for 10. [22]

Model description and selection

After extracting the features from our dataset, we selected several models to perform classification on our project. The models we chose include K-Nearest Neighbors (KNN), Logistic Regression, Nearest Centroid (Closest Mean) and Random Forest. Each of these models has its own characteristics and approaches to making predictions. [15, 17, 18, 19]

In our project, we leveraged the scikit-learn library in Python, which offers a comprehensive set of tools and utilities for machine learning tasks. Scikit-learn provides efficient implementations of various algorithms, including the ones mentioned above, along with functionalities for data preprocessing, model evaluation, and hyperparameter tuning. [16]

KNN

K-Nearest Neighbors (KNN) is a universal classification algorithm that predicts the class of a data point by identifying its k nearest neighbours in the training set. It assigns labels to new instances based on their similarity to previously labeled instances. The predicted class is determined through a majority vote among its neighbours. [17]

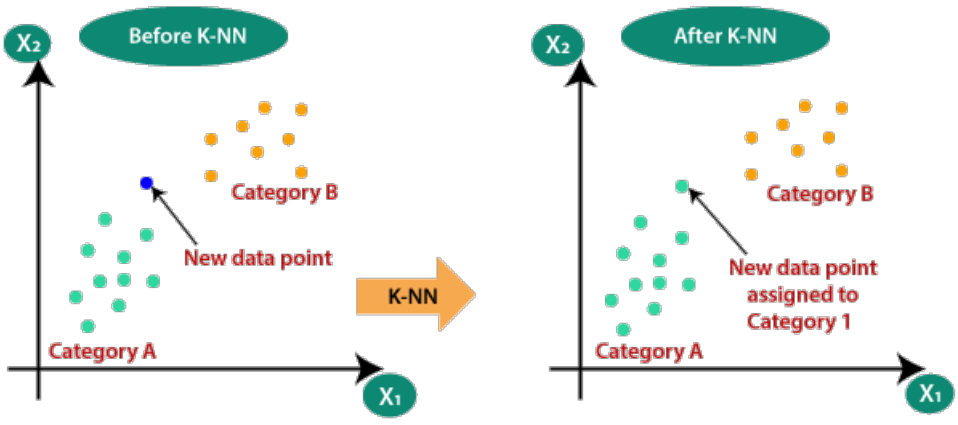

Figure 3: KNN example

Then there are four main steps involved in the KNN algorithm:

1. The first step in KNN is to determine the value of the hyperparameter of k, which represents the number of nearest neighbours to consider when making predictions. The selection of k depends on the dataset and problem at hand.

A smaller value of k can lead to more flexible decision boundaries, but may also be more prone to noise, while a larger value of k can provide smoother decision boundaries but might struggle with capturing local patterns. [17]

2. Once the value of k is determined, the algorithm computes the distances between the new data point and all the instances in the training set. We can do that with the Minkowski distance formula. [17]

$$D(X,Y) = \left(\sum_{i=1}^{n} |x_i - y_i|^p\right)^{\frac{1}{p}},$$

where X and Y are two points,

$$X = (x_1, x_2, ..., x_n) \text{ and } Y = (y_1, y_2, ..., y_n) \in \mathbb{R}^n.$$

The case where p = 1 is equivalent to the Manhattan distance formula and the case where p = 2 is equivalent to the Euclidean distance formula. Although p can be any real value, it is typically set to a value between 1 and 2. For values of p less than 1, the formula above does not define a valid distance metric since the triangle inequality is not satisfied. [17]

3. After calculating distances, the algorithm selects the k instances with the shortest distances to the new data point. These instances are considered the nearest neighbours. [17]

4. Finally, the algorithm determines the class label for the new data point by majority voting among its k nearest neighbours. In other words, it assigns the class label that appears most frequently among the neighbours. [17]

KNN has several advantages. It is simple to understand and implement. It can handle both classification and regression tasks. It can capture complex decision boundaries and work well with nonlinear data. However, its main drawback is the computational cost, as it requires comparing the new instance with all training instances during prediction. It is sensitive to the choice of distance metric and the value of k. It does not provide explicit insights into the underlying relationship between features and the outcome. [17]

In our KNN model we receive as an output predicted probabilities. These probabilities represent the estimated likelihood of each class and can be useful for assessing the model's confidence in its predictions. The predicted probabilities are returned as an array or a matrix, where each row corresponds to an instance and each column represents the probability of a specific class. After that we use this probabilitites in model selection process.

Logistic Regression

Logistic Regression is a widely used classification algorithm that models the relationship between features and a binary or categorical outcome. It is particularly valuable when dealing with datasets where the outcome is binary, such as classifying instances as healthy (1) or unhealthy (0). [15]

The goal of Logistic Regression is to estimate the probabilities of different classes based on the input features. In the case of a binary classification problem, the algorithm calculates the probability of an instance belonging to the positive class (healthy) or the negative class (unhealthy). By comparing these probabilities, Logistic Regression assigns the class with the highest probability as the predicted class for each instance. We can compute Logistic Regression with the next formula. [15]

$$f(x) = \frac{L}{1 + e^{-k(x-x_0)}} \tag{1}$$

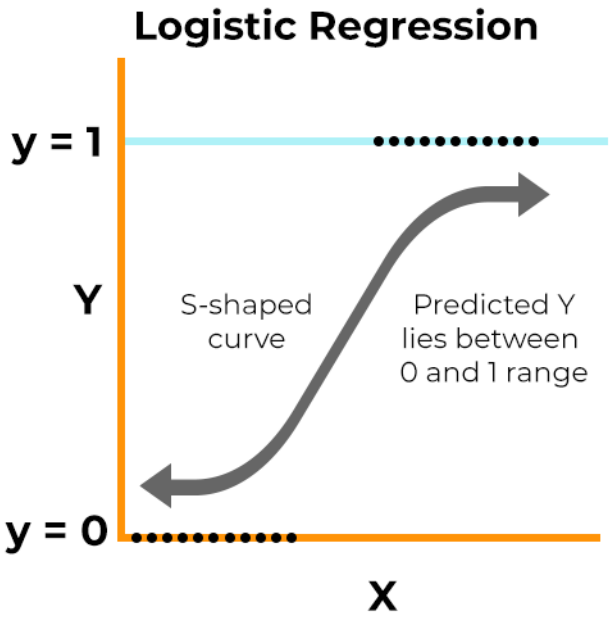

Figure 4: Logistic Regression Example

One of the advantages of Logistic Regression is its interpretability. The coefficients associated with each feature provide insights into the impact of the features on the predicted outcome. The magnitude of the coefficients reflects the strength of the relationship. Logistic Regression is computationally efficient and handles high-dimensional data well, making it suitable for datasets with a large number of features. It can handle both continuous and categorical features, though appropriate encoding may be necessary for categorical variables. [15]

However, it is important to note that Logistic Regression assumes a linear relationship between the features and the log-odds of the outcome. This assumption may not hold in complex datasets where the relationship is nonlinear. In such cases, alternative models or feature engineering techniques may be required to capture the nonlinear relationships effectively. [15]

Random Forest

Random Forest is a learning algorithm that combines multiple decision trees to create a strong classification model. It is particularly useful when working with datasets that have binary outcomes, such as healthy (1) and unhealthy (0) states. [17, 19]

In Random Forest, each tree in the forest is trained on a random subset of the training data and a random subset of the available features. This randomization helps to reduce overfitting and increase the diversity among the individual trees. To make a prediction, each tree independently classifies the input based on the majority class of its leaf nodes. The final prediction is then determined by aggregating the predictions of all the individual trees through majority voting. [17, 19]

Random Forests have several advantages. They can handle high-dimensional data effectively, making them suitable for datasets with a large number of features. Additionally, they are capable of capturing complex interactions and non-linear relationships between features and the outcome. This makes them well-suited for datasets where the relationships may be more complex. [17, 19]

Moreover, Random Forests provide insights into feature importance. By evaluating the impact of each feature across the ensemble of trees, it is possible to determine which features have the most significant influence on the classification. This information can be valuable for understanding the underlying factors that contribute to being healthy or unhealthy. [17, 19]

However, Random Forests can be computationally expensive, especially when dealing with large datasets or a high number of trees in the forest. Additionally, the final model may be less interpretable compared to simpler models like Logistic Regression. [17, 19]

To understand the concept of the random forest there are an image and the formula.

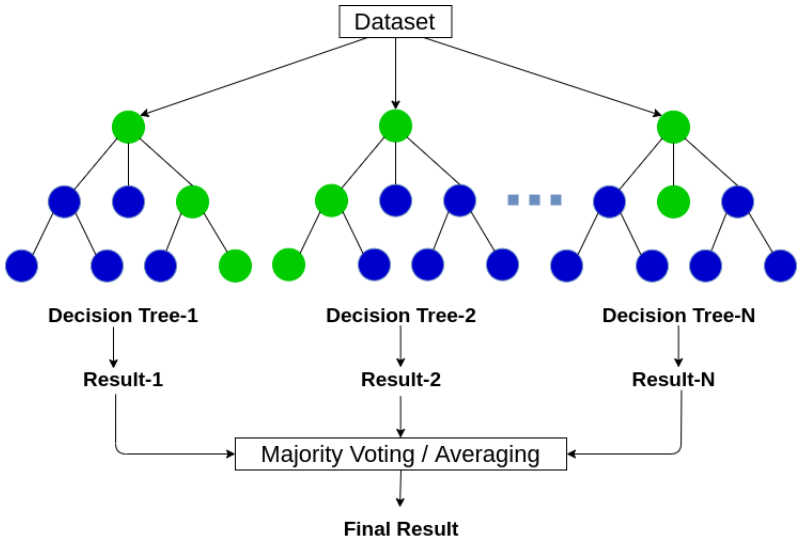

Figure 5: Random Forest Example

We would like to add that the formula for Random Forest is too complicated, so we prefer the formula for Random Forest that represents the overall concept and steps involved in the algorithm, rather than a specific mathematical equation. The prediction of a Random Forest model can be represented as:

$$RF(x) = \text{Aggregation}(Tree_1(x), Tree_2(x), \ldots, Tree_n(x)) \tag{2}$$

where $RF(x)$ represents the prediction for a given instance x, and $Tree_i$ (x) represents the prediction of the i-th decision tree in the Random Forest. The Aggregation step can vary depending on the task (classification or regression) and can involve majority voting, averaging, or other ensemble techniques. [17, 19]

Nearest Centroid (Closest Mean)

The Nearest Centroid algorithm, also known as Closest Mean, is a simple and intuitive classification algorithm. It represents each class by the mean or centroid of its feature vectors. This algorithm assumes that instances belonging to the same class are close to the centroid of that class in the feature space. [18]

During the training phase, the Nearest Centroid algorithm calculates the centroid for each class by computing the mean of the feature vectors belonging to that class. The centroid serves as a representative point that captures the average characteristics of the instances in the class. [18]

To classify a new instance, the algorithm calculates the distance between the instance and each class centroid. The most common distance metric used is the Euclidean distance, although other distance measures can also be used. The new instance is

then assigned to the class with the closest centroid, as it is assumed to be more similar to that class based on the distance calculation. [18]

The Nearest Centroid algorithm has several advantages. First, it is computationally efficient and has low memory requirements, as it only needs to store the centroids for each class. This makes it particularly suitable for large datasets or real-time applications. Additionally, the algorithm provides interpretability, as the class centroids can offer insights into the average values of the features for each class. [18]

However, it's important to consider the limitations of the Nearest Centroid algorithm. It assumes that instances within each class have similar covariance matrices, which means that the variance and shape of the feature distributions are assumed to be the same for each class. This assumption may not hold in datasets with uneven or overlapping class distributions. Additionally, the Nearest Centroid algorithm assumes equal prior probabilities, meaning that it assumes an equal likelihood of encountering instances from each class in the training data. [18]

In our context of a health-related application, the Nearest Centroid algorithm will be utilized to classify instances as healthy or unhealthy based on their feature vectors. By calculating the centroids of the healthy and unhealthy classes during training, the algorithm can determine the proximity of a new instance to each class centroid and assign it to the closest class. This approach can help in distinguishing healthy individuals from unhealthy ones based on the average characteristics captured by the centroids.

Image below will help to understand, how does Closest Mean Algorithm work.

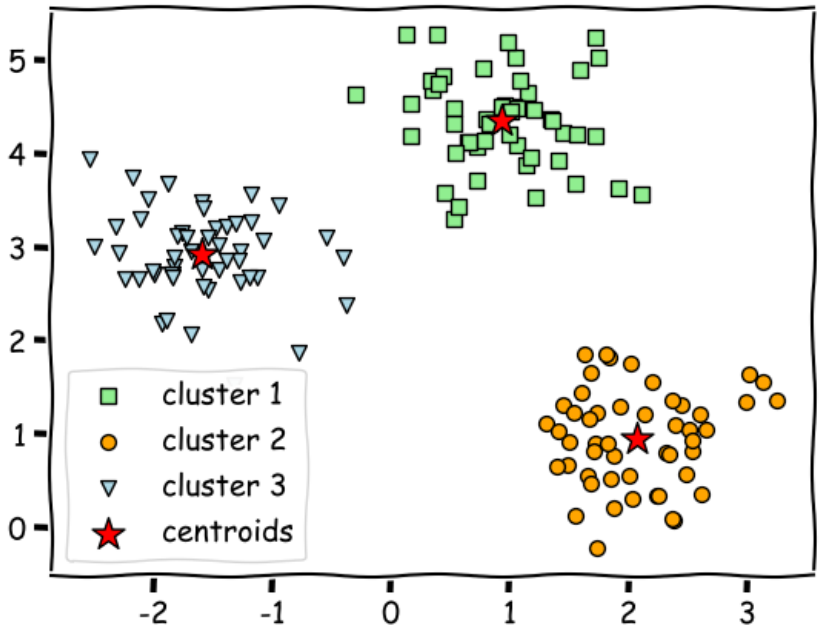

Figure 6: Nearest Centroid Example

Model Selection Process

When comparing and selecting the best model from KNN, random forest, and logistic regression, it is important to evaluate their performance using appropriate metrics that align with the specific task and objectives of your project. Here are some commonly used evaluation metrics for classification tasks that can help in selecting the best model:

1. Accuracy: Accuracy is the most basic and intuitive metric, measuring the overall correctness of the predictions. It represents the ratio of correctly classified instances to the total number of instances in the dataset. However, accuracy may not be suitable when the classes are imbalanced.

2. Cross-Validation Score: The cross-validation score, also known as mean cross-validated accuracy, provides an overall measure of the model's performance across different folds in cross-validation. It represents the average accuracy achieved by the model on the validation sets during cross-validation. Comparing the cross-validation scores of the models can help identify the one that performs consistently well across different subsets of the data.

3. Precision: Precision measures the proportion of true positive predictions out of the total predicted positives. It indicates the ability of the model to avoid false positives. Precision is particularly important when the cost of false positives

is high.

4. Recall (Sensitivity): Recall calculates the proportion of true positive predictions out of the total actual positives. It represents the model's ability to identify all positive instances correctly. Recall is crucial when the cost of false negatives is high, and you want to minimize the number of missed positive instances.

5. F1 Score: The F1 score is the harmonic mean of precision and recall. It provides a balanced measure that considers both precision and recall. It is especially useful when you want to find a balance between precision and recall.

6. Area Under the Receiver Operating Characteristic Curve (AUC-ROC): AUC-ROC measures the model's ability to discriminate between positive and negative instances across different probability thresholds. It provides an overall performance evaluation by considering the entire range of possible classification thresholds.[21]

Results

In the field of skin lesion classification, asymmetry and color features were employed as distinct attributes to discern the health status of a lesion. Our objective was to develop models that incorporate these features as input parameters and evaluate their performance using various metrics. By comparing the evaluation metrics, we aimed to identify the most accurate and reliable model for determining the health or disease status of skin lesions.

To ensure robust model evaluation and avoid overfitting, we employed a cross-validation technique called KFold. The dataset was divided into training and testing samples using a split of 75% and 25%, respectively. KFold then allowed us to further partition the training data into K equally sized folds, where K represents the number of folds or subsets. This process involves training the model K times, each time using a different fold as the validation set and the remaining folds as the training set. By performing KFold on the four different classifiers, we were able to obtain more reliable and generalized performance estimates, as the models were assessed on multiple variations of training and validation sets

Moreover, several bar charts were made, so that we can interpret results visually. The highest scores in both cases across the metrics took the Random Forest classifier, but in the average error it was the lowest. In AUC and cross-validation the model almost was the first, where the Nearest Centroid model took the lead in the situation of using colour features. In the asymmetry ones, the model obtained less results than Linear Regression in F1 Score and Recall.

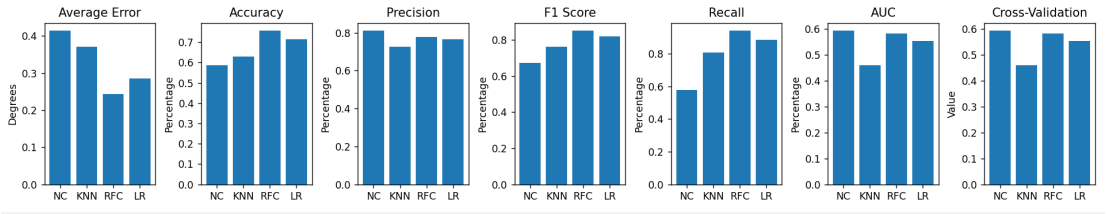

Figure 7: Metrics of the model based on colour features

Among the evaluated models for the color feature, the Random Forest Classifier demonstrated the highest performance based on the provided evaluation scores. The Random Forest Classifier achieved an accuracy of 77.14%, precision of 79.03%, F1 score of 85.96%, and recall of 94.23%. These metrics indicate a strong ability of the model to correctly classify the skin lesions based on the color feature. Additionally, the model showcased a relatively lower average error of 0.2286 degrees compared to the other models. The cross-validation score of 0.8036 further supports the robustness of the Random Forest Classifier. The high precision, recall, and F1 score indicate that the model can effectively distinguish between healthy and unhealthy lesions using color information. Thus, considering its superior performance across multiple evaluation metrics, the Random Forest Classifier stands out as the best model for skin lesion classification based on the color feature.

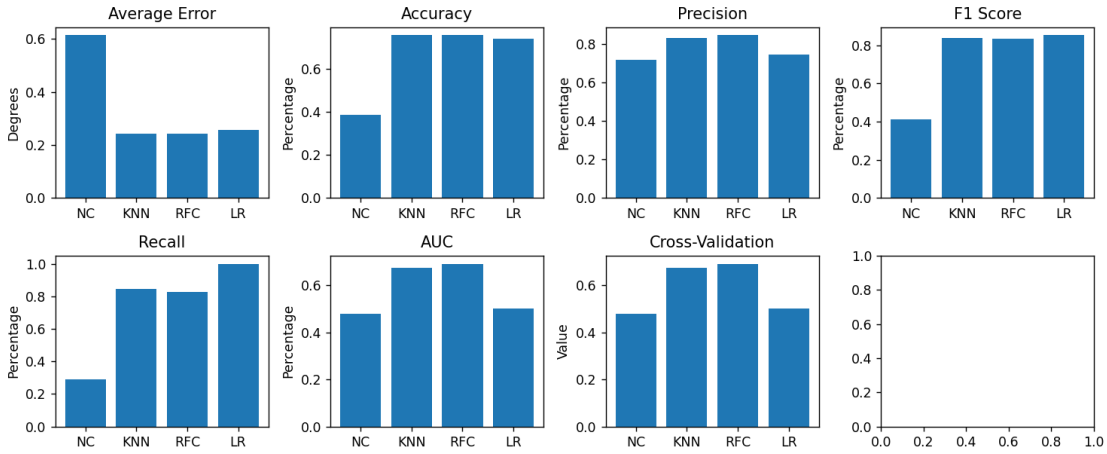

Figure 8: Metrics of the model based on asymmetry features

After evaluating the models based on both the color and asymmetry features, the Random Forest Classifier emerged as the most suitable model for our skin lesion classification task. The Random Forest Classifier demonstrated superior performance in terms of accuracy, precision, F1 score, recall, and ROC-AUC for both the color and asymmetry features. Notably, for the color feature, the Random Forest Classifier achieved an accuracy of 77.14%, precision of 79.03%, F1 score of 85.96%, and recall of 94.23%. Likewise, for the asymmetry feature, the Random Forest Classifier attained an accuracy of 75.71%, precision of 83.02%, F1 score of 83.81%, and recall

of 84.62%.

In addition to its exceptional performance, the Random Forest Classifier showcased a relatively lower average error compared to the other models. Furthermore, the Random Forest Classifier exhibited robustness and generalization ability, as indicated by its higher cross-validation score compared to the other models. The Random Forest Classifier's ability to effectively capture the underlying patterns and relationships within the color and asymmetry features contributes to its selection as the optimal model for skin lesion classification. Therefore, based on its superior performance across multiple evaluation metrics and its robustness, we have chosen the Random Forest Classifier as our preferred model for accurate skin lesion diagnosis.

After carefully evaluating the performance of the Random Forest Classifier on both the color and asymmetry features for skin lesion classification, we have chosen the Random Forest Classifier specifically for the color feature. The Random Forest Classifier demonstrated remarkable performance with an accuracy of 77.14%, precision of 79.03%, F1 score of 85.96%, and recall of 94.23% for the color feature. These evaluation metrics indicate the model's ability to accurately predict the diagnosis of skin lesions based on the color feature. Additionally, the Random Forest Classifier achieved a higher cross-validation score of 80.36% for the color feature, further validating its robustness and generalization ability.

The decision to select the Random Forest Classifier for the color feature was also influenced by its relatively lower average error of 0.2286 degrees, indicating a smaller deviation in its predictions. Furthermore, while the ROC-AUC score for the color feature was 61.00%, which is comparatively lower than the asymmetry feature, the overall performance metrics, including accuracy, precision, F1 score, and recall, were superior for the color feature with the Random Forest Classifier.

Improving model

We successfully enhanced our top-performing model, the Random Forest, by optimizing its hyperparameters. We employed two distinct techniques for this purpose. Initially, we utilized Random Search to fine-tune the hyperparameters, by randomly sampling combinations from a predefined search space to identify the best configuration based on the performance metric. We then performed grid search which exhaustively explores all the possible combinations of hyperparameters within a predefined grid to find the optimal configuration after the initial random search. [23]

Random search

Random search is a hyperparameter optimization technique used in different machine learning models. It involves randomly sampling combinations of hyperparameters from a defined search space and evaluating their performance. By exploring different hyperparameter configurations in a accidental manner, random search helps identify the optimal set of hyperparameters for a given machine learning task, offering an efficient alternative to exhaustive search methods like grid search. However, we did grid search too. [23]

Grid search

Grid search is a hyperparameter optimization technique in machine learning that involves defining a grid of hyperparameter values for a model. It exhaustively searches through all possible combinations of hyperparameters within the defined grid and evaluates the model's performance for each combination using a specified metric. By systematically exploring the hyperparameter space, grid search helps find the optimal set of hyperparameters that yield the best performance for the given machine learning task. It is a straightforward and thorough method, but it can be computationally expensive, especially when the hyperparameter space is large. [23]

Improving model for the asymmetry feature

After tuning hyperparameters we receive new evaluation metrics. Below that you can see old and new data.

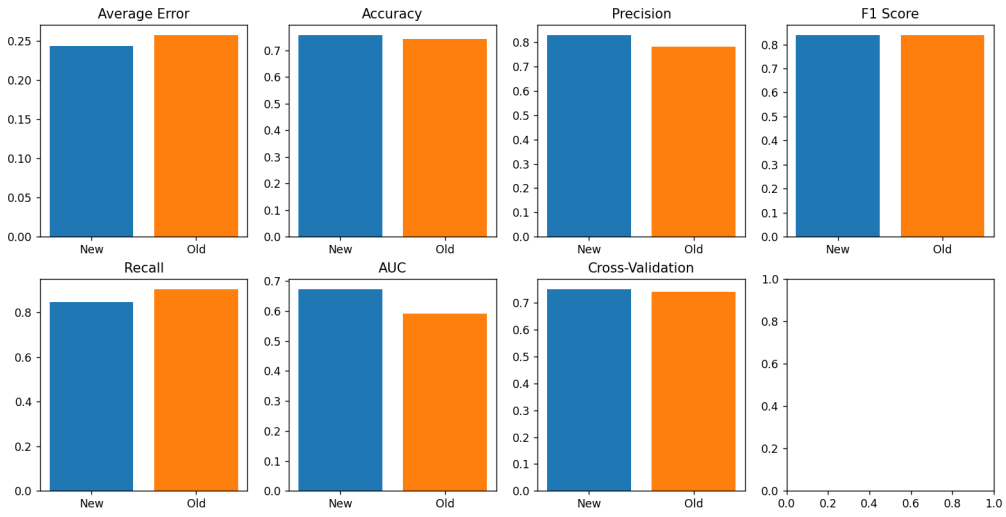

Figure 9: Random Forest's evaluation metrics for the asymmetry before and after Random and Grid Search

After comparing the evaluation metrics of the old and new models, it is evident that

the new model performs slightly better in several aspects. The average error has decreased from 0.2571% to 0.2429%, indicating improved accuracy in predicting outcomes. The new model also exhibits a higher accuracy rate of 75.7143%, compared to 74.2857% in the old model. Precision has significantly increased from 78.3333% to 83.0189%, indicating a better ability to correctly identify positive cases. The F1 score has slightly decreased from 83.9286% to 83.8095%, reflecting on a balanced measure of precision and recall. Although the recall has marginally decreased from 90.3846% to 84.6154%, it remains at a high level. The area under the curve (AUC) has notably increased from 59.0812% to 67.3077%, indicating an improved ability to distinguish between positive and negative instances. Moreover, the new model's cross-validation score for Random Forest has risen to 0.7512 compared to 0.7424 for the old model. Considering these improvements across various evaluation metrics, it can be concluded that the new model outperforms the old model, demonstrating enhanced performance and predictive capabilities. However, it was still worse than the new tuning Random Forest model for colour feature.

Improving model for the colour feature

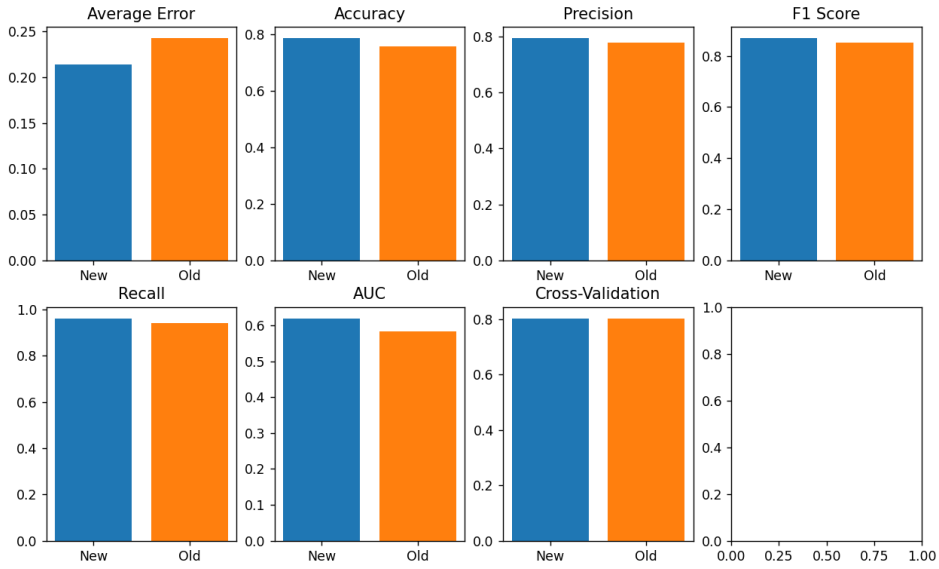

Figure 10: Random Forest's evaluation metrics for the asymmetry before and after Random and Grid Search

Upon comparing the evaluation metrics of the old and new models specifically for the colour feature, it is evident that the new model demonstrates improvements across multiple performance measures. The average error has decreased from 0.2429% to 0.2143%, indicating enhanced accuracy in predicting colour outcomes. The new model exhibits a higher accuracy rate of 78.5714 % compared to 75.7143% in the old model, suggesting better overall classification performance. Precision has also

increased from 77.7778% to 79.3651%, indicating an improved ability to correctly identify positive colour cases. The F1 score has risen from 85.2174% to 86.9565%, reflecting a balanced measure of precision and recall. The new model demonstrates a slightly higher recall of 96.1538% compared to 94.2308% in the old model, indicating an enhanced ability to capture positive colour instances. The ROC AUC score has increased from 58.2265% to 61.9658%, suggesting an improved ability to distinguish between positive and negative colour instances. Additionally, the cross-validation score for Random Forest have improved in the new model, with values of 0.8035 respectively, compared to 0.8033 in the old model.

Based on these improvements across various evaluation metrics, it can be concluded that the new model outperforms the old model for the colour feature, showcasing enhanced performance and predictive capabilities.

Final Result

After evaluating multiple models (KNN, Closest mean, Random Forest, and Logistic Regression), we determined that Random Forest outperformed the rest, leading us to select it as the preferred choice. With two features, colour and asymmetry, we found that colour exhibited superior performance based on evaluation metrics. While initially considering tuning parameters exclusively for the colour feature, we decided to also explore tuning for both colour and asymmetry to assess if the latter could surpass the former. However, the results indicated that it was not worthwhile as the asymmetry evaluation did not surpass the performance of the colour feature. Hence, the best model remains the Random Forest, specifically for the colour feature with parameter tuning.

Conclusion

In conclusion, our research focused on computer-aided diagnosis of skin lesions using image analysis and machine learning techniques. We addressed several key questions related to the selection of informative image features, effective feature extraction methods, and the identification of high-performing machine learning classifiers for accurate skin lesion diagnosis.

To extract relevant features, we employed image segmentation techniques such as gaussian filtering, thresholding, and regional segmentation. Additionally, we utilized the SLIC algorithm to extract color features from the segmented images. Moreover, we considered asymmetry as an important feature in our analysis.

For the classification task, we evaluated multiple machine learning classifiers, including KNN, random forest, logistic regression, and closest mean. We measured the

performance of these classifiers using various evaluation metrics, including Cross-Validation Score, Accuracy, Precision, Recall (Sensitivity), and F1 Score. Through rigorous experimentation, we observed that the image color features extracted using the SLIC algorithm provided the most informative data for training the models, resulting in the highest scores.

Based on the evaluation results, we selected random forest as our preferred classifier due to its superior performance across the evaluated metrics. However, we further enhanced the model's performance by conducting RandomSearchCV and GridSearchCV to optimize the hyperparameters.

Our final model achieved an accuracy of 77% and a cross-validation score of approximately 81%. These results demonstrate the potential of our approach for assisting in the diagnosis of skin lesions using image analysis and machine learning techniques.

Implications for future work

In future work, it would be beneficial to explore more advanced machine learning models, such as deep learning architectures like convolutional neural networks (CNNs), which have shown promising results in image analysis tasks. Additionally, incorporating additional clinical and demographic data could further improve the accuracy and robustness of the diagnosis. Furthermore, expanding the dataset and conducting multi-center studies would enhance the generalizability of the model. These directions can contribute to the ongoing advancements in computer-aided diagnosis of skin lesions, ultimately improving early detection and treatment outcomes for dermatological conditions.

One promising avenue for improvement is the utilization of deep learning models, particularly convolutional neural networks (CNNs), which have demonstrated exceptional performance in various image-based tasks. CNNs can automatically learn hierarchical representations from images, capturing complex spatial dependencies and intricate patterns. By leveraging CNNs, we can extract more informative features and potentially achieve higher prediction accuracy, especially in scenarios with large-scale and high-dimensional image data.

Another avenue to explore is the utilization of ensemble methods, such as gradient boosting machines (GBMs) and XGBoost. These techniques combine multiple weak models to form a stronger, more accurate predictor. GBMs and XGBoost, in particular, have gained significant popularity due to their ability to handle complex relationships, handle missing data, and mitigate overfitting. By leveraging ensemble methods, we can harness the diversity of multiple models and benefit from their

collective decision-making capabilities.

Furthermore, deep reinforcement learning (DRL) offers exciting prospects for advancing our prediction capabilities. DRL integrates reinforcement learning principles with deep neural networks, allowing models to learn optimal decision-making policies through trial and error. By applying DRL techniques, our models can not only predict outcomes but also adapt and improve their predictions based on feedback received during the decision-making process.

Lastly, transfer learning is a technique that warrants exploration. By leveraging pre-trained models on large-scale datasets, we can transfer the knowledge and learned representations to our specific prediction task. This approach can significantly enhance prediction accuracy, especially when faced with limited training data or when dealing with similar domains where pre-trained models have already achieved impressive performance.

The automated system can potentially assist healthcare professionals in accurately diagnosing skin lesions, reducing the risk of misdiagnosis and enabling early detection of malignant lesions. This can lead to timely intervention and improved patient outcomes. By automating the diagnostic process, the system developed in this project has the potential to improve efficiency and reduce healthcare costs. It can alleviate the workload of dermatologists by pre-screening and prioritizing cases, optimizing their time and resources. The findings and insights from this project can contribute to ongoing research in the field of computer-aided diagnosis of skin lesions. They can serve as a foundation for further advancements in image analysis techniques, feature extraction, and machine learning algorithms for more accurate and robust diagnosis.

References

1. World Health Organization. (2019). Skin cancers. Retrieved from https://www.who.int/news-room/fact-sheets/detail/skin-cancers
2. Celebi, M. E., et al. (2018). A state-of-the-art survey on lesion border detection in dermoscopy images. Skin Research and Technology, 24(2), 189-202.
3. Codella, N. C., et al. (2018). Skin Lesion Analysis Toward Melanoma Detection: A Challenge at the 2017 International Symposium on Biomedical Imaging (ISBI), Hosted by the International Skin Imaging Collaboration (ISIC). IEEE Transactions on Medical Imaging, 37(2), 1116-1130.
4. Esteva, A., et al. (2017). Dermatologist-level classification of skin cancer with deep neural networks. Nature, 542(7639), 115-118.
5. Tschandl, P., et al. (2018). Comparison of the accuracy of human readers versus machine-learning algorithms for pigmented skin lesion classification: an open, web-based, international, diagnostic study. The Lancet Oncology, 20(7), 938-947.
6. Pacheco, A. G. C. (2020). PAD-UFES-20: a skin lesion dataset composed of patient data and clinical images collected from smartphones [Data set]. Mendeley. https://doi.org/10.17632/ZR7VGBCYR2.1
7. Tamilselvan, K. S., & Murugesan, G. (2018). Image Segmentation. InTech. doi:10.5772/intechopen.76428
8. Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2017). Imagenet classification with deep convolutional neural networks. Communications of the ACM, 60(6), 84-90.
9. Saponara, S., & Elhanashi, A. (2022). Impact of Image Resizing on Deep Learning Detectors for Training Time and Model Performance. In Lecture Notes in Electrical Engineering (pp. 10–17). Springer International Publishing. https://doi.org/10.1007/978-3-030-95498-7_2
10. Ge, Y., Zhang, Q., Sun, Y. et al. Grayscale medical image segmentation method based on 2D&3D object detection with deep learning. BMC Med Imaging 22, 33 (2022). https://doi.org/10.1186/s12880-022-00760-2
11. Mahmood, H. F. (2023). What is Gaussian Blur in image processing?. Educative. https://www.educative.io/answers/what-is-gaussian-blur-in-image-processing
12. Chatterjee, S. (2022, July 26). What is feature extraction? feature extraction in image processing. Great Learning Blog: Free Resources what

Matters to shape your Career! https://www.mygreatlearning.com/blog/feature-extraction-in-image-processing/

13. Alpaydin, Ethem (2010). Introduction to Machine Learning. London: The MIT Press. p. 110. ISBN 978-0-262-01243-0. Retrieved 1 June 2023

14. Achanta, Radhakrishna & Shaji, Appu & Smith, Kevin & Lucchi, Aurélien & Fua, Pascal & Süsstrunk, Sabine. (2010). SLIC superpixels. Technical report, EPFL.

15. Dreiseitl, S., & Ohno-Machado, L. (2002). Logistic regression and artificial neural network classification models: a methodology review. In Journal of Biomedical Informatics (Vol. 35, Issues 5–6, pp. 352–359). Elsevier BV. https://doi.org/10.1016/s1532-0464(03)00034-0.

16. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... & Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. the Journal of machine Learning research, 12, 2825-2830. Murugan, A., Nair, S. A. H., & Kumar, K. S. (2019). Detection of skin cancer using

17. Murugan, A., Nair, S. A. H., & Kumar, K. S. (2019). Detection of skin cancer using SVM, random forest and kNN classifiers. Journal of medical systems, 43, 1-9. https://doi.org/10.1007/s10916-019-1400-8

18. Levner, I. (2005). Feature selection and nearest centroid classification for protein mass spectrometry. BMC bioinformatics, 6(1), 1-14. https://doi.org/10.1186/1471-2105-6-68

19. Rodriguez-Galiano, V. F., Ghimire, B., Rogan, J., Chica-Olmo, M., & Rigol-Sanchez, J. P. (2012). An assessment of the effectiveness of a random forest classifier for land-cover classification. In ISPRS Journal of Photogrammetry and Remote Sensing (Vol. 67, pp. 93–104). Elsevier BV. https://doi.org/10.1016/j.isprsjprs.2011.11.002

20. Talavera-Martínez, L., Bibiloni, P., Giacaman, A., Taberner, R., Hernando, L. J. D. P., & González-Hidalgo, M. (2022). A novel approach for skin lesion symmetry classification with a deep learning model. In Computers in Biology and Medicine (Vol. 145, p. 105450). Elsevier BV. https://doi.org/10.1016/j.compbiomed.2022.105450

21. Mallett, S., Halligan, S., Collins, G. S., & Altman, D. G. (2014). Exploration of analysis methods for diagnostic imaging tests: problems with ROC AUC and confidence scores in CT colonography. PloS one, 9(10), e107633. https://doi.org/10.1371/journal.pone.0107633

22. Anguita, D., Ghelardoni, L., Ghio, A., Oneto, L., & Ridella, S. (2012, April).

The'K'in K-fold Cross Validation. In ESANN (pp. 441-446).

23. Bergstra, J., & Bengio, Y. (2012). Random search for hyper-parameter optimization. Journal of machine learning research, 13(2).