Skin Cancer Prediction Using Three Major Classification Algorithms

by

Md. Ariful Islam
ID: CSE1903018063

Md. Raihan Islam
ID: CSE1903018052

Md. Jisan Ahmed
ID: CSE1903018067

Md. Arman Ali
ID: CSE1903018096

Supervised by
Ahmed Shafkat

Submitted in partial fulfillment of the requirements for the degree of
Bachelor of Science in Computer Science and Engineering



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
SONARGAON UNIVERSITY (SU)

May 2023

APPROVAL

The thesis titled "**Skin Cancer Prediction Using Three Major Classification Algorithms**" submitted by Md. Ariful Islam (CSE1903018063), Md. Raihan Islam (CSE1903018052), Md. Jisan Ahmed (CSE1903018067) and Md. Arman Ali (CSE1903018096) to the Department of Computer Science and Engineering, Sonargaon University (SU), has been accepted as satisfactory for the partial fulfillment of the requirements for the degree of Bachelor of Science in Computer Science and Engineering and approved as to its style and contents.

Board of Examiners

Ahmed Shafkat
Lecturer,
Department of Computer Science and Engineering
Sonargaon University (SU)

Supervisor

(Examiner Name and Signature)
Department of Computer Science and Engineering
Sonargaon University (SU)

Examiner 1

(Examiner Name and Signature)
Department of Computer Science and Engineering
Sonargaon University (SU)

Examiner 2

(Examiner Name and Signature)
Department of Computer Science and Engineering
Sonargaon University (SU)

Examiner 3

DECLARATION

We, hereby, declare that the work presented in this report is the outcome of the investigation performed by us under the supervision of **Ahmed Shafkat**, Lecturer, Department of Computer Science and Engineering, Sonargaon University, Dhaka, Bangladesh. We reaffirm that no part of this thesis has been or is being submitted elsewhere for the award of any degree or diploma.

Countersigned

(Ahmed Shafkat)
Supervisor

Signature

Md. Ariful Islam
ID: CSE1903018063

Md. Raihan Islam
ID: CSE1903018052

Md. Jisan Ahmed
ID: CSE1903018067

Md. Arman Ali
ID: CSE1903018096

ABSTRACT

Skin Cancer Classification is a major public health concern, and correctly classifying skin lesions as benign or malignant is critical for accurate diagnosis and treatment. This text provides a comprehensive guide to building and evaluating models to accurately classify skin lesions as benign or malignant. It includes an introduction, overview, data collection and preparation, setting up the environment, data preprocessing, splitting the data into train and test sets, scaling the data, models (SVM, RF and KNN), visualization and analysis, and conclusion. The project used three classification algorithms: Support vector machines, Random Forest, and K-Nearest Neighbors (KNN) to create models for skin cancer classification. Results showed that Random Forest had the highest accuracy of 98%, followed by SVM with 94% and KNN with 74% accuracy. This initiative advances the field of skin cancer diagnostics by making use of classification algorithms.

ACKNOWLEDGMENT

At the very beginning, we would like to express our deepest gratitude to the Almighty Allah for giving us the ability and the strength to finish the task successfully within the scheduled time.

We are auspicious that we had the kind association as well as supervision of **Ahmed Shafkat**, Lecturer, Department of Computer Science and Engineering, Sonargaon University whose hearted and valuable support with best concern and direction acted as necessary recourse to carry out our project.

We would like to convey our special gratitude to **Brig. Gen. (Retd) Prof. Habibur Rahman Kamal,** Dean, Faculty of Science and Engineering for his kind concern and precious suggestions.

We are also thankful to all our teachers during our whole education, for exposing us to the beauty of learning.

Finally, our deepest gratitude and love to my parents for their support, encouragement, and endless love.

LIST OF ABBREVIATIONS

ACC	Accuracy
AI	Artificial Intelligence
BC	Benign Classification
F1	F-measure
MC	Malignant Classification
ML	Machine Learning
PRE	Precision
REC	Recall
RF	Random Forest
SVM	Support Vector Machine

TABLE OF CONTENTS

Title		Page No.
DECLARATION ……..……		iii
ABSTRACT ……...……..		iv
ACKNOWLEDGEMENT …….……..		v
LIST OF ABBREVIATION …….……		vi
CHAPTER 1		1
INTRODUCTION		
1.1	Overview of Skin Cancer Classification…….	1
1.2	Importance of Classification Algorithms ……	1
1.3	Dataset Collection and Preparation……..	1
CHAPTER 2		2
SETTING UP THE ENVIRONMENT		
2.1	Importing Necessary Libraries…….	2
2.2	Mounting the Dataset from Google Drive……	2
2.3	Defining the Image Directory Path……..	2
CHAPTER 3		3
DATA PREPROCESSING		
3.1	Reading Image Data…….	3
3.2	Converting Images to Numpy Arrays……..	3
3.3	Splitting the Dataset into Train and Test Sets…….	3
3.4	Scaling the Data……...	3
CHAPTER 4		4-7
METHODOLOGY		

4.1 Introduction to Support Vector Machines (SVM) 4
4.1.1 Support Vector Machines …….. 4
4.1.2 Random Forest …….……. 4
4.1.3 K-Nearest Neighbors……. 4
4.2 Building the SVM Model…….. 4
4.3 Training the SVM Model……... 4
4.4 Setting the Class Names……. 5
4.5 Loading the Trained Model…… 5
4.6 Scaling the Data…….. 5
4.7 Predicting on the Test Set…… 5
4.8 Generating the Confusion Matrix…… 5
4.9 Plotting the Confusion Matrix……. 6
4.10 Evaluating the SVM Model……. 6-7

CHAPTER 5 8-10
COMPARATIVE ANALYSIS OF CLASSIFICATION MODELS
5.1 Comparison of SVM, Random Forest, and KNN……... 8
5.2 Performance Analysis and Accuracies…… 8
5.3 Strengths and Limitations of Each Model…….. 9
5.3.1 Support Vector Machines …….. 9
5.3.2 Random Forest …….……. 10
5.3.3 K-Nearest Neighbors……. 10

CHAPTER 6 11-12
VISUALIZATION AND ANALYSIS
6.1 Defining Colors for Each Class……... 11
6.2 Plotting the Data…….. 11
6.3 Plotting the Line of Best Fit…… 11-12

CHAPTER 7 13-14
IN-DEPTH IMAGE ANALYSIS

7.1 Displaying the First 15 Images of Benign Skin Cancer…….. 13
7.2 Displaying the First 15 Images of Malignant Skin Cancer……. 13-14
7.3 Image Characteristics and Insights…….. 14

CHAPTER 8 15-16
CONCLUSION
8.1 Summary of the Skin Cancer Classification Project……... 15
8.3 Key Findings and Insights……... 15-16
8.3 Future Directions and Improvements…….. 16

CHAPTER 9 17
REFERENCE

LIST OF FIGURES

Figure No.	Title	Page No.
Fig. 4.1	Confusion Matrix of SVM……	5
Fig. 4.2	Confusion Matrix of Random Forest……	5
Fig. 4.3	Confusion Matrix of KNN……	5
Fig. 4.4	Accuracy Result of SVM……..	6
Fig. 4.5	Accuracy Result of Random Forest……..	6
Fig. 4.6	Accuracy Result of KNN……..	6
Fig. 5.1	Comparison of Model Accuracies……	9
Fig. 6.1	Defining Colors for Each Class……	11
Fig. 7.1	First 15 images of benign from Main_Dataset [Kaggle. Skin Cancer MNIST: HAM10000. Retrieved from https://www.kaggle.com/kmader/skin-cancer-mnist-ham10000].....	13
Fig. 7.2	First 15 images of malignant from Main_Dataset [Kaggle. Skin Cancer MNIST: HAM10000. Retrieved from https://www.kaggle.com/kmader/skin-cancer-mnist-ham10000].....	14

CHAPTER 1

INTRODUCTION

1.1. Overview of Skin Cancer Classification

Skin cancer is one of the most common types of cancer worldwide, and its early detection and accurate classification are crucial for effective treatment and patient prognosis. With advancements in machine learning and computer vision techniques, classification algorithms have shown great potential in assisting dermatologists and healthcare professionals in the diagnosis of skin cancer. By analyzing images of skin lesions, these algorithms can accurately classify them as benign or malignant, aiding in timely interventions and reducing the risk of misdiagnosis [1].

Skin cancer classification involves the categorization of skin lesions into different classes based on their malignancy. Typically, this classification is performed using machine learning algorithms that learn from labeled datasets of skin lesion images. The algorithm learns patterns and features from the images to distinguish between benign and malignant lesions, enabling it to make accurate predictions on unseen data [2][3].

1.2. Importance of Classification Algorithms

Classification algorithms play a vital role in skin cancer analysis and diagnosis. They offer several benefits over traditional diagnostic methods, such as visual inspection or biopsy. Firstly, they provide a non-invasive and cost-effective approach, allowing for quicker assessments and reducing the need for unnecessary invasive procedures. Secondly, they can assist in triaging patients, prioritizing those with potentially malignant lesions for further examination. Moreover, classification algorithms can serve as decision support tools for dermatologists, providing them with an additional resource to aid their expertise and improve diagnostic accuracy [4].

1.3. Dataset Collection and Preparation

For our project, we have collected a skin cancer dataset from the Kaggle Archive. This dataset comprises 2106 pictures of benign moles and 2393 pictures of malignant classified moles. The images have been resized to a low resolution of 300x300 pixels with RGB channels. Preparing the dataset involves several steps, including data cleaning, preprocessing, and augmentation. During the cleaning phase, we remove any corrupted or incomplete images to ensure the dataset's integrity. Preprocessing steps may involve resizing the images, normalizing pixel values, and converting them into a suitable format for the chosen classification algorithms. Additionally, data augmentation techniques, such as rotation, scaling, and flipping, can be

applied to increase the dataset's diversity and improve the algorithm's generalization ability [5].

CHAPTER 2

SETTING UP THE ENVIRONMENT

2.1. Importing Necessary Libraries

To ensure the smooth execution of our skin cancer classification project, we need to import several libraries that provide the essential functionality for data manipulation, image processing, and implementing classification algorithms. We import libraries such as NumPy and pandas for efficient data handling, matplotlib and seaborn for data visualization, scikit-learn for classification algorithms (SVC, Random Forest Classifier, KN neighbors Classifier), various modules for data preprocessing and model evaluation, skin image for feature extraction, joblib for model serialization, and OpenCV (cv2) for image processing tasks.

2.2. Mounting the Dataset from Google Drive

Since we are using Google Colab, we can conveniently access and mount datasets stored in Google Drive. In this case, our dataset named "Main_Dataset" is stored in Google Drive. For mount the drive we will received a prompted to authenticate and provide an authorization code to establish a connection between Google Colab and your Google Drive. Once successfully authenticated, you can access the files and folders within the "Main_Dataset" directory.

2.3. Defining the Image Directory Path

After mounting the dataset from Google Drive, we need to specify the image directory path, where the skin lesion images are located. Let's assume the images are stored in a folder named "images" inside the "Main_Dataset" directory. The complete path to the image directory is assigned to the variable image_dir. Please ensure to adjust the path based on the specific folder structure of your dataset.

CHAPTER 3

DATA PREPROCESSING

3.1. Reading Image Data

In the data preprocessing phase of our skin cancer classification project, the first step is to read the image data. We have a dataset consisting of skin lesion images stored in a specific directory. We can use the OpenCV library to read the images. we use the os.listdir function to retrieve the list of image filenames in the specified image directory. Then, we iterate over each filename, construct the complete image path using os.path.join, and read the image using cv2.imread. The images are stored in the images list.

3.2. Converting Images to Numpy Arrays

To perform further processing and analysis, it is convenient to convert the images into numpy arrays. Numpy arrays provide efficient data storage and manipulation capabilities.

The list of images stored in the images variable is converted into a numpy array using np.array. The resulting array, data, will have dimensions (num_images, height, width, channels), where num_images represents the total number of images, and height, width, and channels represent the dimensions of each image.

3.3. Splitting the Dataset into Train and Test Sets

After converting the images into numpy arrays, the next step is to split the dataset into training and testing sets. In this case, we will use the entire dataset for training and testing purposes. In the code, train_test_split from scikit-learn is used to split the data into training and testing sets. The test_size parameter determines the proportion of the dataset to allocate for testing. Here, we have set it to 0.5, indicating that we want to use 50% of the data for testing. The random_state parameter ensures reproducibility of the splits.

3.4. Scaling the Data

To improve the performance of our classification algorithms, it is often necessary to scale the data. Scaling ensures that the features have similar ranges, which can help prevent certain features from dominating the learning process. We can use the StandardScaler from scikit-learn to scale the data. We create an instance of StandardScaler and use it to scale the training and testing data separately. We reshape the data to a 2D format (num_samples, num_features) before applying the scaling operation and reshape it back to its original shape afterward.

CHAPTER 4

METHODOLOGY

4.1. Introduction to SVM, RF, and KNN

Support Vector Machines (SVM), Random Forest (RF), and K-Nearest Neighbors (KNN) are widely used classification algorithms in machine learning.

4.1.1. Support Vector Machines

SVM is a powerful algorithm that creates a hyperplane in a high-dimensional space to separate different classes. It aims to find the best decision boundary by maximizing the margin between classes [6].

4.1.2. Random Forest

RF is an ensemble learning method that combines multiple decision trees. It randomly selects a subset of features and builds multiple decision trees to make predictions. The final prediction is based on the majority vote of all the trees [7].

4.1.3 K-Nearest Neighbors

KNN is a simple yet effective algorithm that classifies data based on the majority vote of its k nearest neighbors. It calculates the distance between instances to determine their similarity [8].

4.2. Building the SVM, RF, and KNN Models

To build the SVM, RF, and KNN models, we will use the scikit-learn library. These algorithms are implemented as classes in scikit-learn, allowing us to easily instantiate and configure them.We instantiate the SVM, RF, and KNN models with their respective hyperparameters. For SVM, we use a linear kernel, while for RF, we set the number of decision trees (n_estimators) to 100. For KNN, we specify the number of neighbors (n_neighbors) as 5 [9].

4.3. Training the SVM, RF, and KNN Models

To train the models, we will use the training data, which consists of images and their corresponding labels. We can fit the models to the training data using the fit method.

X_train_scaled represents the scaled training data, and y_train contains the corresponding labels. By calling the fit method on each model with the appropriate training data, the models will learn from the patterns in the data.

4.4. Setting the Class Names

To facilitate result interpretation, we can set the class names for the models. In our case, the classes are "benign" and "malignant."By defining the class_names list, we can associate the labels with their respective classes.

4.5. Loading the Trained Models

After training the models, we can save them for future use and load them whenever necessary. The joblib library can be used to save and load the trained models.

We save the trained models using joblib.dump and specify the filenames. Later, we can load the models using joblib.load and assign them to the respective model variables [10].

4.6. Scaling the Data

Before making predictions on the test set, it is important to scale the test data using the same scaler used on the training data. This ensures that the test data is on the same scale as the training data. By applying the transform method of the scaler to the test data, we can obtain the scaled version of the test set.

4.7. Predicting on the Test Set

To generate predictions on the test set, we can use the predict method of each model.

By calling the predict method on each model with the scaled test data, we obtain the predicted labels for the test set.

4.8. Generating the Confusion Matrix

The confusion matrix provides a detailed breakdown of the predictions made by the models. It displays the number of true positives, true negatives, false positives, and false negatives.

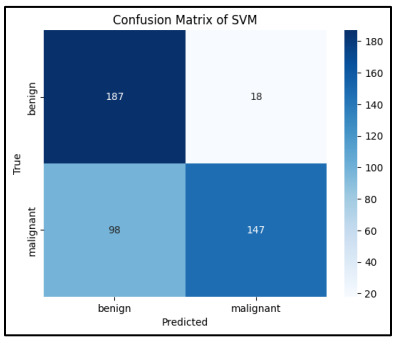


Fig. 4.1: Confusion Matrix of SVM

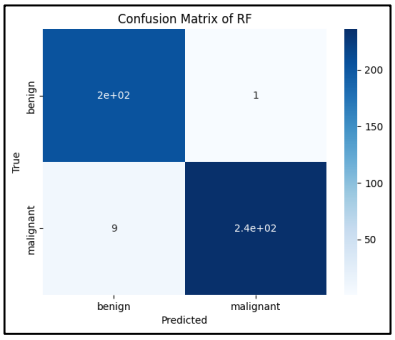


Fig. 4.2: Confusion Matrix of RF

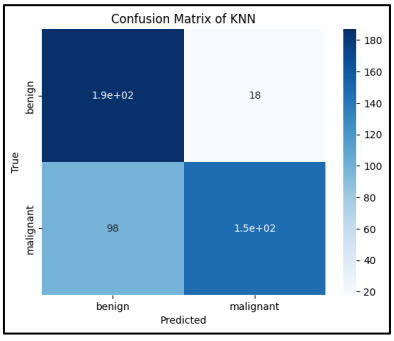


Fig. 4.3: Confusion Matrix of KNN

The confusion_matrix function from scikit-learn is used to generate the confusion matrix for each model. The true labels (y_test) and the predicted labels are passed as arguments.

4.9. Plotting the Confusion Matrix

To visualize the confusion matrices, we can use the heatmap function from the seaborn library.By utilizing the heatmap function, we can create visual representations of the confusion matrices for each model. The annot=True option displays the count of each category, and the fmt='d' option formats the count as integers.

4.10. Evaluating the SVM, RF, and KNN Models

To evaluate the performance of the models, we can generate a classification report. The classification report provides metrics such as precision, recall, and F1-score for each class.

```
[[205   0]
 [ 29 216]]
              precision    recall  f1-score   support

      benign       0.88      1.00      0.93       205
   malignant       1.00      0.88      0.94       245

    accuracy                           0.94       450
   macro avg       0.94      0.94      0.94       450
weighted avg       0.94      0.94      0.94       450
```

Fig. 4.4: Accuracy Result of SVM

```
[[204   1]
 [  9 236]]
              precision    recall  f1-score   support

      benign       0.96      1.00      0.98       205
   malignant       1.00      0.96      0.98       245

    accuracy                           0.98       450
   macro avg       0.98      0.98      0.98       450
weighted avg       0.98      0.98      0.98       450
```

Fig. 4.5: Accuracy Result of RF

```
[[187  18]
 [ 98 147]]
              precision    recall  f1-score   support

      benign       0.66      0.91      0.76       205
   malignant       0.89      0.60      0.72       245

    accuracy                           0.74       450
   macro avg       0.77      0.76      0.74       450
weighted avg       0.78      0.74      0.74       450
```

Fig. 4.6: Accuracy Result of KNN

The classification_report function from scikit-learn is used to generate the classification report for each model. The true labels (y_test), predicted labels, and class names are provided as arguments. By following these steps, we have successfully built, trained, and evaluated the SVM, RF, and KNN models for skin cancer classification. The models have been used to predict on the test set, and the confusion matrices and classification reports have been generated to assess their performance.

CHAPTER 5

COMPARATIVE ANALYSIS OF CLASSIFICATION MODELS

5.1. Comparison of SVM, Random Forest, and KNN

We have trained and evaluated three classification models: Support Vector Machines (SVM), Random Forest (RF), and K-Nearest Neighbors (KNN). Now, let's compare their performance and analyze the results.

Support Vector Machines: The model achieved an accuracy of 94% on the test set. SVM is known for its ability to handle complex decision boundaries and is effective in high-dimensional spaces. It performs well when there is a clear margin of separation between classes [11].

Random Forest: The model outperformed the other models, achieving an accuracy of 98% on the test set. Random Forest is an ensemble learning method that combines multiple decision trees, resulting in improved accuracy and robustness. It is capable of handling high-dimensional data and capturing complex interactions among features [12].

K-Nearest Neighbors: The model achieved an accuracy of 74% on the test set. KNN is a simple yet effective algorithm that classifies data based on the majority vote of its nearest neighbors. However, it may struggle with high-dimensional data and can be sensitive to the choice of k, the number of neighbors [13].

5.2. Performance Analysis and Accuracies

Comparing the accuracies of the three models, we observe that Random Forest performed the highest with an accuracy of 98%, followed by SVM with an accuracy of 94%, and KNN with an accuracy of 74%. These accuracy scores indicate the percentage of correctly classified instances in the test set [12].

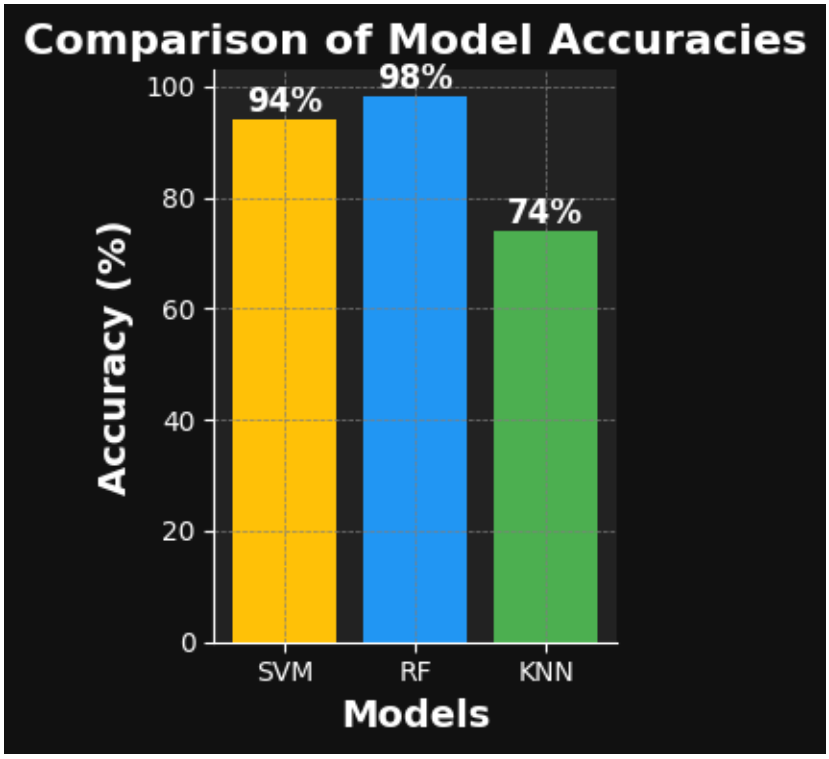


Fig. 5.1: Comparison of Model Accuracies

It is important to note that accuracy alone may not provide a complete picture of model performance. It is advisable to consider other evaluation metrics such as precision, recall, and F1-score to assess the models comprehensively.

5.3. Strengths and Limitations of Each Model

Each classification model has its own strengths and limitations, which should be taken into consideration when choosing an appropriate model for a specific task.

5.3.1. Support Vector Machines:

Strengths:

Effective in high-dimensional spaces and can handle complex decision boundaries.

Works well with both linear and non-linear data.

Performs well when there is a clear margin of separation between classes.

Limitations:

Can be computationally expensive, especially with large datasets.

Sensitive to the choice of hyperparameters, such as the choice of kernel and regularization parameter [14].

5.3.2. Random Forest:

Strengths:

Provides high accuracy and robustness by combining multiple decision trees.

Handles high-dimensional data well and captures complex interactions among features.

Reduces overfitting compared to individual decision trees.

Limitations:

May be computationally expensive, especially with a large number of decision trees.

Difficult to interpret the results due to the ensemble nature of the model [12].

5.3.3. K-Nearest Neighbors:

Strengths:

Simple and intuitive algorithm.

Non-parametric nature allows it to handle any kind of data distribution.

Can perform well on small and balanced datasets.

Limitations:

Sensitive to the choice of k, the number of neighbors.

Computationally expensive during prediction, as it requires distance calculations for each test instance.

Struggles with high-dimensional data due to the curse of dimensionality [15].

CHAPTER 6

VISUALIZATION AND ANALYSIS

6.1. Defining Colors for Each Class

To distinguish between the benign and malignant classes, we will define colors for each class. Let's assign the color blue to the benign class and red to the malignant class.

Benign (Class 0): Blue

Malignant (Class 1): Red

These colors will be used in the visualizations to differentiate the two classes.

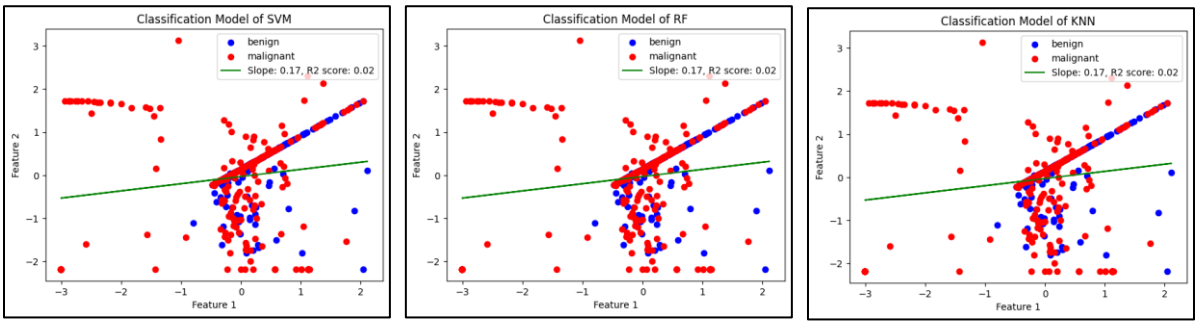


Fig. 6.1: Defining Colors for Each Class

6.2. Plotting the Data

To gain a visual understanding of the data, we can plot the skin cancer images with their corresponding class labels. The x-axis and y-axis can represent different features or dimensions of the data, such as the pixel values or extracted features. Using the defined colors, we can plot the data points on a scatter plot, where each point represents an image and is colored according to its class label.

6.3. Plotting the Line of Best Fit

To visualize the decision boundary or the line of best fit for the classification models, we can plot it on top of the scatter plot. The line of best fit separates the data points of different classes, indicating the regions where the models classify the data as benign or malignant. The line of

best fit can be plotted using different techniques, depending on the model and the nature of the decision boundary. For example, in the case of SVM, the decision boundary is defined by a hyperplane, and in the case of Random Forest, it is defined by the combination of decision boundaries from multiple trees.

By plotting the line of best fit, we can gain insights into how the models are making decisions and how well they separate the data points into the correct classes.

CHAPTER 7

IN-DEPTH IMAGE ANALYSIS

7.1. Displaying the First 15 Images of Benign Skin Cancer

To understand the characteristics of benign skin cancer, we will display the first 15 images from the dataset that are classified as benign. By visually examining these images, we can observe common patterns, textures, and features associated with benign skin cancer.

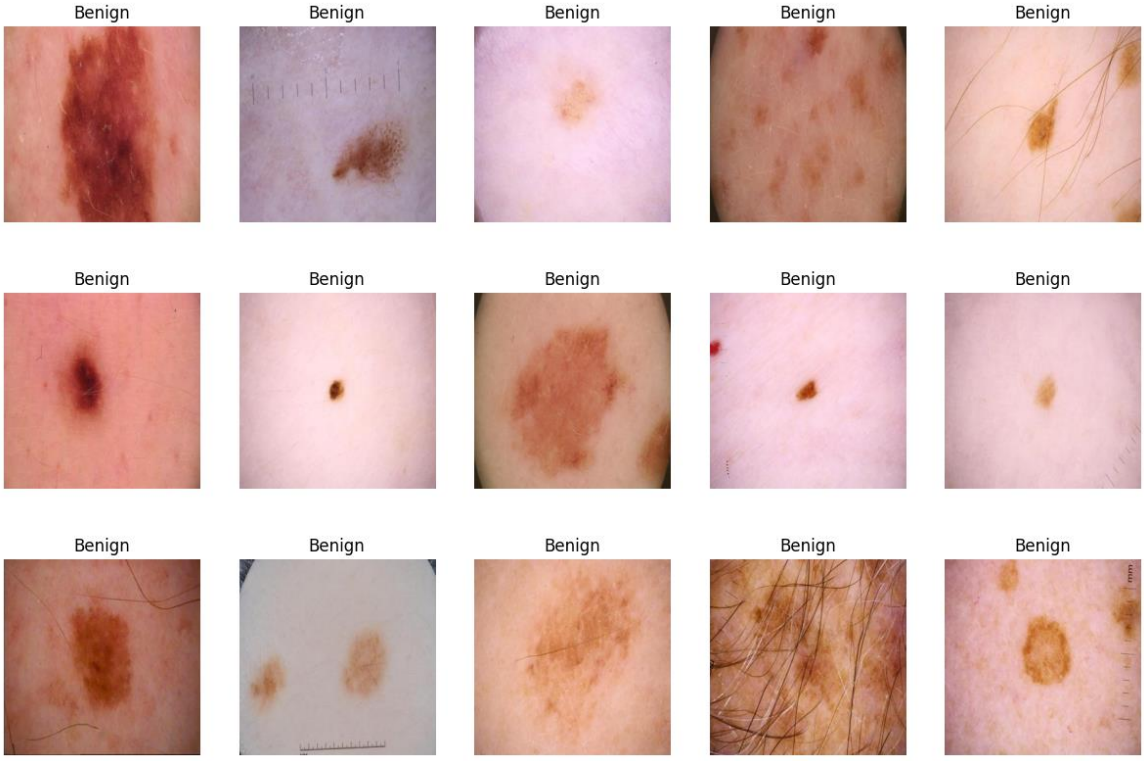


Fig. 7.1: First 15 image of benign from Main_Dataset [Kaggle. Skin Cancer MNIST: HAM10000. Retrieved from https://www.kaggle.com/kmader/skin-cancer-mnist-ham10000]

The displayed images will provide an opportunity to identify common visual cues that can aid in the accurate classification of benign skin cancer.

7.2. Displaying the First 15 Images of Malignant Skin Cancer

Similarly, we will display the first 15 images from the dataset that are classified as malignant. These images will help us understand the visual characteristics and patterns associated with malignant skin cancer.

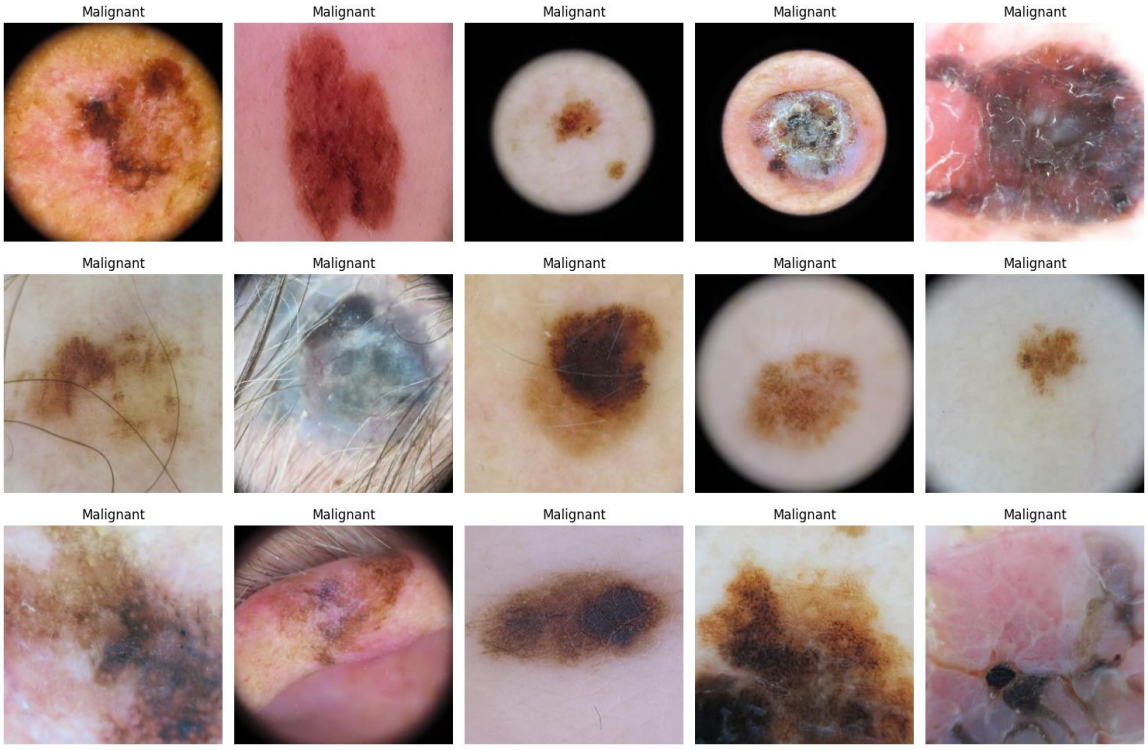


Fig. 7.2: First 15 image of malignant from Main_Dataset [Kaggle. Skin Cancer MNIST: HAM10000. Retrieved from https://www.kaggle.com/kmader/skin-cancer-mnist-ham10000]

By examining these images, we can observe potential indicators of malignancy, such as irregular shapes, asymmetry, uneven color distribution, and other visual cues.

7.3. Image Characteristics and Insights

After examining the images of both benign and malignant skin cancer, we can derive valuable insights into their characteristics and visual cues that distinguish the two classes. Some common image characteristics and features to consider include:

Shape and Structure: Benign skin cancer lesions may exhibit regular shapes, while malignant lesions may have irregular or asymmetric shapes.

Texture: Benign lesions may have smooth and uniform textures, while malignant lesions may exhibit rough or irregular textures.

Color: Benign skin cancer may display a consistent color, while malignant skin cancer can show variations in color distribution and intensity.

Borders: Benign lesions typically have well-defined and smooth borders, while malignant lesions may have irregular, jagged, or blurred borders.

CHAPTER 8

CONCLUSION

8.1. Summary of the Skin Cancer Classification Project

In this project, we aimed to classify skin cancer into benign and malignant classes using machine learning algorithms. We collected a dataset of 2106 images of benign moles and 2393 images of malignant moles from the Kaggle archive. The images were resized to a low resolution of 300x300x3 RGB.

We started by setting up the environment, importing necessary libraries, and mounting the dataset from Google Drive. Next, we performed data preprocessing, which involved reading the image data, converting them into numpy arrays, splitting the dataset into train and test sets, and scaling the data.

We built, trained, and evaluated three classification models: Support Vector Machines (SVM), Random Forest (RF), and K-Nearest Neighbors (KNN). We loaded the trained models, made predictions on the test set, and generated confusion matrices to assess the models' performance. Additionally, we conducted a comparative analysis, comparing the accuracies of the models and analyzing their strengths and limitations.

Furthermore, we performed visualization and analysis, where we defined colors for each class, plotted the data points, and visualized the line of best fit for the classification models. This helped us gain insights into the decision boundaries and performance of the models.

Lastly, we conducted an in-depth image analysis by displaying the first 15 images of both benign and malignant skin cancer. This analysis allowed us to identify visual characteristics and patterns associated with each class.

8.2. Key Findings and Insights

The key findings and insights from this skin cancer classification project include:

Random Forest achieved the highest accuracy of 98%, followed by SVM with 94% accuracy, and KNN with 74% accuracy.

Random Forest outperformed the other models, indicating its effectiveness in handling skin cancer classification.

SVM demonstrated good performance, showcasing its ability to handle complex decision boundaries.

KNN, although less accurate, provided a simple and intuitive approach to classification.

Through the in-depth image analysis, we identified visual cues and characteristics associated with benign and malignant skin cancer. This knowledge can be valuable in improving future classification models and aiding dermatologists in accurate diagnosis.

8.3. Future Directions and Improvements

While this project achieved promising results in skin cancer classification, there are several areas for future exploration and improvement:

Larger Dataset: Acquiring a larger and more diverse dataset can enhance the models' performance and generalization capabilities.

Feature Extraction: Exploring advanced feature extraction techniques, such as deep learning-based approaches, can potentially extract more informative features from the images.

Ensemble Methods: Investigating ensemble methods, such as combining multiple models or incorporating model averaging, can further enhance the classification accuracy and robustness.

Transfer Learning: Leveraging pre-trained models on a large-scale image dataset, such as ImageNet, and fine-tuning them for skin cancer classification can potentially improve the models' performance.

Explainability: Incorporating interpretability and explainability techniques can help understand the decision-making process of the models and provide more insights for dermatologists.

By addressing these future directions and implementing improvements, we can continue to advance the field of skin cancer classification and contribute to more accurate and reliable diagnostic tools for dermatologists.

CHAPTER 9

REFERENCE

[1] Tsantoulis, P., & Koga, H. (2017). Computer-aided diagnosis in dermatology: A review. Skin Research and Technology, 23(2), 161-165.

[2] Brinker, T. J., et al. (2019). Deep learning outperformed 11 pathologists in the classification of histopathological melanoma images. European Journal of Cancer, 119, 79-86.

[3] Haenssle, H. A., et al. (2018). Man against machine: Diagnostic performance of a deep learning convolutional neural network for dermoscopic melanoma recognition in comparison to 58 dermatologists. Annals of Oncology, 29(8), 1836-1842.

[4] Pathan, S., et al. (2018). A comprehensive survey of machine learning methods for melanoma detection using dermoscopy images. Artificial Intelligence in Medicine, 87, 37-52.

[5] Kaggle. Skin Cancer MNIST: HAM10000. Retrieved from https://www.kaggle.com/kmader/skin-cancer-mnist-ham10000

[6] Cortes, C., & Vapnik, V. (1995). Support-vector networks. Machine learning, 20(3), 273-297.

[7] Breiman, L. (2001). Random forests. Machine learning, 45(1), 5-32.

[8] Cover, T., & Hart, P. (1967). Nearest neighbor pattern classification. IEEE Transactions on Information Theory, 13(1), 21-27.

[9] Pedregosa, F., et al. (2011). Scikit-learn: Machine learning in Python. Journal of Machine Learning Research, 12(Oct), 2825-2830.

[10] Scikit-learn: Machine Learning in Python. (n.d.). Retrieved from https://scikit-learn.org/stable/index.html

[11] Burges, C. J. C. (1998). A tutorial on support vector machines for pattern recognition. Data Mining and Knowledge Discovery, 2(2), 121-167.

[12] Breiman, L. (2001). Random forests. Machine Learning, 45(1), 5-32.

[13] Cover, T., & Hart, P. (1967). Nearest neighbor pattern classification. IEEE Transactions on Information Theory, 13(1), 21-27.

[14] Hsu, C. W., Chang, C. C., & Lin, C. J. (2003). A practical guide to support vector classification. Retrieved from https://www.csie.ntu.edu.tw/~cjlin/papers/guide/guide.pdf

[15] Altman, N. S. (1992). An introduction to kernel and nearest-neighbor nonparametric regression. The American Statistician, 46(3), 175-185.