– Independent Work Report Spring 2021 –

Computational Melanoma Detection: As Easy as ABC?

David Lipman
Advisor: Olga Russakovsky

Abstract

In this paper, we develop a non deep-learning approach to melanoma classification that utilizes a feature engineering approach centered around the ABCD rule of skin cancer, which states that skin lesions that are A: asymmetric in terms of shape or texture, B: have irregular or poorly defined borders, C: many colors or color variations from one area to another, and D: diameters of 6mm or larger are more likely to be melanoma. Although deep learning approaches have historically performed better on this task, one of the main advantages of using a non deep-learning approach for melanoma detection is that it increases model interpretability and allows for analysis of feature importance. We focus our analysis on developing a generalizable model to detect melanoma based on these ABCD features, with our optimal random forest model trained on a dataset of 10,180 dermoscopic images yielding an accuracy of 81.9%. Finally, according to our analysis of feature importance, the number of unique colors appearing within a lesion, the intersection between 3D histograms of colors within and outside of each lesion, and the irregularity of a lesion's border are most informative in determining a lesion's diagnosis. In addition, the "C" features overall appear to have the most predictive power in classifying a lesion.

1. Introduction

Melanoma is the deadliest form of skin cancer, and while it only accounts for ~1% of annual skin cancer cases, it causes the vast majority of all skin cancer deaths and is expected to kill close to 7,000 Americans this year[1]. Luckily, Melanoma is usually curable when detected early, as the estimated five-year survival rate for US patients whose melanoma is detected early is 99%[2]. However, if the melanoma is detected once it has already metastasized, survival chances are highly

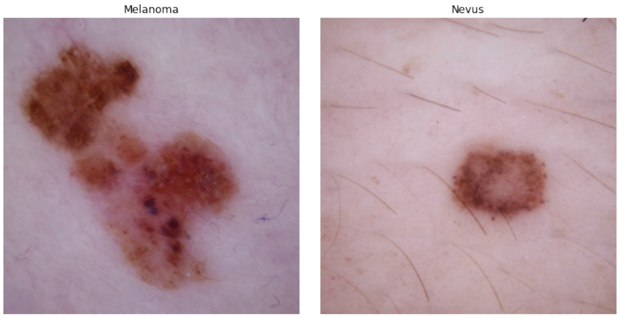


Figure 1: Example of melanoma and nevus skin lesions.

reduced, with the estimated five-year survival rate decreasing to 15-20% [3]. Thus, it is critical to effectively diagnose melanoma early in its development.

Melanoma is typically characterized by the occurrence of skin lesions with some distinguishable features, such as irregular borders, asymmetric shapes, large areas, dark colors, and color variation [4]. An example of a malignant lesion exemplifying some of these features shown side-by-side with a benign skin lesion (known as nevus) is depicted in Figure 1. Even though some lesions can be easily classified by the presence or absence of some of these features, naked eye lesion diagnosis is still unreliable, even by experts [5]. Recent advancements in medical imaging of skin cancers - namely the dermoscope, a skin imaging tool consisting of a magnifier, a light source, and a transparent plate that can magnify lesions up to 100x- have already vastly improved diagnostic accuracy of melanoma when compared to naked eye examination [5]. Experts have additionally designed diagnostic methods to quantify the malignancy of skin lesions based on dermoscopic images. Perhaps the most common diagnostic method used to diagnose melanoma clinically is the ABCD rule of dermoscopy. According to the ABCD rule, skin lesions that are A: asymmetric in terms of shape or texture, B: have irregular or poorly defined borders, C: many colors or color variations from one area to another, and D: diameters of 6mm or larger are more likely to be melanoma [6]. A graphic explaining this rule is shown in Figure 2.

Regardless of these recent advancements, skin cancer diagnosis remains quite subjective and additionally requires a high level of expertise [5]. Due to these factors, the past decade has seen an increase in the development of computer-aided diagnosis (CAD) systems, which seek to further increase diagnostic accuracy and help dermatologists produce accurate diagnoses more rapidly.

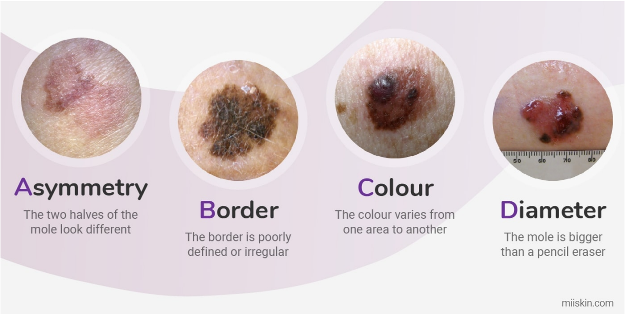

Asymmetry
The two halves of the mole look different
Border
The border is poorly defined or irregular
Colour
The colour varies from one area to another
Diameter
The mole is bigger than a pencil eraser
miiskin.com

Figure 2: Graphic explaining ABCD rule of dermoscopy [4]

A variety of deep learning and non-deep learning techniques have been applied to create these CAD systems, with deep learning approaches historically performing better in terms of pure performance metrics. However, in this research we focus on developing a non deep-learning approach to melanoma detection, primarily because they provide the main advantage of increased model interpretability and insight into feature importance.

A non deep-learning CAD system typically consists of the following three subsections: 1) Lesion segmentation - lesions are segmented from the rest of the image in order to isolate the region of interest, 2) Feature extraction - various features often related to asymmetry, border irregularity, and colors are extracted from each lesion 3) Lesion classification - a machine learning model is trained on the extracted features to predict lesion diagnosis. The majority of the non deep-learning CAD systems developed use the ABCD rule as inspiration for their feature extraction process, as they often engineer features related to lesion asymmetry, borders, and colors. While some of these previous systems have achieved good accuracies ($> 90\%$), they largely contain various shortcomings that we seek to address with this research, as will be explained in Section 2.

The objective of this research is to develop a non deep-learning CAD system utilizing the ABCD rule for feature engineering in order to diagnose melanoma with high accuracy and generalizability, and to experimentally analyze the ABCD rule to determine which features are most informative in diagnosing melanoma. Finally, we seek to identify whether the feature effects determined by our classification model align with the clinical intuition motivating the ABCD rule.

2. Problem Background and Related Work

In order to provide insight into the previous research conducted on computational melanoma detection, we describe two categories of research. Firstly, we provide a brief overview of some recent deep learning CAD systems to provide some context for what current state of the art CAD systems look like. Secondly, we provide an overview of various non deep-learning CAD systems that largely utilize the ABCD rule for feature engineering. Here we also describe papers that are specifically focused on designing optimal feature engineering techniques related to the ABCD rule, and are not CAD systems themselves. The goal of this section is to provide the reader with a broader understanding of the field of automated melanoma detection, and to illuminate two major shortcomings of previous works that we seek to address in this research.

2.1. Deep-Learning CAD Systems

Esteva et. al., 2017 proposes a CAD system that uses a single convolutional neural network (CNN), trained end-to-end from images directly, using only pixels and disease labels as inputs. They train this CNN on a dataset of 129,450 clinical images, and test its performance against 21 board-ceritfied dermatologists on clinical images classified as melanoma and nevus (benign). The CNN achieved performance on par with with these experts (CNN AUC of 0.96 compared to expert AUC of 0.94), showing that deep learning based CAD systems are capable of classifying melanoma at accuracies comparable to dermatologists [7].

Ha et. al., 2020 presents the winning solution to the 2020 SIIM-ISIC Melanoma Classification Challenge [8]. Their method uses an ensemble of CNN models with different backbones (EfficientNets and ResNets trained on ImageNet) and image sizes. They trained these ensembled models on 33,000 dermoscopic skin lesion images using 5-fold cross validation, and achieved a final AUC of 0.949 on the final test set [9].

The intention behind sharing these deep learning approaches is twofold: Firstly, we wish to demonstrate that CAD systems have been trained on large datasets of dermoscopic images to achieve high accuracy - comparable to human experts - on the task of melanoma classification.

Secondly, we wish to provide motivation for why we chose to develop a non deep-learning CAD system. Despite their higher accuracies, deep learning models are significantly less interpretable than non deep-learning models. As such, previous deep learning approaches have yielded these high accuracies without fully comprehending what features are motivating their models' predictions. In our research, we seek to address this knowledge gap by developing a non deep-learning model and analyzing how much each ABCD feature we extract contributes to this model's final prediction.

2.2. Non Deep-Learning CAD Systems

We share an overview of previously developed non deep-learning CAD systems, in addition to papers related to ABCD feature engineering techniques, as these are the most closely related to our research:

Ruela et. al., 2013 analyzes the role of color features in classifying skin lesions by extracting color symmetry features in a dataset of 169 dermoscopic images. To compute the color symmetry for each lesion, lesions are divided based on their axis of symmetry and are divided into $n \times n$ patches on each side of the axis. Color features are then extracted from each patch, consisting of the mean color vector (MCV) of each lesion, the uni-dimensional color histogram (UCH) of each lesion, and the generalized color moments (GCM) of each lesion. Each of these features are incorporated into a vector, and the Euclidean distance between feature vectors for symmetric pairs of patches are computed and summed over all patches. Using this Euclidean distance metric for different symmetry axis rotations as input features to a k-NN classifier, 100% sensitivity and 75% specificity are achieved using automated lesion segmentation [10].

El Abbadi et. al., 2017 proposes a classification method that utilizes the ABCD rule to detect melanoma in a dataset of 220 dermoscopic images of skin lesions. Initially, lesion images are filtered to remove hair and other unwanted particles, and then a method for automatic lesion segmentation based on Markov and Laplace filters to detect edges is applied to each lesion image. Asymmetry, border, color, and diameter features are extracted from each image. The Total Dermoscopy Score (TDS) is computed for each image according to the equation TDS = 1.3A + 0.1B + 0.5C + 0.5D.

Any lesion with TDS > 5.45 is classified as melanoma, and any other lesion is classified as benign. This technique achieved an accuracy of 95.45% [11].

Kasmi et. al., 2016 proposes a classification method using the ABCD rule of dermoscopy that automatically segments lesions using Gabor filters and geodesic active contours. Similar to the El Abbadi et. al. study, this research extracts ABCD features from a dataset of 200 dermoscopic images, computes the TDS score for each lesion, and achieves a sensitivity of 91.25% and specificity of 95.83% using this strategy [12].

Murugan et. al., 2019 proposes using the watershed algorithm for lesion segmentation. Shape, ABCD, and GLCM (Gray-Level Co-Occurrence Matrix) features are extracted from each segmentation. Lesions are classified based on these extracted features using 10-fold cross-validation on Support Vector Machine (SVM) and Random Forest models. A dataset of 1,000 skin lesion images is used, and average validation accuracy of 89.43% is achieved with an SVM training only on the ABCD features, while average validation accuracy of 76.87% is achieved with a Random Forest training only on the ABCD features [13].

Pham et. al., 2019 proposes extracting HSV, LBP, HOG, and SIFT features from images of skin lesions and using Balanced Random Forests to classify them. A dataset consisting of over 10,000 skin lesion images is analyzed, and accuracy of 74.75% with AUC of 81.46% is achieved on a test set of 1,000 images using only HSV features. This is essentially testing the classification ability of color-specific features [14].

Almeida et. al., 2020 proposes a method in which image keypoints are computed using both first and second order Gray Level Co-occurrence Matrix features and RGB component information from each image. A dataset of 2,000 dermoscopic images is used in this study, and a logistic regression model trained on these features yields an AUC of 97%. The key insight in this study is that lesion segmentations were not computed, and GLCM features were computed over each entire lesion image, thus not making the accuracy of computed features reliant upon the success of the segmentation algorithm [15].

While not a paper proposing a CAD system, Toureau et. al., 2019 proposes computational methods to detect symmetry in dermoscopic images of skin lesions. Algorithms are proposed to detect a lesion's symmetry based on its shape, color/texture, and a combination of the two. To develop these algorithms, the authors used the PH2 database, a dataset of 200 dermoscopic images annotated by expert dermatologists[16]. For each image, manual expert segmentations of lesions were provided, along with manually computed information about dermoscopic features such as asymmetry and colors. Symmetry scores of 0, 1, or 2 are predicted for each lesion, with 0 corresponding to a fully symmetric lesion, 1 corresponding to asymmetric lesions with respect to one axis of symmetry, and 2 corresponding to asymmetric lesions with respect to two axes of symmetry. Here, the axes of symmetry refer to two orthogonal axes that intersect each other at the centroid of each lesion. The success of this feature engineering technique is evaluated by comparing predicted asymmetry scores to expert-labeled asymmetry scores. The shape asymmetry feature yields an accuracy of 86% (meaning 86% of lesions were scored the same as the expert-labels), while the texture asymmetry feature yields an accuracy of 84% [17]. We utilize this same approach to extract asymmetry features in our research, as mentioned in Subsection 4.4.

2.3. Drawbacks of Prior Research

While many of the non deep-learning CAD systems presented in Subsection 2.2 yielded accuracies greater than 90%, they face two primary drawbacks that we seek to address in our research.

1. Many of these previously proposed approaches seem to have poor generalizability for a variety of reasons. Chiefly among these reasons is the usage of small datasets, as the majority of the previously proposed non deep-learning CAD systems rely on datasets with $\leq 1,000$ images, such was the case with Ruela et. al. (169 images), El Abbadi et. al. (220 images), Kasmi et. al. (200 images), and Murugan et. al (1,000 images). Due to the small size of these datasets, it is unclear whether the methods proposed in these works would generalize well to significantly larger real-world datasets. This is especially true considering the fact that El Abbadi et. al. and Kasmi et. al. use rule-based classification systems and do not evaluate their systems on out of sample data, in addition to the fact that Ruela et. al. only reports evaluation metrics on training

data.

2. Most importantly, one of the main advantages of using non deep-learning approaches for melanoma detection is that they allow for increased model interpretability, particularly through analysis of feature importance. None of the prior approaches mentioned in Subsection 2.2 analyze the respective importances of the various ABCD features they extract. It would be extremely useful to gain knowledge about which features and which classes of features (Asymmetry, Border, Color, and Diameter) contribute most to lesion classification, and to gain knowledge about whether the clinical intuition behind these ABCD features align with their effects in classification models.

3. Approach

Our approach aims to address the drawbacks of prior non deep-learning CAD systems using the following methods:

In order to create a more generalizable non deep-learning CAD system utilizing the ABCD rule for feature engineering, we use a dataset of 10,180 dermoscopic images (half melanoma, half nevus) collected by the International Skin Imaging Collaboration and published for both the 2019 and 2020 SIIM-ISIC Melanoma Classification Challenges [8] [18] [19] [20]. In addition to dermoscopic images, this dataset also provides accompanying patient-level metadata - namely a patient's age and sex - which we add to our feature space to gain slight performance boosts. We experiment with training classification models using linear models such as Logistic Regression and Linear SVMs, in addition to Random Forest Classifiers. We hyperparameter tune these models using 4-fold cross validation with validation sets containing ~2,000 images, and finally evaluate these models on a test set containing ~2,000 images. We argue that the combination of using a significantly larger dataset than many previous works and training classification models that generalize well on this dataset will help to create a more robust and generalizable non deep-learning CAD system.

Finally, the novel insight of our research is the analysis of feature importance in our optimal Random Forest model. We conduct three types of analysis to accomplish this goal: Firstly, we conduct feature importance analysis of individual features using the permutation importance metric and sequential feature selection to determine how much each individual ABCD feature (listed in the Appendix) contributed to the final model. Secondly, we train a Random Forest model on each class of features (out of A, B, C, and D) to gain an experimental understanding of which feature class performs best at classifying skin lesions in isolation. Finally, we use analysis of SHAP values to determine the effects of each individual ABCD feature (i.e., does a higher value in a given feature lead the model to predict melanoma or benign?) in order to analyze whether the clinical intuition behind these features align with their contribution to classification models.

The final non deep-learning CAD system that we develop has the key subsections mentioned in Figure 3: Image preprocessing, lesion segmentation, ABCD feature engineering, lesion classification, and feature importance analysis. Image preprocessing, lesion segmentation, and ABCD feature engineering will be described in detail in Section 4. Lesion classification and feature importance analysis will be described in Section 5.

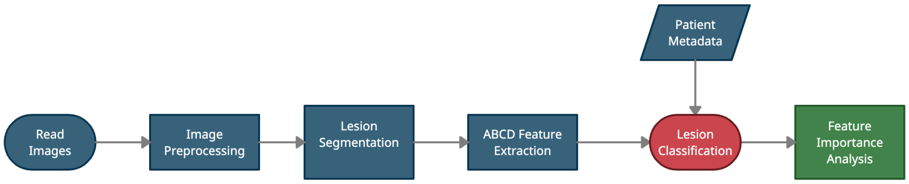


Figure 3: Flowchart of our CAD system.

4. Implementation

4.1. Image Preprocessing

Dermoscopic skin images are in their nature not clean, and contain undesirable particles such as hairs, air bubbles, and gel. We thus wrote an algorithm that uses the Python package OpenCV to preprocess each image and removes hairs [21]. The algorithm works as follows:

We first convert each image to grayscale. Then, we apply a morphological black hat transformation to the resulting grayscale image. The black hat transformation - defined as the difference between an input image and its closing - is commonly used in image processing to extract the elements in an image that are smaller than a given structuring element (in this algorithm, we use a structuring element of size 17 × 17 pixels) and are darker than their surroundings[21]. Next, we apply a binary threshold to this transformed image, replacing any pixel with value ≥ 10 with a value of 255, and any pixel with value ≤ 10 with a value of 0[21]. Finally, we inpaint with the original image and the threshold image in order to remove small hairs and unwanted particles. Inpainting is a technique commonly used in image processing to replace small, unwanted marks in an image with their neighboring pixels so that they appear similar to their local neighborhood[21]. The inpainting algorithm we use - which is built-in to OpenCV - is based on the Fast Marching Method, and replaces the desired region to be inpainted (in our case all pixels of value 255 identified in the threshold) with a weighted average of its local neighborhood of pixels[22]. The intermediary results of this algorithm are displayed in Figure 4.

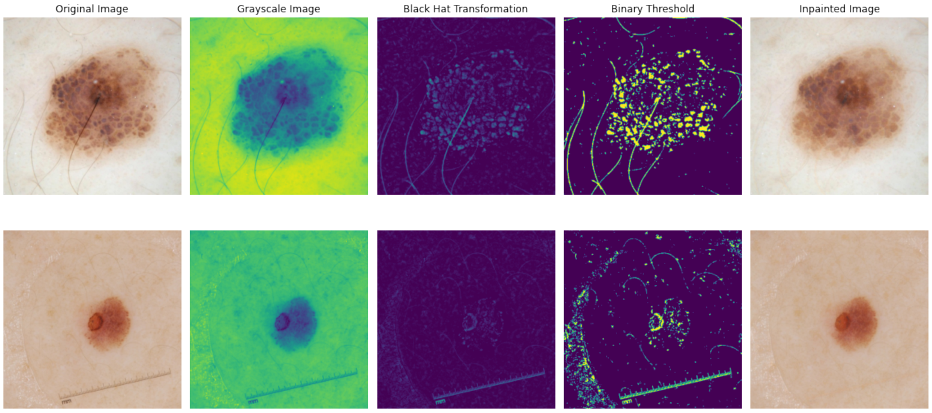


Figure 4: The preprocessing process. From left to right, we have: The original image, the image converted to grayscale, the image after the black hat transformation is applied with a 17 × 17 structuring element, the transformed image after the binary threshold is applied, and the final inpainted image. As can be seen, the hairs in the first image and dermoscope remnants in the second image are removed by this algorithm, and are replaced with their neighboring pixels.

After applying this algorithm, we further apply a 5 × 5 median filter to each image before segmentation in order to minimize the effects of remaining hairs and remove any noise and unwanted objects such as air bubbles. This filter is applied by convolving each image with a 5 × 5 kernel that computes the median value of each kernel and replaces the center pixel with the median value[21]. After this preprocessing phase is complete, we are able to compute lesion segmentations.

4.2. Lesion Segmentation

In order to properly extract meaningful features from skin lesions, it is first necessary to segment these lesions from the rest of the image to isolate the region of interest from which features should be extracted. Skin lesion segmentation is an open area of research, with multiple papers published using deep learning approaches to segment lesions with high accuracy [23][24][25]. In this research, we developed a simpler segmentation algorithm that works as follows:

First, we use Otsu's thresholding to separate the image into two regions: one for lesions and one for skin. Otsu's thresholding differs from the binary threshold used in preprocessing because it determines an optimal global threshold value from an image's histogram rather than arbitrarily choosing a value as a threshold (as we did with 10 while preprocessing) [21]. Applying this thresholding results in a binary image, where lesion pixels have value 255 and skin pixels have value 0. Next, a morphological opening transformation is applied to remove excess small objects (small, disconnected regions with value 255 in the binary image), and a morphological closing transformation is applied to fill small holes within segmented lesions (small regions of value 0 fully enclosed in a region of value 255)[21].

Due to the fact that many images in our dataset contained one larger, central lesion surrounded by other smaller peripheral lesions, it was necessary after applying these transformations to apply a heuristic to identify the primary lesion contour within each image. This is because in order to extract features from these images, we must identify only one lesion that we wish to make the object of our computations. The heuristic that we developed to identify this primary lesion works as follows: First, discard any identified lesions (fully enclosed regions with value 255 in the binary segmentation image) that have small areas (< 300 pixels), or are greater than 200 pixels from the center of the image (in terms of Euclidean distance). Next, if there exist lesion contours close to the center of the image (< 150 pixels away), sort these contours and select the largest one as the primary contour. If there do not exist any contours that are close to the center of the image, we sort the remaining contours by their distance to the center and select the one that is closest to the center as the primary lesion. We validated that this heuristic was effective experimentally using visual inspection.

Initially, we experienced difficulty segmenting lesions in over 17% of images in the dataset that had black backgrounds, as depicted in Figure 5. Segmentations could either not be identified for these images, or were identified as the border between the image's black background and the patient's skin. In order to fix this, we modified the segmentation algorithm specifically for these images with black backgrounds. First, we identified images with black backgrounds by computing whether the four corner pixels were all pure black or close to pure black. For those images that were identified as having black backgrounds, we replaced all pixels with values < 10 with the median pixel value from the remainder of the image, and applied the segmentation algorithm to the resulting image. To account for the case where this process still fails to identify a correct segmentation, we add specifications into the heuristic that discards large lesions (area $>$ 50,000 pixels) that are almost perfectly centered in images with black backgrounds. This helps to eliminate the case where the primary lesion is identified as the border between the black background and the patient's skin. Figure 5 shows the result of applying our segmentation algorithm to images with black backgrounds before and after making these modifications.

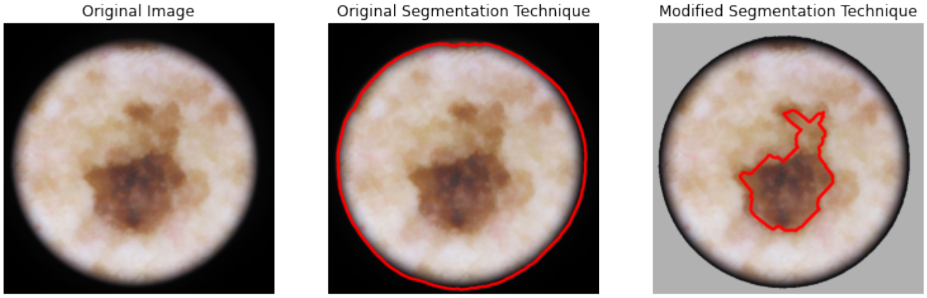

Original Image
Original Segmentation Technique
Modified Segmentation Technique

Figure 5: Sample lesion with black background. Before making the described modifications to the segmentation algorithm, the identified lesion was the border between the black background and the patient's skin. After making the mentioned modifications, the lesion was properly segmented.

We applied this algorithm to all preprocessed lesion images and evaluated its success based on visual inspection, as our dataset lacked expert segmentations to use as ground-truths. Some results of applying the algorithm are depicted in Figure 6.

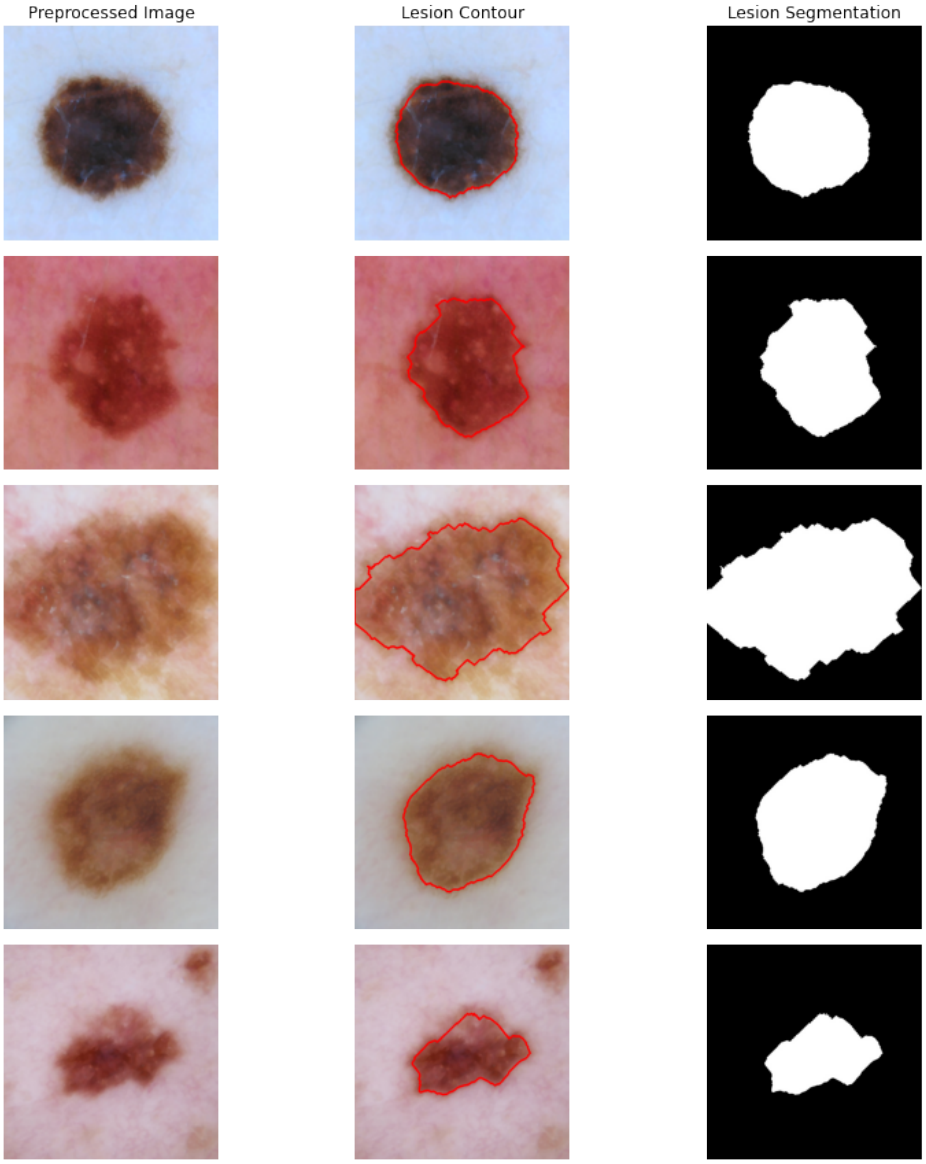


Figure 6: Example lesion segmentations.

4.3. ABCD Feature Engineering

After creating segmentations for each lesion, we extracted a variety of features from lesion images based on the ABCD rule of skin cancer. That is, we attempted to measure A: how asymmetric a given lesion was in terms of shape and texture, B: the irregularity of lesions' borders, C: the occurrence of certain colors and color variation within lesions, and D: the diameters of given lesions.

4.4. Asymmetry (A) Features

First, we compute asymmetry features that quantitatively measure lesion asymmetry both in terms of shape and in terms of texture by adopting the asymmetry feature engineering approach proposed by Toureau et. al. in their paper *Automatic Detection of Symmetry in Dermoscopic Images Based on Shape and Texture*[17], as described earlier in Section 2. We compute these scores by making slight modifications to the paper's accompanying code published on the website OpenDemo as a Python package dermoscopic_symmetry [26]. According to this proposed method, we compute both shape-based and texture-based asymmetry scores as follows:

4.4.1. Shape-Symmetry

To compute shape-based asymmetry, axes of symmetry are first computed for each lesion as orthogonal lines through their centroids that split them into two regions of approximately equal area. To assess whether an axis of symmetry divides a lesion symmetrically, the Jaccard index, a statistic for gauging the similarity of sample sets, is employed. A line is considered to be a perfect axis of symmetry if the second divided region is "equal" to the reflection of the first divided region with respect to the axis. Equality in this case refers to whether a pair of corresponding pixels in each half of the lesion are both tagged as lesion or both tagged as skin. To define the shape-based symmetry index of this line, first let l be the line and M_+, M_- be the two halves of the image on opposite sides of l. Then let $R_l(M)$ denote the reflection of half M with respect to l and let $|A|$ to be the area of region A. Then the shape based symmetry index ($S_s(l)$) of line l is defined by Equation 1, which is based on the Jaccard index $J(A,B) = \frac{|A \cap B|}{|A \cup B|}$. A $S_s(l)$ score of 1 indicates complete overlap between the two halves of the divided lesion, and a score of 0 indicates no overlap between the two halves of the divided lesion.

$$S_s(l) = \frac{|M_+ \cap R_l(M_-)|}{|M_+ \cup R_l(M_-)|} \tag{1}$$

The final asymmetry score is computed by considering N axes drawn through the centroid of the lesion by rotating the image $\frac{180}{N}$ times (in our case, N=20), and computing the shape-based symmetry index $S_s(l)$ for each rotation. Finally, a Random Forest Classifier aggregates all of these scores into a final shape symmetry score, which is defined in Table 1. This final score (denoted shape_symmetry) is either 0: fully symmetric with respect to two orthogonal axes of symmetry, 1: symmetric with respect to 1 axis of symmetry, or 2: not symmetric with respect to any axes. Examples of each of these cases are depicted in Figure 7. Additionally, a lesion receives an asymmetry score of -1 if it's shape_symmetry could not be computed. These lesions are presumed to be asymmetric.

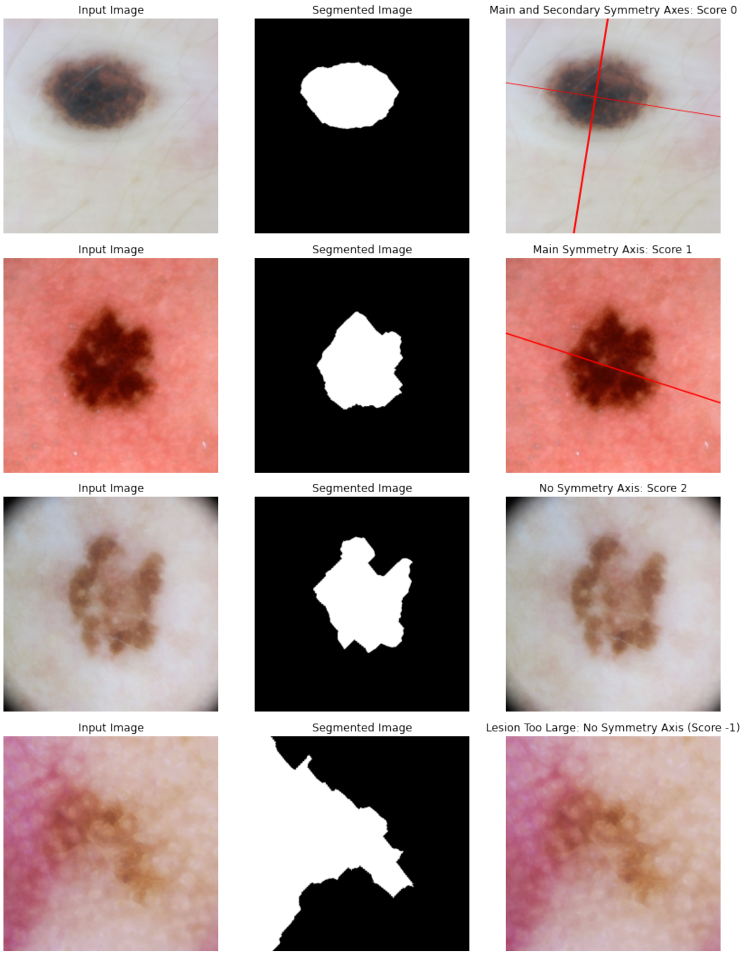

Input Image
Segmented Image
Main and Secondary Symmetry Axes: Score 0
Input Image
Segmented Image
Main Symmetry Axis: Score 1
Input Image
Segmented Image
No Symmetry Axis: Score 2
Input Image
Segmented Image
Lesion Too Large: No Symmetry Axis (Score -1)

Figure 7: Lesion axes of symmetry. The first row is a lesion with a shape symmetry score of 0 and is symmetric with respect to 2 orthogonal axes of symmetry. The second row is a lesion with a shape symmetry score of 1 and is symmetric with respect to 1 main axis of symmetry. The third row is a lesion with a shape symmetry score of 0 and is not symmetric with respect to any axis. The final row is a lesion whose symmetry score could not be computed, but is presumed asymmetric and receives a score of -1.

Asymmetry Score	Description
0	Fully symmetric lesion with respect to two orthogonal axes of symmetry
1	Symmetric with respect to one axis of symmetry
2	Not symmetric with respect to any axis
-1	Symmetry score could not be computed. Lesion presumed asymmetric

Table 1: Asymmetry scores used for both shape-symmetry and texture-symmetry.

4.4.2. Texture-Symmetry

According to the ABCD rule, a lesion's symmetry is jointly based on its shape and on the appearance of similar structures and patterns within it [6]. As such, it is important to create a feature that takes into account a lesion's texture-based symmetry. We employ a patch-based approach to computing texture-based symmetry as follows: First, compute the same axes of symmetry as when computing shape-based symmetry. Then, randomly sample 32×32 pixel patches from each side of the axis of symmetry, and compute whether or not those patches represent the same textures.

This similarity of texture is predicted using a Random Forest model trained on five different texture features extracted from the gray level co-occurrence matrix (GLCM) of each patch with two-pixel distance and a horizontal orientation. These five texture features are dissimilarity, correlation, energy, contrast, and homogeneity. Correlation is a measure of the linear dependence of gray levels between pixels at the specified distance. Energy measures the brightness of the images and the repetition of subunits. Contrast refers to the local gray level variations, and homogeneity is a measure of the smoothness of the gray level distribution. The patches used to train the Random Forest were randomly selected from expert-segmented lesions in the PH2 database mentioned in Toureau et. al.[17], with tagged "similar" patches both extracted from within a lesion, and tagged "different" patches being extracted one from within the lesion, and one from the outside skin. This Random Forest model (which we will denote by T) predicts 1 if the two randomly sampled patches from each half of the lesion (denoted by p_+ and p_-) represent the same texture, and 0 if the two patches represent a different texture. We compute a texture-symmetry index based on these patch similarities (by sampling as many partially overlapping $n \times n$ patches as possible from within each lesion) according to Equation 2. This index equals 1 if all patches are predicted to be "similar" and

0 if all patches are predicted to be "different." Next, the image is again rotated $\frac{180}{N}$ times (N=20), and we compute the texture-based symmetry index for each rotation. Finally, another Random Forest Classifier aggregates all of these indexes into a final texture symmetry score (denoted by texture_symmetry), which takes on the same values as the shape symmetry scores as described in Table 1. Examples of lesions and their computed texture symmetry scores are depicted in Figure 8.

$$S_t(l) = \frac{1}{N}\sum_{i=1}^{N} T(p_+^i, p_-^i) \tag{2}$$

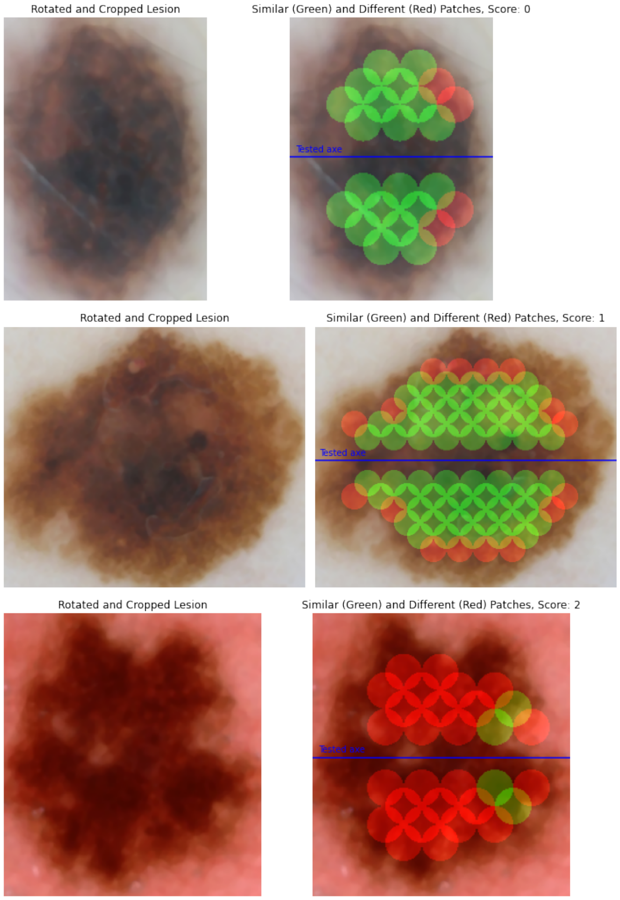

Rotated and Cropped Lesion
Similar (Green) and Different (Red) Patches, Score: 0
Tested axe
Rotated and Cropped Lesion
Similar (Green) and Different (Red) Patches, Score: 1
Tested axe
Rotated and Cropped Lesion
Similar (Green) and Different (Red) Patches, Score: 2
Tested axe

Figure 8: Texture symmetry computation. This figure shows the results of applying texture symmetry computations to various skin lesions. Images in the first column are rotated and cropped lesions. Images in the second column show the main axis of symmetry that divides a lesion, and symmetric 32×32 patches on opposite sides of the axis. Patches are displayed as green if the random forest model T predicts them to represent the same texture, and are red if it predicts them to represent different textures. From top to bottom, these lesions have final texture-symmetry scores of 0, 1, and 2 respectively.

In addition to saving these final texture symmetry scores as features to use in our classification model, we also save the mean dissimilarity, correlation, energy, and contrast over all patches for a given lesion as features to be used in the classification model.

4.5. Border (B) Features

Another defining feature of malignant lesions is that they often have irregular, poorly defined borders. We therefore seek to develop a feature that captures this border irregularity. We use a metric known as the Polsby-Popper compactness score, which is described by Equation 3, to compute this irregularity [27]. In this equation, L is a given lesion, $A(L)$ is the area of lesion L, $P(L)$ is the perimeter of lesion L, and $PP(L) \in [0,1]$ is a score representing the compactness of a lesion's border, where a score of 1 indicates a perfectly compact border and a score of 0 indicates a complete lack of compactness. Examples of two lesions - one with a high $PP(L)$, and the other with a low $PP(L)$ - are shown in Figure 9. The area and perimeter of lesions were computed using OpenCV [21].

$$PP(L) = \frac{4\pi A(L)}{P(L)^2} \tag{3}$$

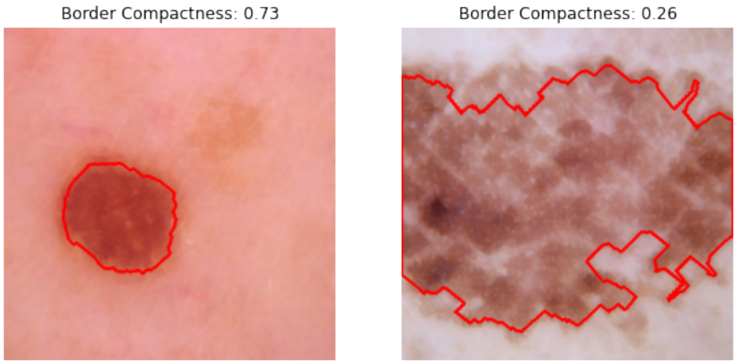


Figure 9: Border feature examples. The lesion on the left has a $PP(L)$ (Border Compactness) close to 1, and its border appears very compact, close to a perfect circle. The lesion on the right has a $PP(L)$ (Border Compactness) closer to 0, and its border appears highly non-compact and irregular.

This metric was initially developed to quantify the degree of gerrymandering in political districts - with a non-compact district corresponding to a district that has been heavily gerrymandered - and is now used widely to measure compactness of shapes more generally. We compute this Polsby-Popper score (which we call border_compactness) for each lesion, and use it as the "B" feature in our final model. We anticipate that lesions that have more compact borders will more frequently correspond to benign lesions, and that lesions with less compact borders will more frequently correspond to melanoma.

4.6. Color (C) Features

According to the ABCD rule, another one of the most important features to account for in classifying a lesion is its color irregularity. Specifically, dermatologists claim that lesions that vary significantly in their color profile and that contain certain irregular colors are more likely to be melanoma. There have been many features proposed in prior works that have been used to measure these characteristics quantitatively, ranging from those as simple as measuring the mean color within a lesion to those as complex as using K-Means Clustering to identify the dominant K colors within a lesion [28]. The features that we choose to focus on in our research all relate to variegated coloring (a lesion comprised of multiple unique colors is said to have variegated coloring). The most common way to quantify color variegation is to count the occurrence of certain colors within a lesion. According to the ABCD rule, there are six colors that it is important to account for in a given lesion: black, white, red, dark brown, light brown, and blue/gray [28]. We therefore seek to identify which of these colors accounts for at least 1% of pixels in a given lesion. We define these colors according to the RGB color ranges defined by El Abbadi et. al. [11], which are displayed in Table 2. We create indicator features for each of these six colors where, for instance, the feature "black" takes on the value 1 if the color black accounts for at least 1% of pixels in a lesion. Additionally, we define num_colors as a feature which captures the total number of these unique colors that comprise at least 1% of pixels in a given lesion. Examples of lesions and the corresponding colors identified in them are depicted in Figure 10.

In addition to detecting the presence of certain colors within a lesion, we also engineer a set of features that utilize color histograms to measure the difference between the color profiles of a given

Color	RGB Range	Examples
Black	$R \leq 62, G \leq 52, B \leq 52$	⬛
White	$R \geq 205, G \geq 205, B \geq 205$	⬜
Red	$R \geq 150, G < 52, B < 52$	🟥
Dark Brown	$62 < R < 150, 0 < G < 100, 0 < B < 100$	🟫
Light Brown	$150 \leq R \leq 240, 50 < G \leq 150, 0 < B \leq 100$	🟧
Blue Gray	$0 \leq R \leq 150, 100 \leq G \leq 125, 125 \leq B \leq 150$	🟦

Table 2: Color ranges for the six colors to identify within each lesion.

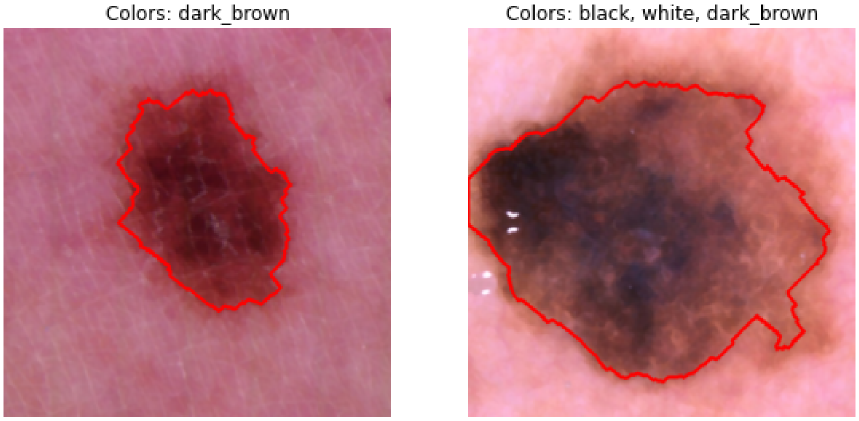


Figure 10: Color occurrence feature examples. The lesion on the left has num_colors=1, because at least 1% of it's pixels are dark brown. The lesion on the right has num_colors=3, because at least 1% of it's pixels are black, white, and dark brown.

lesion and the "normal" skin that surrounds it. A color histogram represents the distribution of the composition of colors in an image by quantizing a color space into various bins and counting the frequency of pixels belonging to each bin [28]. For each image, we therefore seek to model the distribution of the RGB channels of both lesions and the normal skin that surrounds them using 3D color histograms with 8 bins for each channel. To accomplish this, we use functionality provided by OpenCV to compute a 3D histogram of all pixels within each segmented lesion, and a 3D histogram of all pixels outside of each segmented lesion (excluding pure black pixels, in the case where an image has a black background)[21]. We use functionality provided by OpenCV to compute two metrics about the distance between these 3D histograms, using OpenCV's compareHist function [21]. Before computing these metrics, it is necessary to normalize both histograms to be probability distributions, where all bins sum to 1.

The first metric we compute is the intersection between the lesion and skin histograms, which is calculated according to Equation 4. Here c corresponds to a color channel (R, G, or B), $H_1(c)$ is the 1D lesion histogram for channel c, and $H_2(c)$ is the 1D normal skin histogram for channel c. The term $min(H_1(c), H_2(c))$ calculates the intersection of the two color distributions for channel c, with possible intersection values ranging between 0 (meaning no overlap) and 1 (meaning identical distributions). We sum over all color channels to yield a final intersection score, which we denote by color_intersect.

$$int(H_1, H_2) = \sum_c min(H_1(c), H_2(c)) \tag{4}$$

The second metric we compute is the chi-squared distance between the lesion and skin histograms (denoted by color_chisq), which is calculated according to Equation 5 (all variables have the same meanings as in Equation 4). Histograms with low chi-squared distance scores (close to 0) represent similar color distributions, while histograms with higher chi-squared distance scores represent dissimilar color distributions. In the context of this research, higher chi-squared distance scores are those considerably above the median computed chi-squared distance score of ~50. Examples of skin lesions and their corresponding color histograms, as well as their color_intersect and color_chisq scores are displayed in Figure 11.

$$chi(H_1, H_2) = \sum_c \frac{(H_1(c) - H_2(c))^2}{H_1(c)} \tag{5}$$

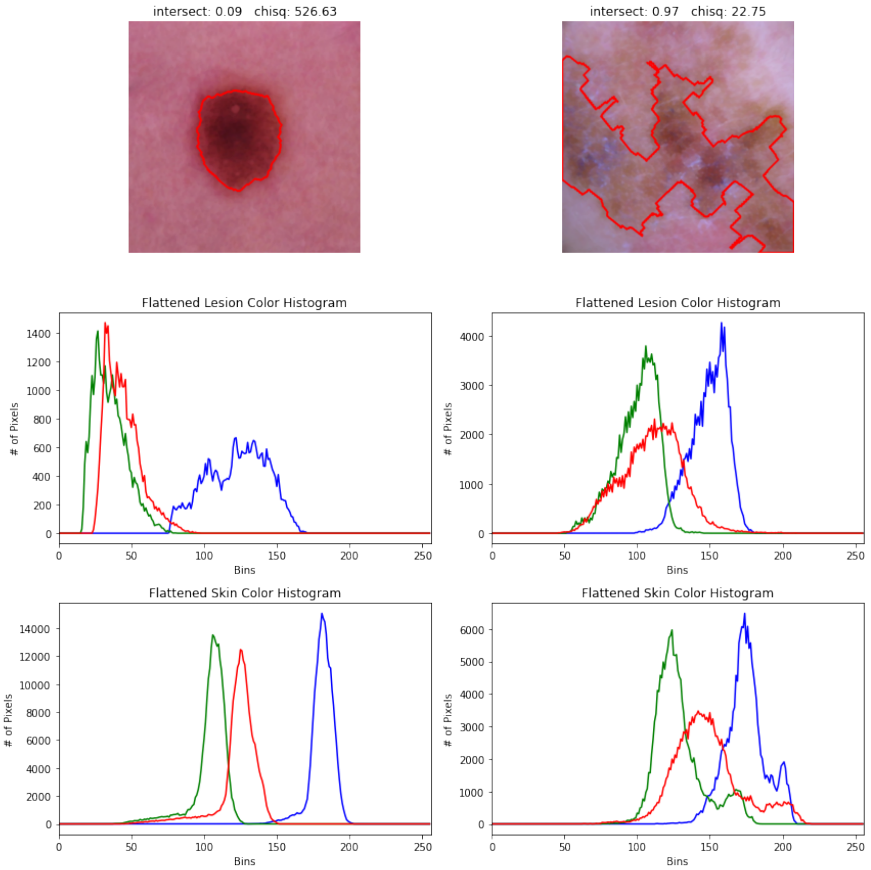


Figure 11: Color histogram feature examples. The lesion on the left has a low intersection score and high chi-squared distance score, indicating that it has quite a different color distribution than the surrounding skin. This is further illustrated by the clear differences between the lesion color histogram and skin color histogram below, which appear to represent entirely different distributions. The lesion on the right has a high intersection score and low chi-squared distance score, indicating that it has a similar color distribution to the surrounding skin. This is clearly illustrated by the similar appearing color distributions of the lesion and skin. We can expect images of this nature where the segmentation algorithm struggles to find a clear border between the lesion and surrounding skin to have high intersection and low chi-squared distance scores.

4.7. Diameter (D) Features

According to the ABCD rule, skin lesions with diameters greater than 6 millimeters are more likely to be melanoma. Unfortunately, our dataset contained no accurate indication of the actual physical size of lesions. As such, in order to still maintain some representation of a lesion's size in our feature space, we instead estimate each lesion's diameter in pixels using its area, computed with OpenCV[21]. While we do not know if the images in the dataset were all taken at precisely the same scale, due to the fact that each image was taken with a dermoscope, it is reasonable to assume that the image scales are consistent enough for these estimated diameters to provide some meaningful notion of a lesion's size. We use this diameter feature as our stand-in "D" feature in our classification model.

Finally, all engineered ABCD features used to train our model are presented in the Appendix in Table 4.

5. Evaluation and Results

5.1. Classification Model

To complete our non deep-learning CAD system, we trained a variety of classification models on our extracted ABCD features in order to experimentally determine which model would perform optimally at classifying skin lesions. To evaluate the performance of these classification models we compute the classifiers' accuracy on test data, as this seems to be a reasonable performance metric given the fact that our dataset consists of balanced classes (50% of images are melanoma, 50% of images are nevus). We experimented with the following three classification models - Logistic Regression, Linear SVMs, and Random Forest Classifiers (all using the Python package scikit-learn[29]) - and will explain each in detail below.

In training both the Logistic Regression model and the Linear SVM, we divide our dataset into 80% training/validation data and 20% test data. We hyperparameter tune both of these models using 4-fold cross validation on the training data. For the Logistic Regression model, we vary the regularization parameter (using L2 regularization) and whether or not an intercept should be added to the decision function. For the Linear SVM, we again vary this same regularization parameter, in addition the kernel type to be used (linear, polynomial, or radial basis function). The optimal logistic regression model uses a regularization parameter of 1 and includes an intercept in the decision function, while the optimal Linear SVM also uses a regularization parameter of 1, in addition to a linear kernel.

In training the Random Forest Classifier, we also divide our dataset into 80% training/validation data and 20% test data. We hyperparameter tune this model using the randomized search algorithm (RandomizedSearchCV in scikit-learn [29]), a hyperparameter tuning method that randomly samples hyperparameters from a defined search space of values n times, and runs K-fold cross validation on a classification model using each set of sampled hyperparameters [30]. In our research, we define a hyperparameter sample space that varies the following hyperparameters: the number of trees in the Random Forest, the maximum depth of a tree, the minimum number of samples required to split an internal node, and the minimum number of samples required to be at a leaf node. We sample $n = 50$ different hyperparameter combinations from this sample space, and train each model using 4-fold cross validation on the training set. We select the optimal Random Forest model as the model containing the combination of hyperparameters that yields the highest average validation accuracy. This model used 175 trees in the Random Forest, had a maximum tree depth of 12, required 2 samples to split an internal node, and required a minimum of 2 samples per leaf node.

The performances of all of these attempted classification models were evaluated on a test set of 2,036 images, and are displayed in Table 3. As shown in this table, models were trained both exclusively using the extracted ABCD features, in addition to using the ABCD features as well as patient-level metadata (described in Table 4 in the Appendix). The results of these classification models clearly show that the Random Forest Classifier has the best performance, as it achieves a test accuracy of 81.9% (including metadata) as compared to accuracies of 77.6% and 77.1% with the Logistic Regression model and Linear SVM respectively. As such, we conduct our analysis of feature importance using this optimal Random Forest model (without metadata, as for the purpose of this research we care about analyzing the importance of ABCD features).

Model	**Test Accuracy (no metadata)**	**Test Accuracy (metadata)**
Logistic Regression	74.0%	77.6%
Linear SVM	73.5%	77.1%
Random Forest	78.1%	81.9%

Table 3: Performance of various classification models training with and without patient metadata.

5.2. Feature Importance Analysis

Finally, after developing a complete non deep-learning CAD system that is capable of classifying skin lesions with greater than 80% accuracy, we seek to conduct analysis of feature importance in order to determine which ABCD features are most influential towards classifying skin lesions, and to identify whether the feature effects determined by our model align with the clinical intuition behind the ABCD rule. We thus divide the analysis of feature importance into three sections: feature importance analysis of individual features, analysis of isolated feature class performance, and analysis of feature effects and clinical intuition.

5.2.1. Feature Importance of Optimal Random Forest Model

Firstly, we conduct feature importance analysis of individual features using both permutation importance and sequential feature selection to determine how much each individual ABCD feature (listed in Table 4 in the Appendix) contributes to the final model. Permutation importance is a metric that calculates the importance of a given feature as the decrease in a model's accuracy when the values of that feature are randomly shuffled [31]. This procedure breaks the relationship between the feature and the target, and thus the drop in model accuracy is indicative of how much the model depends on the feature. Formally, this score (which is also known as Mean Decrease in Accuracy) is computed for each feature as follows [31]:

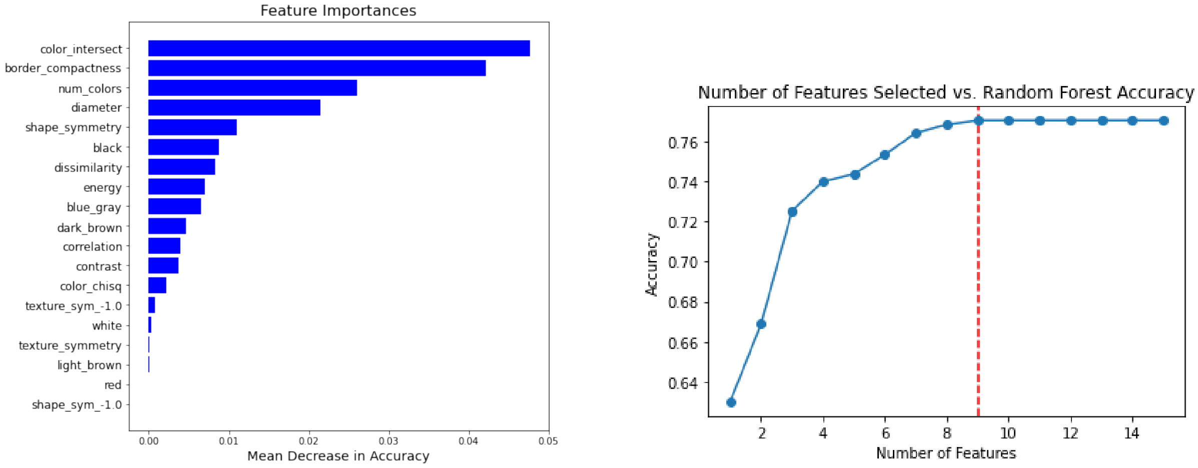
1. For model m and testing dataset D: Compute the reference accuracy s of the model m on data d (in our case, the accuracy of our Random Forest on the test set)
2. For each feature f_j:
   (a) For each repetition k in 1, ..., K (in our case $K = 10$):

   Randomly shuffle column j of dataset D to generate a corrupted version of the data $\tilde{D}_{k,j}$

   Compute the score $s_{k,j}$ of model m on corrupted data $\tilde{D}_{k,j}$

   (b) Compute importance i_j of feature f_j by the equation: $i_j = s - \frac{1}{K}\sum_{k=1}^{K} s_{k,j}$

As illustrated in Figure 12 (a), the three most important features in our optimal Random Forest model based on permutation importance were color_intersect, border_compactness, and num_colors, which had permutation importances of 0.048, 0.042, and 0.026 respectively. This means that for



(a) Permutation importance of features in Random Forest model

(b) Accuracy of Random Forest model using sequential feature selection to add features

Figure 12: Random Forest Feature Importance

instance, when color_intersect was randomly shuffled $K = 10$ times, the Random Forest model's accuracy was 4.8% worse on average. We recall that color_intersect is a Color feature that computes the intersection between the 3D histograms of pixels within a lesion and outside of a lesion, border_intersect is our only Border feature that computes the compactness of a lesion's border, and num_colors is a Color feature that computes the number of unique colors (as defined in Table 2) that account for at least 1% of all pixels in a given lesion.

After computing these permutation importance scores, we additionally performed sequential forward feature selection on our Random Forest model in order to determine the most descriptive features in the model, and in order to determine what subset of features could be trained on to yield nearly the same accuracy as the original model. Sequential forward feature selection is an algorithm that initializes an empty feature subset X_k and at each iteration, adds an additional feature, x^+, to this feature subset [32]. Here, x^+ is the feature that maximizes the accuracy of the Random Forest model trained only on X_k. We run this algorithm over all features in our dataset, and display the accuracy of our Random Forest model trained on each feature subset in Figure 12 (b). The first three features added here are again color_intersect, border_compactness, and num_colors, which yield an accuracy of ~73% on their own, again demonstrating that these are the three most descriptive features in our model.

5.2.2. Performance of Feature Classes in Isolation

Secondly, we train a Random Forest model on each individual class of features (A, B, C, and D) to gain an experimental understanding of which feature class performs best at classifying skin lesions in isolation. In order to train these Random Forests, we use the same hyperparameter tuning and cross validation methods used in Subsection 5.1 to find the best model for a given class of features. As illustrated in Figure 13, the Color features perform best at classifying lesions on their own, achieving an accuracy of 73.5% without metadata. This is followed by Asymmetry features, Border features, and Diameter features, which achieve accuracies of 68.2%, 64.0%, and 59.3% respectively. While we cannot make the generalization that Color features are most useful in all cases at classifying melanoma (particularly due to the fact that in our research, each feature class consists of a different number of individual features), this analysis does demonstrate that using our feature engineering and classification methods, Color features are the most informative class of features in isolation.

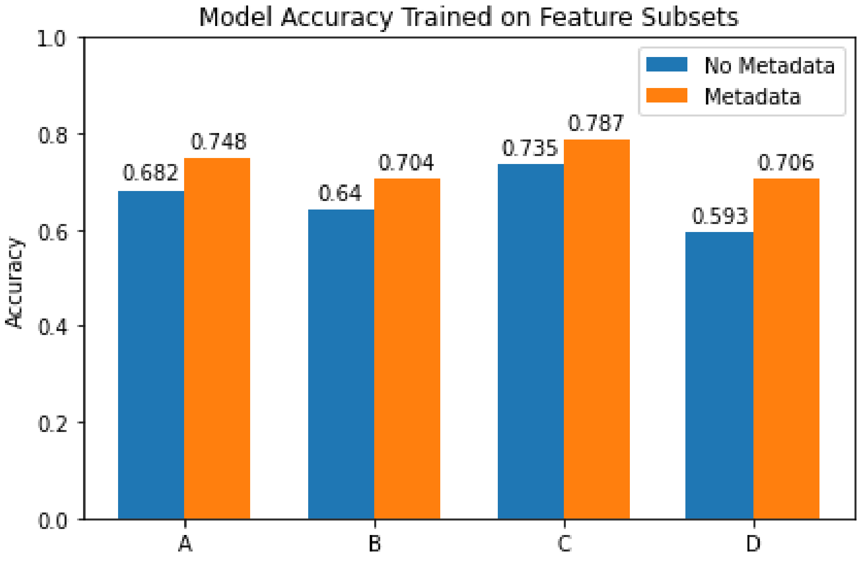


Figure 13: Analysis of how feature classes (A: Asymmetry, B: Border, C: Color, D: Diameter) perform in isolation at classifying melanoma.

5.2.3. SHAP Value Feature Importance + Feature Effects

Finally, we use another feature importance metric known as SHAP values - which was originally proposed by Lundberg and Lee in 2017 [33] - in order to compute the contribution of each feature to our model's output, and to determine the effect of each feature on our model's predictions. The details of computing SHAP values are explained elaborately in Lundberg and Lee's 2017 paper *A Unified Approach to Interpreting Model Predictions*, but to summarize briefly, SHAP values break down individual predictions to show the impact of each feature on model output. They do this by comparing the prediction a model makes when a given feature takes its given value to the prediction that model would make if the given feature took some baseline value. That is, the SHAP values of all features sum up to explain why a model's prediction was different from the baseline for a given sample. Thus, SHAP values allow us to decompose model's predictions into graphs like those shown in Figure 14, computed using the Python package SHAP [34].

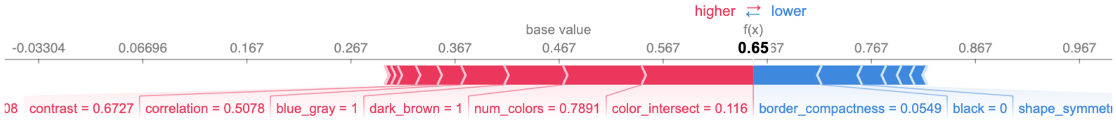


Figure 14: SHAP values for one melanoma sample. Here the features color_intersect and num_colors largely push the model to predict melanoma (1) rather than nevus (0).

By aggregating these SHAP values across all predictions in our test set, we yield interesting insights into feature importance not provided by permutation importance. SHAP summary plots - plots that combines feature importance with feature effects - provide us with this alternative to permutation importance, as depicted in Figure 15 [35]. These plots are again computed using the SHAP Python package. This plot is comprised of many dots, each with three characteristics: 1) Vertical location shows what feature a dot depicts, 2) Color shows whether that feature had a high or low value, 3) Horizontal location depicts whether the effect of the feature taking a given value caused a higher or lower prediction. For example, the blue point in the upper left shows that a low value of color_intersect contributed to the model predicting 0 (nevus) for that given sample. Additionally, features are sorted vertically by their feature importances, where feature importance here is computed as the sum of the magnitudes of a feature's SHAP values over all samples in the test set.

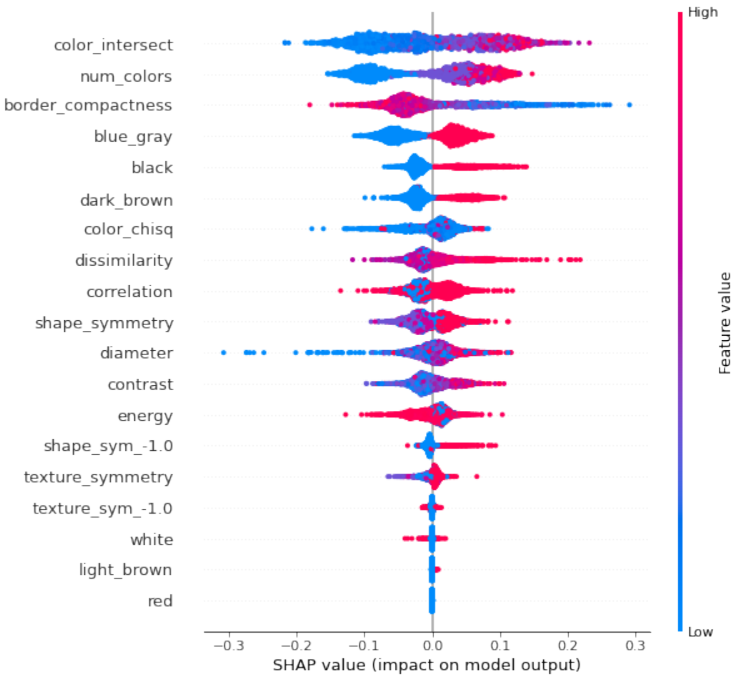


Figure 15: SHAP summary plot for our test dataset. Again, color_intersect, num_colors, and border_compactness were the three most important features.

We can finally interpret this plot of aggregated SHAP values to gain more insight into feature importance in our model, and to analyze whether the effects of our extracted features align with clinical intuition according to the ABCD rule. According to features' SHAP values, the most important features are consistent with those calculated using permutation importance, as here the three most informative features in our model are again color_intersect, num_colors, and border_compactness. According to Figure 15, we also see that the effects of the majority of our extracted features on our model's predictions largely correspond with clinical intuition. For Asymmetry features, high shape_symmetry scores and texture_symmetry scores contribute to our model predicting melanoma. This makes sense, considering we expect asymmetric lesions (those with scores of 1 or 2) to more frequently be malignant. For Border features, low border_compactness scores correspond to our model predicting melanoma, and high border_compactness scores correspond to our model predicting nevus. This is again intuitive, as we expect lesions with more irregular borders (lower border_compactness scores) to more frequently be melanoma. For Color features, as expected, high values of num_colors, in addition to the occurrence of blue_gray, black, and dark_brown correspond to our model predicting melanoma (white, red, and light_brown were fairly irrelevant features). The one feature that seems to contradict clinical intuition is color_intersect. According to the ABCD rule, we would expect lesions with color profiles highly distinct from their surrounding skin to be indicative of melanoma. However, according to our analysis of SHAP values, high values of color_intersect (meaning lesions where 3D histograms of lesion pixels are similar to 3D histograms of surrounding skin pixels) correspond to our model predicting melanoma. It seems likely that this could be due to the fact that many images of melanoma (such as those shown in the right lesion of Figure 9) consisted of lesions that essentially covered the entire image. In many of these cases, segmentations only captured part of a lesion, and the color of the entire image thus more consistently shared a similar color structure. Lastly, for Diameter features, lesions with higher diameters corresponded to our model predicting melanoma, as we would expect according to the ABCD rule.

6. Conclusions

Based on the initial goals laid out for this research, we come to two core sets of conclusions:

Firstly, we manage to develop a complete non deep-learning CAD system based around the ABCD rule of dermoscopy that classifies skin lesions with an accuracy of 81.9% on a test set of $> 2,000$ dermoscopic skin lesion images. Although our optimal Random Forest model's accuracy is slightly lower than those of previously proposed non deep-learning CAD systems, as many of these works achieved accuracies $> 90\%$, the methodologies we employ are inherently more generalizable particularly due to our usage of a significantly larger dataset and a classification model that is not rule-based. For those previous non deep-learning CAD systems utilizing the ABCD rule that do have access to larger datasets and seem likely to generalize (namely Pham et. al. [14], which achieves 74.75% accuracy), our proposed system yields improved accuracy on this task. Thus, we are confident that we have developed a non deep-learning CAD system that is useful and will accurately classify skin lesions in additional dermoscopic skin images.

Secondly, we experimentally analyzed the ABCD rule to determine which features were most influential in diagnosing melanoma based on our model. In terms of individual features, the intersection between 3D histograms of pixels within and outside of each lesion, the number of unique colors appearing within a lesion, and the irregularity of a lesion's border were the most informative in determining a lesion's diagnosis (based on permutation importance and SHAP values). In terms of feature classes, Color features are largely the most influential in determining a lesion's diagnosis according to our model. This is evident both due to the fact that color features performed best in isolation at classifying skin lesions, and due to the fact that according to both permutation importance and SHAP values, two of the three most informative individual features were based on color. In fact, in the case of SHAP values, six of the seven most informative individual features were based on color. We further use our analysis of SHAP values to demonstrate that the ABCD features we extracted have effects in our model that are largely consistent with the clinical intuition behind them.

7. Future Work

One caveat to note from our research is that according to the ABCD rule, Asymmetry is the attribute that contributes the most to a lesion's final diagnosis [6]. In our research, asymmetry features appeared to be largely less informative than anticipated. Although Asymmetry was the second most informative feature class in isolation, shape_symmetry was only the fifth most informative feature according to permutation importance, and texture_symmetry was essentially non-informative. We hypothesize that this is due to the fact that the asymmetry features we develop are entirely reliant on the accuracy of lesion segmentations. This is intensified by the fact that our feature engineering method is largely borrowed from Toureau et. al.'s paper [17], which computes asymmetry scores using expert lesion segmentations on a dataset of 200 dermoscopic images. Thus applying this same method to automatically computed segmentations could have significantly decreased the effectiveness of asymmetry features in our research.

Therefore, one clear avenue for future work in this domain is to use a state of the art lesion segmentation algorithm (such as the one proposed in *Skin Lesion Segmentation with improved*

Convolutional Neural Network [24]) to segment lesions. This would ideally make asymmetry features more accurate and informative, and would generally increase the reliability of nearly all ABCD features, as all of the ABCD features we extracted rely in some way on accurate lesion segmentations.

One last caveat to address is that, as noted in Subsection 4.7, we used pixels as a proxy for lesion size without knowing if all dermoscopic images were captured at the same scale. Therefore, in future works it would be useful to obtain datasets containing calibrated scale measurements to ensure that Diameter features are as informative as possible.

References

[1] Key statistics for melanoma skin cancer. URL: https://www.cancer.org/cancer/melanoma-skin-cancer/about/key-statistics.html.

[2] Melanoma overview. URL: https://www.skincancer.org/skin-cancer-information/melanoma/.

[3] American cancer society: Melanoma skin cancer. URL: https://www.cancer.org/acs/groups/cid/documents/webcontent/003120-pdf.

[4] Signs and symptoms of melanoma: What you should look for. URL: https://miiskin.com/melanoma/symptoms-signs/.

[5] Holt; PE Menzies; SW Vestergaard; ME, Macaskill; P. Dermoscopy compared with naked eye examination for the diagnosis of primary melanoma: a meta-analysis of studies performed in a clinical setting. *British Journal of Dermatology*, 159, 08 2008. doi:https://doi.org/10.1111/j.1365-2133.2008.08713.x.

[6] Franz Nachbar, Wilhelm Stolz, Tanja Merkle, Armand B. Cognetta, Thomas Vogt, Michael Landthaler, Peter Bilek, Otto Braun-Falco, and Gerd Plewig. The abcd rule of dermatoscopy: High prospective value in the diagnosis of doubtful melanocytic skin lesions. *Journal of the American Academy of Dermatology*, 30(4):551–559, 1994. URL: https://www.sciencedirect.com/science/article/pii/S0190962294700613, doi:https://doi.org/10.1016/S0190-9622(94)70061-3.

[7] Andre Esteva, Brett Kuprel, Roberto A. Novoa, Justin Ko, Susan M. Swetter, Helen M. Blau, and Sebastian Thrun. Dermatologist-level classification of skin cancer with deep neural networks. *Nature*, 542(7639):115–118, February 2017. URL: http://www.nature.com/articles/nature21056, doi:10.1038/nature21056.

[8] International Skin Imaging Collaboration. SIIM-ISIC 2020 Challenge Dataset. *International Skin Imaging Collaboration*, 2020. URL: https://doi.org/10.34970/2020-ds01.

[9] Qishen Ha, Bo Liu, and Fuxu Liu. Identifying Melanoma Images using EfficientNet Ensemble: Winning Solution to the SIIM-ISIC Melanoma Classification Challenge. *arXiv e-prints*, page arXiv:2010.05351, October 2020. arXiv:2010.05351.

[10] Margarida Ruela, Catarina Barata, and Jorge S. Marques. What is the role of color symmetry in the detection of melanomas? pages 1–10, 2013.

[11] Nidhal El abbadi and Zahraa Faisal. Detection and analysis of skin cancer from skin lesions. *International Journal of Applied Engineering Research*, 12:9046–9052, 01 2017.

[12] Reda Kasmi and Karim Mokrani. Classification of malignant melanoma and benign skin lesions: implementation of automatic abcd rule. *IET Image Processing*, 10(6):448–455, 2016. arXiv:https://ietresearch.onlinelibrary.wiley.com/doi/pdf/10.1049/iet-ipr.2015.0385, doi:https://doi.org/10.1049/iet-ipr.2015.0385.

[13] A. Murugan, S.Anu H. Nair, and K. P. Sanal Kumar. Detection of Skin Cancer Using SVM, Random Forest and kNN Classifiers. *Journal of Medical Systems*, 43(8):269, August 2019. URL: http://link.springer.com/10.1007/s10916-019-1400-8, doi:10.1007/s10916-019-1400-8.

[14] T. C. Pham, G. S. Tran, T. P. Nghiem, A. Doucet, C. M. Luong, and V. Hoang. A comparative study for classification of skin cancer. In *2019 International Conference on System Science and Engineering (ICSSE)*, pages 267–272, July 2019. doi:10.1109/ICSSE.2019.8823124.

[15] Marcos Almeida and Iury Santos. Classification models for skin tumor detection using texture analysis in medical images. *Journal of Imaging*, 6:51, 06 2020. doi:10.3390/jimaging6060051.

[16] T. Mendonça, P. M. Ferreira, J. S. Marques, A. R. S. Marcal, and J. Rozeira. Ph2 - a dermoscopic image database for research and benchmarking. pages 5437–5440, 2013. doi:10.1109/EMBC.2013.6610779.

[17] Vincent Toureau, Pedro Bibiloni, Lidia Talavera-Martínez, and Manuel González-Hidalgo. Automatic detection of symmetry in dermoscopic images based on shape and texture. pages 625–636, 2020.

[18] Philipp Tschandl, Cliff Rosendahl, and Harald Kittler. The HAM10000 dataset, a large collection of multi-source dermatoscopic images of common pigmented skin lesions. *Scientific Data*, 5(1):180161, December 2018. URL: http://www.nature.com/articles/sdata2018161, doi:10.1038/sdata.2018.161.

[19] Noel C. F. Codella, David Gutman, M. Emre Celebi, Brian Helba, Michael A. Marchetti, Stephen W. Dusza, Aadi Kalloo, Konstantinos Liopyris, Nabin Mishra, Harald Kittler, and Allan Halpern. 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), 2018. arXiv:1710.05006.

[20] Marc Combalia, Noel C. F. Codella, Veronica Rotemberg, Brian Helba, Veronica Vilaplana, Ofer Reiter, Cristina Carrera, Alicia Barreiro, Allan C. Halpern, Susana Puig, and Josep Malvehy. Bcn20000: Dermoscopic lesions in the wild, 2019. arXiv:1908.02288.

[21] G. Bradski. The OpenCV Library. *Dr. Dobb's Journal of Software Tools*, 2000.

[22] Alexandru Telea. An image inpainting technique based on the fast marching method. *Journal of Graphics Tools*, 9(1):23–34, 2004. arXiv:https://doi.org/10.1080/10867651.2004.10487596, doi:10.1080/10867651.2004.10487596.

[23] Seeja R D and Suresh A. Deep Learning Based Skin Lesion Segmentation and Classification of Melanoma Using Support Vector Machine (SVM). *Asian Pacific Journal of Cancer Prevention*, 20(5):1555–1561, May 2019. URL: http://journal.waocp.org/article_87402.html, doi:10.31557/APJCP.2019.20.5.1555.

[24] Şaban Öztürk and Umut Özkaya. Skin Lesion Segmentation with Improved Convolutional Neural Network. *Journal of Digital Imaging*, 33(4):958–970, August 2020. URL: http://link.springer.com/10.1007/s10278-020-00343-z, doi:10.1007/s10278-020-00343-z.

[25] Halil Murat Ünver and Enes Ayan. Skin Lesion Segmentation in Dermoscopic Images with Combination of YOLO and GrabCut Algorithm. *Diagnostics*, 9(3):72, July 2019. URL: https://www.mdpi.com/2075-4418/9/3/72, doi:10.3390/diagnostics9030072.

[26] Dermoscopy | opendemo. URL: http://opendemo.uib.es/dermoscopy/.

[27] Daniel D. Polsby and Robert Popper. The third criterion: Compactness as a procedural safeguard against partisan gerrymandering. *Yale Law Policy Review*, 9(2), 1991. URL: http://dx.doi.org/10.2139/ssrn.2936284.

[28] Ali Madooei and Mark S. Drew. Incorporating Colour Information for Computer-Aided Diagnosis of Melanoma from Dermoscopy Images: A Retrospective Survey and Critical Analysis. *International Journal of Biomedical Imaging*, 2016:1–18, 2016. URL: https://www.hindawi.com/journals/ijbi/2016/4868305/, doi:10.1155/2016/4868305.

[29] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[30] James Bergstra and Yoshua Bengio. Random search for hyper-parameter optimization. *J. Mach. Learn. Res.*, 13(null):281–305, February 2012.

[31] Leo Breiman. Random Forests. *Machine Learning*, 45(1):5–32, 2001. URL: http://link.springer.com/10.1023/A:1010933404324, doi:10.1023/A:1010933404324.

[32] F.J. Ferri, P. Pudil, M. Hatef, and J. Kittler. Comparative study of techniques for large-scale feature selection. In *Pattern Recognition in Practice IV*, volume 16 of *Machine Intelligence and Pattern Recognition*, pages 403–413. North-Holland, 1994. URL: https://www.sciencedirect.com/science/article/pii/B9780444818928500407, doi:https://doi.org/10.1016/B978-0-444-81892-8.50040-7.

[33] Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL: https://proceedings.neurips.cc/paper/2017/file/8a20a8621978632d76c43dfd28b67767-Paper.pdf.

[34] Scott M Lundberg, Bala Nair, Monica S Vavilala, Mayumi Horibe, Michael J Eisses, Trevor Adams, David E Liston, Daniel King-Wai Low, Shu-Fang Newman, Jerry Kim, et al. Explainable machine-learning predictions for the prevention of hypoxaemia during surgery. *Nature Biomedical Engineering*, 2(10):749, 2018.

[35] Scott M. Lundberg, Gabriel Erion, Hugh Chen, Alex DeGrave, Jordan M. Prutkin, Bala Nair, Ronit Katz, Jonathan Himmelfarb, Nisha Bansal, and Su-In Lee. From local explanations to global understanding with explainable ai for trees. *Nature Machine Intelligence*, 2(1):2522–5839, 2020.

8. Appendix

Feature Set		
A: Asymmetry	shape_symmetry	Shape symmetry score of a lesion $\in \{-1, 0, 1, 2\}$
	texture_symmetry	Texture symmetry score of a lesion $\in \{-1, 0, 1, 2\}$
	dissimilarity	Mean dissimilarity over all lesion patches
	correlation	Mean correlation over all lesion patches
	energy	Mean energy over all lesion patches
	contrast	Mean contrast over all lesion patches
B: Border	border_compactness	Polsby-Popper border compactness score of a lesion's border
C: Color	black	1 if $\geq 1\%$ of lesion pixels are black, 0 otherwise
	white	1 if $\geq 1\%$ of lesion pixels are white, 0 otherwise
	red	1 if $\geq 1\%$ of lesion pixels are red, 0 otherwise
	light_brown	1 if $\geq 1\%$ of lesion pixels are light brown, 0 otherwise
	dark_brown	1 if $\geq 1\%$ of lesion pixels are dark brown, 0 otherwise
	blue_gray	1 if $\geq 1\%$ of lesion pixels are blue/gray, 0 otherwise
	num_colors	Number of unique colors within the lesion $\in [0, 6]$
	color_intersect	Intersection between 3D histogram of lesion pixels and 3D histogram of skin pixels
	color_chisq	Chi-squared distance between 3D histogram of lesion pixels and 3D histogram of skin pixels
D: Diameter	diameter	Approximate lesion diameter in pixels
Metadata	age_approx	Approximate patient age
	sex	Patient sex

Table 4: Final feature set used to train our classification model.