Projects in Data Science Spring 2024
BSPRDAS1KU

Andreas Glenn Patton-Jensen
angp@itu.dk

Johan Hausted Schmidt
jhsc@itu.dk

Al Sahabir Nur Yeamin
alye@itu.dk

Elias Fischer Hegelund
elhe@itu.dk

Kristoffer Schmidt
krisc@itu.dk

Mikkel Emil Jensen
memj@itu.dk

1 Introduction

In 2020 alone, the World Health Organization reported approximately 1.5 million skin cancer diagnoses ("WHO", 2022) (2). A major concern is patients being diagnosed at advanced stages, given the strong correlation between late diagnoses and increased mortality risk (Markers H.S.B., Xavier, 2016) (12). This can be attributed to a variety of factors, including the reluctance to get tested, or by limited access to health care resources due to cost and availability. A way to mitigate the problem of late diagnosis could be a tool capable of diagnosing skin cancer in lesions with relatively high accuracy via a submitted photograph. If such a tool was to be developed with high enough precision to give an accurate initial assessment, it could significantly decrease the mortality risk associated with skin cancer. In this report, we aim to illustrate the methods and processes we have undertaken, including the guidance provided by dermatologists. Our objective is to investigate if this can be done reliably such that people have an idea of whether or not they are at risk of skin cancer and should seek a professional's opinion.

2 Dataset

In this project we are working with part of the PAD-UFES-20 dataset (Pacheco, et al., 2020)(6). 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, skin lesion location, Fitzpatrick skin type, and skin lesion diameter. To create our model we only use two features which are "patient_id" for splitting the data correctly and the "diagnostics" as the target variable.

Since our group is the result of the merging of group J and N we are working with images 1154 - 1281 from group J and 1665 - 1791 from group N, which results in a total of 253 images.

2.1 Data cleaning

We started the data cleaning process by reviewing each individual image one by one. We manually removed images that may cause difficulty for the feature extractions method that we are going to be implementing. These were images with multiple possible lesions as well as images where it was not obvious where the mask should be drawn.

3 Masking

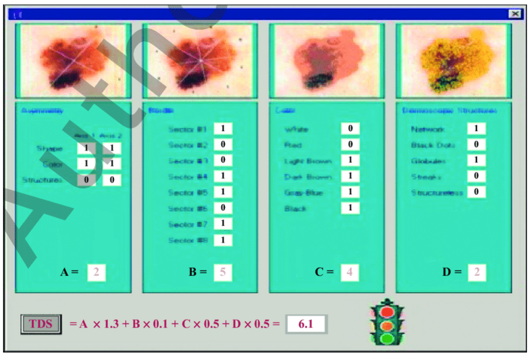
After the data cleaning, the images were distributed equally among the different group members where a mask around the lesion was manually created for each of the images. This also served as a second visual inspection where we discovered a few more images that could potentially be difficult for our feature extraction to operate on. After the initial inspection and masking process we ended up with a total of 222 images.

4 Features

We decided to use 5 features in order to determine whether or not a lesion was cancerous. Here, we started with three features, which was followed by the addition of two more. We first considered the ABCDE method; asymmetry, border, color, diameter and evolution (Thomas, 1998) (9). We thought asymmetry would be most feasible to measure, as we had masks for the lesions already. In addition, we thought color would be a good feature to analyze as the present colors were easily recognizable in most of the images. We decided that due to the thick brush used while masking, we did not feel the perimeter of the lesions were accurate enough to give us a good result for the border feature. This is because we found that the border is related to the presence of sharp, abrupt endings of a pigment pattern (Kunz M, n.d.) (8), which we do not feel is illustrated very well in the masking process given the brush we used. We also found that diameter was unreasonable to implement because the algorithm has no way of measuring the length of the diameter without more information on the photo, such as the distance the photo was taken from the lesion. Lastly, evolution is also not measurable from a single photo which is what the algorithm uses. Although we had metadata for an approximate diameter and evolution, this is of no use when creating an algorithm to measure a singular image. Along with the two features asymmetry and color, we concluded that blue-white veil would be the best third feature to implement, because of its high sensitivity and high specificity (Wolner ZJ, et al. 2017) (14).

5 Hand annotation and Krippendorf

Hand annotations needed to be done on approximately 100 images, so we decided to use group J's images. To form the basis of our hand annotation, given our already decided features, we had to find a method to rate them. To do this, we looked into a variety of studies and resources. ResearchGate (13) provided Figure 1, which helped us decide on what scale to use when annotating the features of the photos. It also seemed to be very intuitive to give a binary response when considering the presence of a feature. This being whether the feature in question is there or not - 2 outcomes. This is typically given as 0 (absence) and 1 (presence) respectively; however, this was later adapted to 1 and 2 in order to more easily be interpreted by Krippendorf. This is briefly illustrated in Figure 1.



Asymmetry

Axis 1 Axis 2
Shape 1 1
Color 1 1
Structures 0 0

A = 2

Border

Sector #1 1
Sector #2 0
Sector #3 0
Sector #4 1
Sector #5 1
Sector #6 0
Sector #7 1
Sector #8 1

B = 5

Color

White 0
Red 0
Light Brown 1
Dark Brown 1
Gray-Blue 1
Black 1

C = 4

Dermoscopic Structures

Network 1
Black Dots 0
Globules 1
Streaks 0
Structureless 0

D = 2

TDS = A × 1.3 + B × 0.1 + C × 0.5 + D × 0.5 = 6.1

Figure 1: Computer-automated ABCD

DermnetNZ (1) gave us a good understanding of what asymmetry and blue-white veil looks like, as well as providing two studies that illustrated the high degree of accuracy in skin cancer detection of these two features, particularly in the hands of a "non expert", which we are considered to be. (Zalaudek, I., et al., 2006) (10)

For asymmetry, we visually referenced the masks of the given lesion, and chose the visibly longest straight line from one side of the lesion to the other. Then we tried to imagine it flipped onto itself along this axis, and decide whether or not enough of the flipped lesion fit well with the opposing half. If not, the score of 2 was returned, implying asymmetry. The same was visually implemented in terms of color, where we visually guessed where the color's would be when flipped over the longest line, and if similar colors were flipped onto themselves, it was considered symmetrical, and vice versa. These two estimations in terms of shape and color symmetry were also done for the perpendicular axis. Therefore four scores of 1 or 2 were outputted for the asymmetry of a given image. The second feature we checked for was the presence of blue-white veil. This was done by comparison to the images presented on the website ScienceDirect (11).

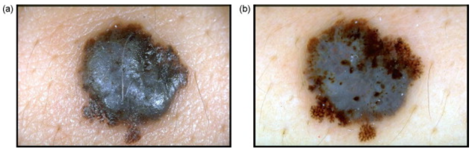


Figure 2: Blue white veil images

The final feature that we decided to use was color. Such that all the annotators had the same interpretation of the colors in terms of the lesion, our group frequently observed in Figure 3 from Dermoscopedia (8).

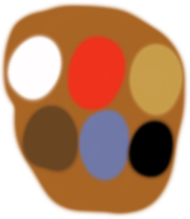


Figure 3: ABCD colors schematic

This also gave us insight into the relevant colors in terms of whether or not they are a signal of cancerous lesion. Because color can be ambiguous, we agreed that when we were creating the annotation guide that we would simply count all the visible colors present in the lesion provided they accounted for a reasonable amount of the picture.

After we had finished our hand-annotations, we used the Krippendorf's Alpha algorithm to measure the inter-rater reliability between our annotators. Although we only had two annotators, we still thought it would be interesting to see the agreement between the annotators for the different features. For this, we had a score of 0.663, which indicates relatively low levels of agreement. This is likely due to the vastly different interpretations of symmetry, and how much asymmetry is required in order to consider an image asymmetrical.

6 Feature extraction

6.1 Color

Our color feature starts by defining the RGB ranges for the six different colors (white, red, light-brown, dark-brown, green/blue and black) we used during the hand annotation process. The algorithm examines all the pixels of the image within the masked lesion area. If any pixel in the lesion falls within any of the color ranges, that color is recorded as detected. The resulting score reflects the different colors present in the image.

6.2 Symmetry

Our symmetry feature is comparing the symmetry of the shape as well as the color information in the lesion. Using the mask, the longest axis of the lesion is found as well as the axis that runs perpendicular to the longest axis. The longest axis was chosen to give consistent results, even if the images were taken from different rotations. For example, the same lesion would return the same score, regardless of the rotation of the photo.

6.2.1 Shape symmetry

For symmetry assessment of the shape, the lesion mask is flipped around both its axis and a symmetry score is calculated using the formula:

$$\text{symmetry} = \frac{\text{intersection of the masks}}{\text{total area of mask}} \tag{1}$$

The symmetry score is then calculated by taking the intersection divided by the total area of the mask.

6.2.2 Color symmetry

In addition to geometric symmetry, color symmetry within a skin lesion is assessed using a method called Structural Similarity Index or SSIM for short. SSIM excels at quantifying how similar two sections of an image are. The structural similarity index measures three different attributes: luminance, contrast and structure. These are expanded upon below in the corresponding sections.

Luminance
The luminance compares the mean pixel values of the original and the flipped lesion, which is crucial for assessing the overall brightness symmetry of the lesion. This is quantified by the formula:

$$L(x, y) = \frac{2\mu_x\mu_y + c_1}{\mu_x^2 + \mu_y^2 + c_1} \tag{2}$$

Here μ_x and μ_y are the mean intensities of the original and the flipped images respectively, and c_1 is typically a small constant to avoid dividing by zero. The multiplication of the mean intensities by 2 in the numerator combined with squaring of the mean intensities in the denominator ensures a standardized output between 0 and 1, where 1 is very symmetrical and 0 is very asymmetrical.

Contrast
To evaluate the similarity in the color contrast we compare the standard deviation of pixel's intensity in both the original and flipped images using the formula:

$$C(x, y) = \frac{2\sigma_x\sigma_y + c_2}{\sigma_x^2 + \sigma_y^2 + c_2} \tag{3}$$

The standard deviation of the pixel intensity is represented by σ_x and σ_y, for the original and flipped images respectively. Similar to the formula for luminance the results are standardized, and the output is a number between 0 and 1 where a number closer to 1 suggests more similarity between the images.

Structure
Lastly, the comparison of structure is done by assessing the correlation between the pixels of the original and the flipped image. The formula quantifying the structural similarity is given by:

$$S(x, y) = \frac{\sigma_{xy} + c_3}{\sigma_x\sigma_y + c_3} \tag{4}$$

Where σ_{xy} is the covariance of the original and flipped images. Covariance is a measure of how much the respective pixels of each image are similar. The constant c_3 prevents division by 0 exactly the same way the other equations do.

Overall equation

These three formulas combined results in an overall SSIM score:

$$SSIM(x, y) = [L(x, y)] \cdot [C(x, y)] \cdot [S(x, y)] \tag{5}$$

The final output is the product of all the different features resulting in a score between 0 and 1 where 1 indicates a perfect symmetry score and 0 suggests less similarity.

7 Blue-white veil

To check for the presence of blue-white veil, every pixel within the image is inspected. This being the blue channel of each pixel as well as the luminance. The algorithm calculates the ratio of blue channel intensity to the total luminance of the pixel. Pixels with a blue channel intensity greater than or equal to 0.3 are considered and are classified as potential veil pixels. The total luminance for the pixel must also fall within the range 0.6 and 2. If the pixel meets both these conditions it is classified as a blue-white veil pixel (Madooei A, el al., 2013) (4).

8 Haralick and additional color feature

In addition to the three features, we were considering other potential ways we could increase the accuracy of our diagnoses tool. Here we found 2 different studies, one that demonstrates the relationship between the texture and cancerous lesions (Dalia N., et al., 2014) (7), and another that indicates the presence of different colors in melanoma (Cancer.org, 2023) (3).

8.1 Haralick

Haralick features measure spatial relationships between adjacent pixels in an image using the gray-level co-occurrence matrix (GLCM) approach. This method captures statistical patterns or textures based on the distribution of pixel pairs or patterns. Haralick features include contrast, correlation, dissimilarity, energy, and homogeneity. Contrast shows how much adjacent pixels differ in brightness, with high contrast indicating significant differences, such as sharp edges. Correlation describes how pixels relate to each other, where high correlation means the pixels are similar, and low correlation means they are different. Dissimilarity measures the average difference in brightness between neighboring pixels, with high dissimilarity indicating a lot of variation. Energy indicates how uniform an image is, with high energy meaning the image is more consistent and less noisy. Homogeneity measures the similarity of neighboring pixels, with high homogeneity indicating that they are more alike, leading to smoother textures.

8.2 Color variability

For the additional color feature, we employ the SLIC segmentation method. Initially, SLIC places markers within the masked region of the lesion. Each pixel within the lesion is then assigned to a seed that is both close in distance and in color, resulting in the grouping of pixels. The assignment and grouping are iteratively refined, ensuring that each superpixel accurately represents a segment of the lesion with similar color characteristics. This step is repeated several times until you end up with a group of super pixels.

The standard deviation between the superpixels and the average color of the lesion is then calculated to give an overall idea of the variability of color within the lesion.

9 Model

9.1 Our chosen model

There are many models that can be used to do binary classification such as logarithmic regression models, Linear support vector classification (svc's), k-nearest neighbors(KNN), and various types of trees. There are of course drawbacks and benefits of each, so we should try to use the one that works with our data the best.

In the beginning we were considering using a KNN model, it was however quickly apparent that it was not optimal for our use case. KNN classifies by finding a k-amount of nearest neighbors in an n-dimensional system, and then classifies the lesion/item according to the majority of its k-neighbors. This sounded very promising, but due to the curse of dimensionality, KNN's performance can degrade in high-dimensional spaces. We have implemented our features in a way where we have a lot of different features under the same categories, under the premise that more data is more beneficial. We have, due to that train of thought, been very sparse when pruning our dataset, and there may therefore be a bit of noise in our dataset which the KNN will be sensitive to since it relies on distance metrics. Relying on distance is also suboptimal since we have done very little feature-scaling/normalization.

A common mantra in machine learning circles has for many years been that more data means more accuracy and better results. This, as we just learned from our KNN model, is not always true. But to use our large amount of data (features) to as much of an advantage as possible, we have decided to use a decision tree model, more specifically, the DecisionTreeClassifier from scikit-learn. When dealing with a large quantity of varying attributes like ours, decision trees have a lot of benefits.

1. Flexible Handling: Decision trees handle both categorical and numerical data without data transformation.
2. Handling of Interaction Effects/Patterns: They capture complex relationships between attributes and the categorization effectively.
3. Automatic Feature Selection: Decision trees inherently perform feature selection, focusing on the most discriminative attributes at each node.
4. Interpretability: Easy to interpret and visualize, aiding understanding and explanation of predictions. A visualization of our final model can be found on our GitHub.

9.2 Training (weighing and grid search cross validation)

When running our program for feature extraction, a file called "features.csv" will be created. This file has the lesion id in the first column, and in the last two a boolean diagnosis for cancer as well as patient id. All three of these columns get removed in the start of the training file, so that the model can not learn from it. It is, however, necessary to have these columns in the features.csv file, since when splitting the data in the grid search cv (cross validation) it is usually recommended to not split images from the same patient. The diagnosis is also necessary since it is what we need to train the model to predict, it is of course just taken from the metadata file we have been given.

Before the grid search, we also split the data into two sets with 80% in the first and 20% in the second. Then the grid search is performed using the portion with 80% of the data and the final 20% is saved for testing the model.

Since we are dealing with cancer, we want to focus on reducing false negatives. We implement this by assigning higher weights to cancer-positive cases during training. This prioritizes sensitivity, minimizing the chance of missing potentially dangerous lesions. This is very beneficial in achieving our purpose of identifying cancerous lesions.

As mentioned before we use cross validation to improve our model. We do this together with grid search, to fine tune parameters before the final model is trained. The point of cross-validation is to help check how well our machine learning model will work on new data, this is done by splitting our data into subsets. We use the scikit-learn function GroupKFold to keep images from the same patient together during training and testing. This prevents the model from learning details specific to individual patients, making it more generalizable.

We use grid search to find the best settings for our Decision Tree model. Grid search tries out different values for parameters like max_depth, min_samples_split, and min_samples_leaf. Max depth refers to the maximum depth of the decision tree. Having a "shallow tree" - low max depth, relates to having fewer splits, and therefore less complexity which can cause underfitting. This is compared to a deeper tree (higher max depth), which refers to having too many splits and causing overfitting of the data. The grid search finds the optimal depth, without capturing too much noise, but maintains the ability to identify patterns. The minimum samples split is a parameter that sets the minimum number of samples a node must have before it can be split into smaller nodes. The minimum samples leaf parameter controls the minimum number of samples that are required to be at a leaf node. By setting a minimum number of samples required at a leaf node, we can prevent the tree from overfitting by creating branches with very few samples.

GroupKFold is used beforehand to group the data into the different subsets that the grid search is then used on. The model is trained and tested multiple times with different settings, and the best model is chosen. This model is then tested on the 20% "test set" which the grid search hasn't been used on. We then check its accuracy, confusion matrix, and recall and retrain the best model on all of our data, such that it's the best possible. Since we no longer need to have separate training data, there is no point in not making our model benefit from it. Finally it is saved with pickle.

9.3 How our model works

A decision tree works as follows, in the specific case of our data: It takes one classifier at a time with a specific constraint. The specific node has an interval in which the decision tree will check and mark all the samples and divide them into two groups which then continues the path of the tree. The samples will either be marked as cancer, non-cancer or it continues to the next node. A specific example shown below in Figure 3 runs the masks through the symmetry feature and checks for a specific symmetry score and thereafter divides the samples into their respective groups with regards to the data. Here the samples are marked as non-cancer or cancer and the tree is continued to the next node.

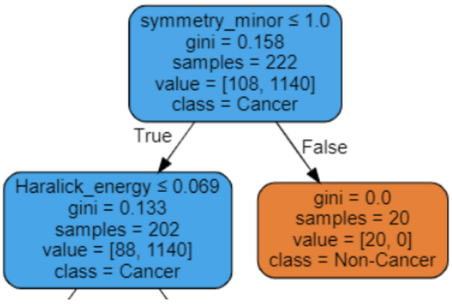
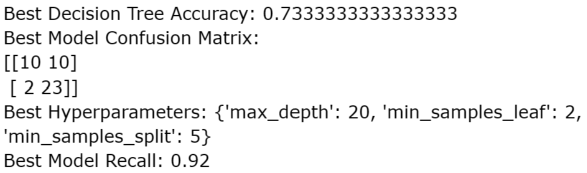


Figure 4: Model example

9.4 Model results



```
Best Decision Tree Accuracy: 0.7333333333333333
Best Model Confusion Matrix:
[[10 10]
 [ 2 23]]
Best Hyperparameters: {'max_depth': 20, 'min_samples_leaf': 2,
'min_samples_split': 5}
Best Model Recall: 0.92
```

Figure 5: Model output

From the results we performed on the test data, we can see that the best decision tree accuracy is 0.733. This result is obtained through the best set of hyperparameters found through grid search cross-validation. The accuracy score of 0.7333 suggests that 73.33% of the predictions made by the model on the test data were correct. The confusion matrix breaks down how many of the 45 photos we tested the model on were correctly and incorrectly identified, and what the model classified them as. The top left represents 10 of the test lesions being correctly classified as non-cancerous (true negative). In the top right, 10 lesions are incorrectly classified as cancerous when they were actually non-cancerous (false positive). 2 of the lesions were incorrectly classified as negative (non-cancerous) when they were actually cancerous (false negative). Lastly, 23 of the lesions were correctly classified as cancerous. Through this, we can see that 33 of the lesions were correctly classified, compared to the 12 that weren't. This gives us the ratio:

$$\frac{33}{33+12} = \frac{33}{45} = 0.7333$$

which matches what was shown above. In terms of the best hyperparameters, that resulted in the highest accuracy during the grid search cross-validation process:
max_depth: 20
min_samples_leaf: 2
min_sample_split: 5

The "Best Model Recall", represents the proportion of positive cases that were correctly identified by the model. Here we were able to score 92% as calculated below.

$$\frac{23}{23+2} = \frac{23}{25} = 0.92$$

This is also known as the sensitivity of our model. The specificity of our model, also known as the percentage of how many non-cancerous lesions were correctly identified, was at $\frac{10}{20} = 0.5$ or 50%. We also believe that classifying non-cancerous lesions as cancerous (false positive) compared to failing to identify cancerous lesions (false negative), is not a good trade-off which is why we penalize false negatives.

10 Discussion of features

For our model we have implemented a total of 5 features, these being symmetry, color SLIC, color counter, blue-white veil and haralick. To better analyze and understand how our features impact the performance of our model, we plotted some of the features, measuring the frequency of observing

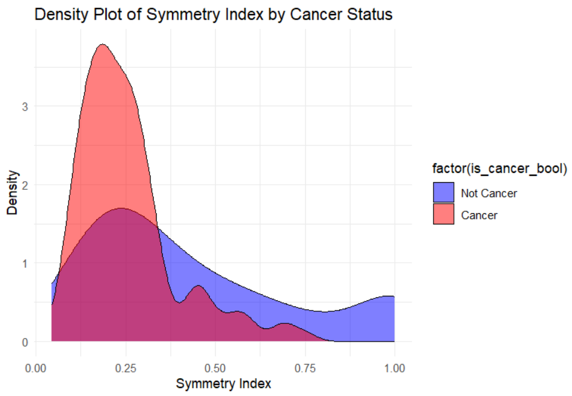


Figure 6: Density plot for assymetry

different values for both cancerous and noncancerous images.

In figure 6, we observe that a large proportion of cancerous images (red area), more often tend to have a lower symmetry index (indicating that the lesion is more asymmetric), and that almost none of the cancerous photos have measured a symmetry score of higher than approximately 0.75. For the non-cancerous images (blue area) the same trend of having a higher frequency for low symmetry scores can be seen, but at the same time, the variance/spread is higher. Here, the variance is at 0.086 for non-cancerous density, compared to the cancerous symmetry density at 0.019. From the above figure, we can conclude that the above feature does well in distinguishing between cancer and non cancer. From figure 6, we identify a clear pattern, where non-cancerous lesions tend to vary more in asymmetry, compared to cancerous ones which are more grouped on the less symmetrical end of the plot.

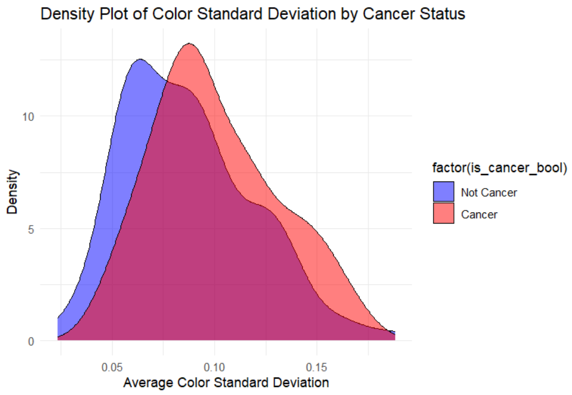


Figure 7: Density plot for standard deviation

Figure 7 illustrates the average standard deviation of the 3 colors red, green and blue from each image and plots it as a density plot for the observed values categorized cancerous images (red) and non cancerous images (blue). We observe the above density plot to be bimodal with quite a lot of overlap, though it is still quite apparent that for cancerous images, the average standard deviation tends to be higher. We can interpret this as a higher standard deviation does not necessarily guarantee cancer, but will always have a higher probability of being cancer than not according to our data. Admittedly this feature is still not the best at separating cancer and non cancer since the overlap of the two functions is quite large. We also calculated the variance for the two plots which gave us very similar results: 0.0195 for cancer and 0.0200 for non-cancerous lesions.

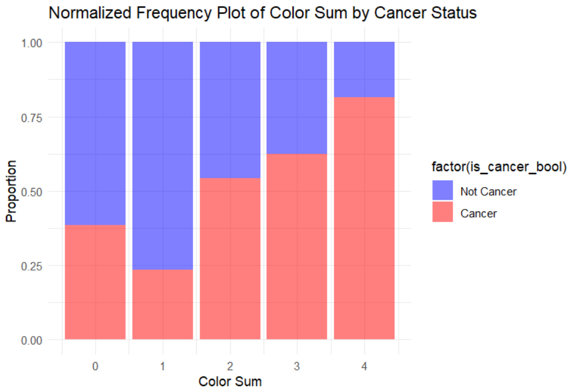


Figure 8: Bar plot demonstrating the amount of specified colors within a lesion

Figure 8 illustrates the detected number of different colors in a lesion from the color count feature, and is then normalized to take account for the number of samples in each category. In Figure 8 we see that images with a higher frequency of colors are more likely to be cancerous. The cancerous lesion color frequency seems to be following a linear pattern corresponding to the amount of colors. However this does not apply for the 0 colors present category. A reason for this could be because some of the colors in the lesions are not within the color ranges in the program, particularly in terms of the cancerous ones, thereby not following the linear pattern presented by the other color frequencies.

Another concern could also be that no images were observed to have more than 4 colors even though there were 6 possible colors, though it is of course not guaranteed that 5 or more colors would be present in our dataset.

10.1 Blue-White veil and Haralick

For blue white veil and Haralick we observed zero to little differentiation between cancer and non cancer images.

11 Discussion of model results

As we can see from our discussion of the features, two of our features, namely blue white veil and Haralick could be considered quite unimpactful when looking at the density distribution. When looking at a representation of our model exported via graphviz we can see that it does use the features in its classification progress, so it could be assumed that they do in some small degree help make the model better, or perhaps it is just noise. When we tested the features by making an identical model without them, it was slightly worse, by a small margin, of just about 8 percent points. This might be enough to conclude that it does help, but it could also just be beneficial noise rather than meaningful data. Further analysis and testing would be necessary if we want to be sure whether their inclusion genuinely enhances model performance or if their benefit is incidental.

12 Discussion Of Related Work, future endeavors and current limitations

The use of predictive models for skin cancer detection holds great promise. An automated system for detecting skin cancer through images could significantly enhance the speed of diagnosis, potentially lowering mortality rates and increasing healthcare efficiency. However, the current system has limitations. Even though we achieved promising results using asymmetry and color separation features, the blue-white veil and Haralick features did not perform as expected. With additional refinement, these features could improve the precision of our model further.

Despite these limitations, our project demonstrated the possibility of diagnosing skin cancer from images with accuracy surpassing random chance, with probability of random classification reaching our accuracy of $p \leq 0.001$. This suggests that it is possible to reliably diagnose skin lesions using a classification model, offering individuals an early assessment of their risk and prompting them to seek professional evaluation if necessary. Our final model achieved approximately 73% accuracy on the test data, a substantial improvement over random guessing, though still not highly reliable. Given the rudimentary nature of our manually extracted features, we are satisfied with this outcome. With more data, advanced extraction methods, fine-tuning, and additional features, there is potential for even better results, albeit requiring numerous iterations and extensive feature research.

If it had been possible, we believe we could have developed a more sophisticated model by avoiding the ”manual” scoring of an image’s attributes. Here, ”manual” refers to features chosen and extracted by us, not the manually annotated data. Instead, we could use a model that processes the entire image and then learns by itself the best features for detecting the presence of cancer. Many models capable of this exist, such as Convolutional Neural Networks (CNNs). A CNN or a similar type of model is something we would have liked to have made, but that was outside of the scope of this project. However, a way we could have improved our model without using a CNN, could have been to find some better features, since we found a few that did not contribute as much as initially expected .

Using a CNN which “picks” its own features, it is possible to get much more accurate results. Because of this, we feel spending time trying to find more optimal features could result in us with spending a lot of time chasing very marginal results. So if we were to remake this project in the future we would find it more wise to rethink our entire approach/models than to nitpick over feature selection. It has already been shown that a much better result is achievable in that way, by Bhuvaneshwari Shetty, and her colleagues from India, using the skin lesion dataset HAM10000, where they reached an accuracy of up to 95.18%. This is significantly better than our model, especially when it is taken into account that their model has 7 different classifications, compared to ours being simply binary (B. Shetty et al. 2022) (5).

13 Conclusion

In this project, we investigated whether we could reliably detect skin cancer by developing a predictive model to classify cancer from images. Our model utilized features commonly analyzed by dermatologists, such as asymmetry, color, the presence of blue-white veil, and lesion structure, all supported by relevant literature.

After implementing these features, we optimized the model's hyperparameters through grid search cross-validation to enhance predictive accuracy. The model achieved an overall accuracy of 73.33% in distinguishing between cancerous and non-cancerous lesions, and also correctly identified 92% of the cancerous lesions, meeting our objective of minimizing false negatives.

References

[1] Three-point checklist . *Dermnetnz*, 2008. https://dermnetnz.org/cme/dermoscopy-course/three-point-checklist.

[2] Automatic classification of skin lesions using colormathematical morphology-based texture descriptors. *World Health Organization*, 2022.

[3] Types of cancer melanoma. *CancerOrg*, 2023. https://www.cancer.org.au/cancer-information/types-of-cancer/melanoma.

[4] Madoei A. Automatic detection of blue-white veil by discrete colour matching in dermoscopy images. *SimonFraserUniversity*, 2013. https://www.sfu.ca/~amadooei/files/MICCAI2013.pdf.

[5] Shetty B, et al. Skin lesion classification of dermoscopic images using machine learning and convolutional neural network. *Nature*, 2022. https://www.nature.com/articles/s41598-022-22644-9#citeas.

[6] Andre G. C. Pacheco et al. PAD-UFES-20: A skin lesion dataset composed of patient data and clinical images collected from smartphones. *Mendeley*, 2020. https://data.mendeley.com/datasets/zr7vgbcyr2/1.

[7] Dalia J., et al. Diagnosis of skin cancer using image texture analysis. *ResearchGate*, 2014. https://www.researchgate.net/profile/Loay-George-2/publication/281715077_Diagnosis_of_Skin_Cancer_Using_Image_Texture_Analysis/links/55f5b40008ae63926cf4e897/Diagnosis-of-Skin-Cancer-Using-Image-Tex pdf.

[8] Michael Kunz, et al. ABCD rule. *Dermoscopedia*, n.d. https://dermoscopedia.org/ABCD_rule.

[9] Thomas L, et al. Semiological value of ABCDE criteria in the diagnosis of cutaneous pigmented tumors. *NCBI*, 1998. https://pubmed.ncbi.nlm.nih.gov/9693179/.

[10] Zaludek L, et al. Three-point checklist of dermoscopy: an open internet study. *NCBI*, 2006. https://pubmed.ncbi.nlm.nih.gov/16445771/.

[11] Emre Celebi M., et al. Automatic detection of blue-white veil and related structures in dermoscopy images. *ScienceDirect*, 2006. https://www.sciencedirect.com/science/article/pii/S0895611108000815.

[12] Xavier Markers H.S.B. Delay in cutaneous melanoma diagnosis. *PMC pubmed central*, 2016. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4979809/.

[13] et al. Piccolo, Domenico. Computer-automated ABCD. *ResearchGate*, 2014. https://www.researchgate.net/publication/261602961_Computer-automated_ABCD_versus_dermatologists_with_different_degrees_of_experience_in_dermoscopy.

[14] Wolner Zachary j., et al. Enhancing skin cancer diagnosis with dermoscopy. *NCBI*, 2017. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5659633/.