Skin Cancer Detection

By: T.Nethmi Sarandi Sumathipala (COHNDDS231F-008)

Higher National Diploma in Data Science (Full Time)

National Institute of Business Management, Colombo, Sri Lanka

Date of submission - 26th April 2024

A project report submitted for the partial fulfillment of the requirement of the Higher National Diploma in Data Science (Full-time) Program

DECLARATION

I hereby declare that the work presented in this project report was carried out independently by myself and have cited the work of others and given due reference diligently.

…….

Nethmi Sarandi

…….

Date

Supervisor's Certification

I certify that the above student carried out his/her project under my supervision and guidance.

……

Ms.Chamilanka Wanigasekara

……

Data

DEDICATION

This project is dedicated to my mother, who has been a wonderful supporter throughout the entire process and has patiently encouraged me to complete my work with genuine self-confidence in the months prior.

ACKNOWLEDGEMENT

I want to express my sincere gratitude to Ms.Chamilanka Wanigasekara, who served as both my supervisor and one of the best lecturers, for his insightful advice and encouragement in helping me finish my project.

Also, I want to thank my family and friends for helping me out by giving me access to the resources and information I needed to complete my assignment.

Finally, I'd like to thank all of my colleagues and everyone else who helped make this study a success.

ABSTRACT

Skin cancer is a growing public health issue globally, posing significant morbidity and mortality risks. Timely detection is paramount for enhancing patient prognosis; however, accurately diagnosing skin cancer presents challenges. This study addresses this concern by employing convolutional neural networks (CNNs) for image classification to distinguish between malignant and benign lesions. Furthermore, logistic regression analysis is utilized to pinpoint key features associated with individual skin lesions, providing valuable insights to augment diagnostic accuracy. By integrating advanced machine learning techniques with traditional statistical methods, this research aims to enhance the efficiency and precision of skin cancer diagnosis, ultimately contributing to improved patient outcomes and reduced disease burden.

Table of Contents

Table of Tables

Table of Figures

Chapter 1 – Introduction

1.1 Background

Skin cancer stands out as the most prevalent form of cancer, encompassing various types such as squamous cell carcinoma, basal cell carcinoma, and melanoma. While melanoma is less common, its propensity to invade surrounding tissue and metastasize makes it particularly lethal, accounting for most skin cancer-related deaths. Alarming trends reveal a steady rise in the incidence of both non-melanoma and melanoma skin cancers over recent decades.

Presently, between 2 and 3 million cases of non-melanoma and 132,000 cases of melanoma occur worldwide annually. Shockingly, one in every three cancer diagnoses is skin cancer, with statistics from the Skin Cancer Foundation indicating that one in every five Americans will grapple with this condition in their lifetime. These figures underscore the urgent need for heightened awareness, preventive measures, and effective management strategies to curb the escalating burden of skin cancer on a global scale.

1.2 Research Problem

cancerous and non-cancerous skin lesions remain a significant challenge in dermatology. While classification models based on lesion features, such as morphology, texture, and color, have shown promise, there is a need to identify the most essential features that contribute to accurate diagnosis and treatment decisions. Additionally, the effectiveness of these features in differentiating between various types of skin lesions requires further exploration. Addressing these issues is critical for improving diagnostic accuracy, optimizing treatment outcomes, and reducing the burden of skin cancer globally

1.3 Objective

- Classification of Skin Lesions as Cancerous or Non-Cancerous Based on Their Features.
- Identification of Essential Features Associated with Skin Lesions.

1.4 Research Questions

- How can we effectively classify skin lesions as cancer or non-cancer based on their features?
- what are the essential features associated with different types of skin lesions that contribute to accurate classification?

1.5 Scope of the research

The goal of the study is to investigate and create an effective framework for skin cancer prediction using key characteristics connected to specific skin lesions. To accurately identify between skin cancer and non-skin cancer situations, a thorough investigation of several demographic parameters, lesion features, and clinical data is included. The study encompasses an examination of several demographic groups, sampling methodologies, and data-gathering approaches to guarantee the dependability and applicability of the predictive model. The study also attempts to find new biomarkers and diagnostic indicators that can improve the precision and effectiveness of skin cancer detection. Through the pursuit of these goals, the study advances the field of dermatology and oncology's patient outcomes and medical diagnostics.

1.6 Justification of the Research

In the significant impact of skin cancer on public health and the need for improved diagnostic tools and methodologies. Skin cancer is one of the most prevalent cancers globally, with rising incidence rates and substantial morbidity and mortality. Early detection plays a crucial role in improving patient outcomes, as timely intervention can lead to better treatment efficacy and prognosis. However, accurately diagnosing skin cancer can be challenging, requiring expert evaluation and sometimes invasive procedures. Therefore, developing a reliable and non-invasive method for skin cancer prediction based on essential features associated with skin lesions is imperative. This research seeks to address this need by leveraging advanced data analysis techniques, incorporating diverse demographic and clinical factors, and exploring innovative biomarkers to enhance the accuracy and efficiency of skin cancer detection. Ultimately, the research aims to contribute to the development of accessible and effective tools for early skin cancer detection, thereby reducing the burden of the disease and improving patient care.

1.7 Proposed Work Schedule

Table 1.1: Proposed Work Schedule

Activity	Delivering date
Research Proposal	15th March
Literature review	19th March
Collecting data	21st March

Data analysis	25th March
First draft preparation	20th April
Final report submission	26th April

Chapter 2 – Literature Review

2.1 Introduction to the Research Theme

Skin cancer poses a significant public health concern globally, characterized by the abnormal growth of skin cells, primarily due to exposure to ultraviolet (UV) radiation from sunlight. While sun-exposed areas of the skin are most commonly affected, instances of skin cancer can also arise in regions not typically exposed to sunlight. This multifaceted disease encompasses three major types: basal cell carcinoma, squamous cell carcinoma, and melanoma, each with distinct characteristics and varying degrees of severity. Basal cell carcinoma and squamous cell carcinoma typically manifest as localized lesions, with relatively low metastatic potential, while melanoma presents a more aggressive phenotype, capable of rapid spread to other organs if left untreated. Given the rising incidence rates and potential for adverse outcomes associated with skin cancer, there is a pressing need for effective predictive models to aid in early detection and intervention. Leveraging advancements in machine learning and deep learning techniques, alongside comprehensive datasets containing diverse clinical and demographic variables, researchers are poised to develop robust predictive models capable of accurately identifying individuals at heightened risk of developing skin cancer. By harnessing the power of data-driven approaches and interdisciplinary collaborations, this research theme seeks to enhance our

understanding of skin cancer etiology and improve clinical outcomes through proactive prevention and early intervention strategies.

2.2 Theoretical Explanation About the Key Words in the Topic Table

Table 2.1: Theoretical Explanation of the keywords the topic

Key Words	Theoretical Definition	Reference
Skin Cancer	Skin cancer is the uncontrolled growth of abnormal skin cells. It occurs when damage to skin cells most often caused by ultraviolet radiation from sunshine or tanning beds causes skin cells to multiply rapidly and form malignant tumors.	Official WHO Information - Different types of skin cancer
Biopsied	The case of a rash, a fresh, but well-developed, lesion is chosen and if possible in a site of minimal cosmetic concern, if possible. Following a total body skin examination, the most worrisome lesion(s) are biopsied.	Official WHO Information - Different types of skin cancer

2.3 Finding by Other Research

Skin cancer represents a significant global health burden, characterized by abnormal cellular proliferation primarily triggered by exposure to ultraviolet (UV) radiation. Despite being largely preventable, its incidence continues to rise worldwide, necessitating effective strategies for early detection and intervention. Within the realm of skin cancer research, a pivotal component lies in the thorough examination and synthesis of existing literature. This literature review serves as a cornerstone in elucidating the current landscape of skin cancer prediction, offering insights into key themes, findings, and gaps in knowledge. By systematically analyzing a wealth of studies encompassing various predictive models, risk factors, and diagnostic approaches, this review aims to provide a comprehensive overview of the state-of-the-art in skin cancer prediction. Through an interdisciplinary lens, it explores the intersection of dermatology, oncology, and computational science, highlighting advancements in machine learning and deep learning techniques. By contextualizing the evolving trends and challenges in skin cancer prediction, this review sets the stage for the subsequent analysis and exploration conducted in this project. Leveraging models such as [mention specific models you plan to explore], this research aims to contribute to the development of robust predictive models and innovative strategies for early detection and prevention of skin cancer, ultimately advancing clinical practice and improving patient outcomes.

In one paper authors present a comparative analysis of machine learning algorithms for skin lesion classification using two distinct datasets: one sourced from Kaggle and the other from the Harvard Data verse (HAM10000). The Kaggle dataset focuses on distinguishing between benign and malignant lesions, while the HAM10000 dataset encompasses a broader spectrum of skin lesion types. Traditional machine learning algorithms were applied to the Kaggle dataset,

including Logistic Regression, Linear Discriminant Analysis, K-Nearest Neighbors classifier, Decision Tree Classifier, and Gaussian Naive Bayes, whereas the HAM10000 dataset utilized deep learning architectures such as Xception, VGG16, and ResNet50. Results revealed that deep learning models, particularly ResNet50, outperformed traditional machine learning algorithms in both datasets. However, the HAM10000 dataset, with its diverse range of lesion types, posed a more challenging classification task compared to the Kaggle dataset. These findings underscore the significance of dataset diversity in skin lesion classification tasks and highlight the efficacy of deep learning models, especially when trained on comprehensive datasets like HAM10000. Further research could explore ensemble techniques and data augmentation methods to enhance classification accuracy and robustness in this domain.

In another paper, the authors concentrated on utilizing theISIC_MSK- 2 dataset, which comprises biopsy-verified benign and malignant skin lesions, including melanocytic and nonmelanocytic lesions. Employing deep learning architectures, specifically VGGNet 16 and ResNet 50, they conducted an expansive analysis to estimate the performance of these models in classifying skin lesions. After comparing the results, it was set up that ResNet50 achieved the highest training accuracy for images with a resolution of 256x256, reaching an impressive accuracy of 0.9613. This finding highlights the effectiveness of deep learning models, particularly ResNet50, in directly distinguishing between benign and malignant skin lesions when trained on high-resolution images.

In one paper, a model-driven architecture deployed in the cloud, powered by deep learning algorithms, serves as the foundation for constructing predictive models aimed at enhancing the accuracy of skin cancer prediction. The study elucidates the process of model construction and deployment, focusing on the classification of dermal cell images. By leveraging deep learning

techniques embedded within the architecture, the research aims to develop robust and reliable models capable of accurately discerning between benign and malignant skin lesions. Through the utilization of cloud-based infrastructure, the study not only showcases the potential of advanced computational resources but also underscores the scalability and accessibility of the proposed approach.

In this study, the dataset encompasses approximately 25,000 images, reflecting a significant level of complexity. To streamline the analysis, a subset of 800 images is selected, with each class represented by 200 images, ensuring balanced representation across classes. The training-to-testing ratio is set at 70:30, facilitating robust model training and evaluation. Employing the MSVM (Multiclass Support Vector Machine) algorithm for classification, the study achieves notable accuracy and precision metrics, reaching approximately 96.25% and 96.32%, respectively. These results underscore the efficacy of the MSVM approach in effectively discerning between different classes of skin lesions within the dataset, thereby demonstrating its potential utility in clinical settings for accurate diagnosis and prognosis of skin cancer.

This paper presents a thorough and systematic literature review focusing on classical approaches of deep learning, including artificial neural networks (ANN), convolutional neural networks (CNN), Kohonen self-organizing neural networks (KNN), and generative adversarial neural networks (GAN), specifically in the context of skin cancer detection. Leveraging diverse types of skin lesions sourced from the International Skin Imaging Collaboration (ISIC) dataset, the review synthesizes existing research findings and methodologies employed across these deep learning paradigms. By comprehensively examining the strengths, limitations, and comparative performances of these approaches, the paper offers valuable insights into the evolution and current state-of-the-art in utilizing deep learning techniques for skin cancer detection. This

systematic review serves as a foundational resource for researchers and practitioners in dermatology and computer vision, informing future research directions and facilitating the development of more accurate and robust diagnostic tools for skin cancer detection.

The dataset utilized in this study is sourced from Kaggle, specifically curated for skin cancer detection. Comprising 10,000 images depicting various instances of skin cancer, the dataset is partitioned into training and testing subsets, with 8,000 images allocated for training purposes and 2,000 images reserved for testing and evaluation. Employing the Convolutional Neural Network (CNN) method, the study aims to leverage the power of deep learning algorithms in accurately classifying skin cancer instances within the dataset.

2.4 The Research Gap

While existing studies have explored the classification of skin lesions as cancerous or non-cancerous using CNN and logistic regression models, there remains a gap in understanding the specific features that are essential for accurate classification. Most studies have focused on the development and evaluation of model performance without explicitly identifying the key features contributing to classification decisions. Additionally, there is limited research examining the effectiveness of logistic regression in identifying essential features associated with skin lesions when combined with CNN-based classification. Addressing this gap is crucial for enhancing the interpretability of classification models and improving their clinical utility in dermatological practice.

2.5 Table for Variables, Their Definitions and Sources

Table 2.2: Variables, Their Definitions and Sources

Variables	Definitions	sources
Smoke	Whether the patient has a history of smoking or not. Smoking was inversely associated with melanoma risk, especially on the head and neck.	Skin Cancer(PAD-UFES-20) (kaggle.com) https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3535753/
Drink	Whether the patient has a history of alcohol consumption or not. Drinking alcohol can make the skin more sensitive to sunlight and vulnerable to skin cancer.	https://www.ncbi.nlm.nih.gov/pmc/articles/PMC10087036/ Skin Cancer(PAD-UFES-20) (kaggle.com)
Age	Age of the patient at the time of examination.	Skin Cancer(PAD-UFES-20) (kaggle.com)
Pesticide	Whether the patient has been exposed to pesticides or other chemicals. Pesticide exposure to increased melanoma risks.	https://www.scientificamerican.com/article/farm-pesticides-linked-to-skin-cancer/#:~:text=Previous%20research%20in%20Europe%20an

		d,of%20people%20who%20used%20less. Skin Cancer(PAD-UFES-20) (kaggle.com)
Gender	Gender of the patient.	Skin Cancer(PAD-UFES-20) (kaggle.com)
skin_cancer_history	History of skin cancer in the patient's family. Family history of cancer is often used to evaluate an individual's risk for developing a particular malignancy.	https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4679454/ Skin Cancer(PAD-UFES-20) (kaggle.com)
cancer_history	History of cancer in the patient's family.	Skin Cancer(PAD-UFES-20) (kaggle.com)
has_piped_water	Indicates whether the location or area of the patient's residence has access to piped water or not.	Skin Cancer(PAD-UFES-20) (kaggle.com)
has_sewage_system	Indicates whether the location or area of the patient's residence has a proper sewage system or not.	Skin Cancer(PAD-UFES-20) (kaggle.com)
Fitspatrick	Skin tolerance to sunlight.	Skin Cancer(PAD-UFES-20) (kaggle.com)
Region	The area of the body where the lesion or wound has been examined.	Skin Cancer(PAD-UFES-20) (kaggle.com)

diameter_1	Primary diameter of the lesion or wound.	Skin Cancer(PAD-UFES-20) (kaggle.com)
diameter_2	Secondary diameter of the lesion or wound.	Skin Cancer(PAD-UFES-20) (kaggle.com)
Diagnostic	The type of lesion or wound is diagnosed.	Skin Cancer(PAD-UFES-20) (kaggle.com)
Itch	Whether the lesion or wound has itched or not. Itching is not the usual symptom of skin cancer, and an itch, in general, is more commonly caused by something else," says Ilene Rothman, MD, formerly of Roswell Park Comprehensive Cancer Center.	Skin Cancer(PAD-UFES-20) (kaggle.com)
Grew	Whether the size of the lesion or wound has grown or not.	Skin Cancer(PAD-UFES-20) (kaggle.com)
Hurt	Whether the lesion or wound has hurt or not.	Skin Cancer(PAD-UFES-20) (kaggle.com)
Changed	Whether the appearance of the lesion or wound has changed or not.	Skin Cancer(PAD-UFES-20) (kaggle.com)
Bleed	Whether the lesion or wound has bled or not.	Skin Cancer(PAD-UFES-20) (kaggle.com)
Elevation	Description of the of the lesion or wound relative to the skin surface of the patient.	Skin Cancer(PAD-UFES-20) (kaggle.com)

Biopsied	Whether the lesion or wound has been biopsied or not.	Skin Cancer(PAD-UFES-20) (kaggle.com)

2.6 Chapter conclusion

In this chapter, we embarked on a comprehensive journey through the landscape of skin cancer prediction, delving into key themes, theoretical underpinnings, empirical findings, and research gaps identified in the existing literature. Commencing with an introduction to the research theme, we underscored the urgency and significance of skin cancer prediction in the context of its escalating global health burden, emphasizing the critical need for early detection and intervention. Subsequently, we synthesized a wealth of findings from other researchers, highlighting the efficacy of deep learning models in achieving superior performance in lesion classification tasks, particularly in distinguishing between benign and malignant lesions. Despite these advancements, our review also uncovered notable research gaps, particularly in the areas of dataset diversity, model interpretability, and real-world applicability of predictive models. Finally, we presented a comprehensive table detailing key variables, their definitions, and sources, offering readers a holistic understanding of the variables under consideration and their relevance to skin cancer prediction research. In conclusion, this chapter lays the groundwork for our research endeavor, providing a foundational framework for subsequent analysis and exploration. Through a synthesis of existing knowledge and identification of critical gaps, we are poised to contribute to the advancement of skin cancer prediction methodologies, ultimately striving towards improved clinical practice and patient outcomes in the fight against skin cancer.

Chapter 3 – Data Preparation Process - Data Pre-processing and Data Wrangling

3.1 Data Cleaning

The research integrates two datasets, with one specifically comprising images relevant to the study. This dataset includes samples of both skin cancer and non-skin cancer images. By incorporating diverse visual data, the study aims to develop robust models capable of distinguishing between different dermatological conditions effectively.

The other dataset comprises essential features associated with each skin lesion, including factors such as smoking habits, alcohol consumption, age, pesticide exposure, gender, skin cancer history, cancer history, access to piped water, presence of sewage system, Fitzpatrick skin type, primary and secondary lesion diameters, diagnostic status, itchiness, growth pattern, pain sensation, observed changes, bleeding propensity, and lesion elevation. Patient identifiers (patient_id, lesion_id, img_id) and background information (background_father, background_mother) have been excluded to streamline the dataset, focusing solely on attributes directly relevant to skin lesions.

3.1.1 Removed unwanted columns.

The dataset is depicted in Figure 3.1 before the data cleaning. Five columns were removed following the cleaning of the original 26 columns. There were 21 columns left in the final dataset. (For more details, see Figure 3.2.) " patient_id, lesion_id, img_id, background_father, background_mother" were the columns that were removed.

```
data.head()
```

	patient_id	lesion_id	smoke	drink	background_father	background_mother	age	pesticide	gender	skin_cancer_history	...	diameter_2	diagnostic	itch
0	PAT_1516	1765	NaN	NaN	NaN	NaN	8	NaN	NaN	NaN	...	NaN	NEV	FALSE
1	PAT_46	881	False	False	POMERANIA	POMERANIA	55	False	FEMALE	True	...	5.0	BCC	TRUE
2	PAT_1545	1867	NaN	NaN	NaN	NaN	77	NaN	NaN	NaN	...	NaN	ACK	TRUE
3	PAT_1989	4061	NaN	NaN	NaN	NaN	75	NaN	NaN	NaN	...	NaN	ACK	TRUE
4	PAT_684	1302	False	True	POMERANIA	POMERANIA	79	False	MALE	True	...	5.0	BCC	TRUE

5 rows × 26 columns

```
data.shape
```

```
(2298, 26)
```

Figure 3.1: Before removing unwanted columns.

```
data.head()
```

	smoke	drink	age	pesticide	gender	skin_cancer_history	cancer_history	has_piped_water	has_sewage_system	fitspatrick	...	diameter_1	diameter_2	di
0	NaN	NaN	8	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	
1	False	False	55	False	FEMALE	True	True	True	True	3.0	...	6.0	5.0	
2	NaN	NaN	77	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	
3	NaN	NaN	75	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	
4	False	True	79	False	MALE	True	False	False	False	1.0	...	5.0	5.0	

5 rows × 21 columns

```
data.shape
```

```
(2298, 21)
```

Figure 3.2: After removing unwanted columns

3.1.2 Removed null values

The dataset is depicted in Figure 3.3 before the remove null values. Then I remove null values in my dataset. (For more details, see Figure 3.4.)

```
data.isnull().sum()
```

```
patient_id              0
lesion_id               0
smoke                 804
drink                 804
background_father      818
background_mother      822
age                     0
pesticide             804
gender                804
skin_cancer_history   804
cancer_history        804
has_piped_water       804
has_sewage_system     804
fitspatrick           804
region                  0
diameter_1            804
diameter_2            804
diagnostic              0
itch                    0
grew                    0
hurt                    0
changed                 0
bleed                   0
elevation               0
img_id                  0
biopsed                 0
dtype: int64
```

Figure 3.3 before the remove null values

```
data.dropna(inplace=True)
```

```
data.isnull().sum()
```

```
smoke                   0
drink                   0
age                     0
pesticide               0
gender                  0
skin_cancer_history     0
cancer_history          0
has_piped_water         0
has_sewage_system       0
fitspatrick             0
region                  0
diameter_1              0
diameter_2              0
diagnostic              0
itch                    0
grew                    0
hurt                    0
changed                 0
bleed                   0
elevation               0
biopsed                 0
dtype: int64
```

Figure 3.4 After the remove null values

Chapter 4 – Methodology

4.1 Introduction

This chapter delves into the prediction of skin cancer presence, focusing on key features associated with individual skin lesions. It encompasses an analysis aimed at distinguishing between skin cancer and non-skin cancer conditions. The chapter provides insight into the population demographics, sampling procedures, data collection techniques, and sources of data utilized in the analysis. By meticulously examining these components, the chapter establishes a

foundation for accurate skin cancer prediction, contributing to advancements in medical diagnostics and treatment strategies.

4.2 Population, Sample and Sampling Technique

Population- Here we have targeted the cancer.

Sample- Skin cancers and key features associated with individual skin lesions. (According to the Kaggle website)

Sampling technique - Not specifically mentioned in the source I obtained the data set.

4.3 Type of Data to be Collected and Data Sources

The data for this study came from a website called "Kaggle". This datasets provides a comprehensive overview of skin cancer and essential features associated with each skin lesion.

4.4 Data Collection Tools and Plan

As previously stated, the datasets were obtained through "Kaggle." As a result, because the data had already been prepared and published, data collection techniques were no longer required. Once the datasets were received, any unwanted variables were removed. Chapter three demonstrates this.

4.5 Method of Data Analysis

- Classification of Skin Lesions as Cancerous or Non-Cancerous Based on Their Features.
    - Convolutional Neural Networks(CNN)
- Identification of Essential Features Associated with Skin Lesions.
    - Logistic Regression

Chapter 5 – Data Analysis, Visualization and Interpretation

5.1 Data Analysis

Data analysis is the process of using tools to organize and classify the collected data. In the study, the acquired data was examined using the required techniques. This methodology was used to predict skin cancer. The data collection contains twenty-one columns.

Table 5.1: Variables and Data Types

Variables	Data type
Smoke	Object
Drink	Object
Age	Integer
Pesticide	Object
Gender	Object
Skin_cancer_history	Object
Cancer_history	Object
Has_piped_water	Object
Fitspatrick	float
Region	Object
Diameter_1	Float
Diameter_2	float
Diagnostic	Object
Itch	Object

Grew	Object
Hurt	Object
Changed	Object
Bleed	Object
Elevation	Object
biopsed	Bool

5.2 Model Building

Image Classification

This study's task revolves around skin cancer classification utilizing Convolutional Neural Networks (CNNs), a specialized neural network architecture renowned for its effectiveness in image recognition and classification tasks. The dataset employed in this research, sourced from the online platform "Kaggle," comprises skin cancer and non-skin cancer images. To commence the investigation, essential libraries are imported, and the dataset is loaded. It is pertinent to note that the dataset is already partitioned into distinct sets for training and testing purposes, obviating the necessity for further data-splitting procedures. This streamlined dataset acquisition process expedites subsequent model development and evaluation phases.

```
import numpy as np
import tensorflow as tf
import matplotlib.pyplot as plt
import time
import opendatasets as od
```

```
od.download("https://www.kaggle.com/datasets/mdismielhossenabir/skin-cancer-or-not-skin-cancer-image-datasets")
#nethmisarandi
#d0e64c7acad746d87d71f87a47044b21
```

```
Please provide your Kaggle credentials to download this dataset. Learn more: http://bit.ly/kaggle-creds
Your Kaggle username: nethmisarandi
Your Kaggle Key: ··········
Downloading skin-cancer-or-not-skin-cancer-image-datasets.zip to ./skin-cancer-or-not-skin-cancer-image-datasets
100%|██████████| 3.21M/3.21M [00:00<00:00, 51.4MB/s]
```

```
batch_size=32
image_size=(128,128)   #input image size
```

```
train_data_dir="/content/skin-cancer-or-not-skin-cancer-image-datasets/train/train"
test_data_dir="/content/skin-cancer-or-not-skin-cancer-image-datasets/test/test"
```

Figure 5.1: Load the dataset

The study utilizes images from the train_data_dir to construct the training dataset, while also reserving a subset for validation purposes. Similarly, images from the test_data_dir are utilized to create the testing dataset. This segregation of data into distinct training and validation sets is crucial for effectively

training the model while also ensuring its generalization ability can be assessed accurately. By partitioning the data this way, the study lays the groundwork for robust model training and evaluation, essential for accurate skin cancer classification.

```
train_data=tf.keras.utils.image_dataset_from_directory(train_data_dir,batch_size=batch_size,image_size=image_size,subset="training",
                                                       validation_split=0.1,seed=42)
```

```
Found 210 files belonging to 2 classes.
Using 189 files for training.
```

```
validation_data=tf.keras.utils.image_dataset_from_directory(train_data_dir,batch_size=batch_size,image_size=image_size,subset="validation",
                                                       validation_split=0.1,seed=42)
```

```
Found 210 files belonging to 2 classes.
Using 21 files for validation.
```

```
test_data=tf.keras.utils.image_dataset_from_directory(test_data_dir,batch_size=batch_size,image_size=image_size)
```

```
Found 11 files belonging to 2 classes.
```

Figure 5.2: Segregation of data into distinct training and validation sets

Then I snippet plots a selection of images along with their corresponding labels from the training dataset.

Figure 5.3: Images of dataset

After that, I did feature scaling on the datasets by mapping a lambda function to each element in the datasets. This lambda function divides each pixel value of the images by 255, effectively scaling the pixel values between 0 and 1. This normalization process helps in stabilizing the training process and improves the convergence of the model during training. It's a common preprocessing step in deep learning tasks, ensuring consistent behavior and performance across different datasets.

```
#feature scaling
train_data=train_data.map(lambda x,y:(x/255,y))
validation_data=validation_data.map(lambda x,y:(x/255,y))
test_data=test_data.map(lambda x,y:(x/255,y))
```

```
for image,label in train_data.take(1):
  for i in range(1):
    print(image)
```

```
tf.Tensor(
[[[[0.6668658  0.54137564 0.5335325 ]
   [0.66939336 0.5456955  0.53665745]
   [0.659758   0.5460325  0.5293505 ]
   ...
   [0.45951286 0.40261948 0.40261948]
   [0.3920037  0.33377758 0.32712927]
   [0.38019302 0.3213695  0.30980393]]

  [[0.6691487  0.5366877  0.53112745]
   [0.66939336 0.54255766 0.5338861 ]
   [0.6585367  0.5388174  0.5239298 ]
   ...
   [0.44168186 0.38360476 0.37822163]
```

Figure 5.4: Feature Scaling

Then I added data augmentation to my image classification model using TensorFlow's Keras Sequential API with layers for flipping, rotating, and zooming. This aimed to boost accuracy in identifying skin cancer images.

```
data_augmentation=tf.keras.Sequential(
    [tf.keras.layers.RandomFlip("horizontal",input_shape=(128,128,3)),
     tf.keras.layers.RandomRotation(0.2),
     tf.keras.layers.RandomZoom(0.2),]
)
```

Figure 5.5: Data augmentation

Then build a model architecture. The model incorporates data augmentation to enhance dataset diversity before passing through multiple convolutional layers with increasing feature depth and

max-pooling layers for downsampling. Dropout and BatchNormalization layers are included to mitigate overfitting and ensure stable training. Following convolutional operations, the feature maps are flattened and passed through densely connected layers for classification. The model culminates with a single neuron output layer utilizing a sigmoid activation function, yielding a probability score indicating the likelihood of an input image representing skin cancer.

```
#model building
model=tf.keras.models.Sequential()

model.add(data_augmentation)

model.add(tf.keras.layers.Conv2D(32,kernel_size=3,activation="relu"))
model.add(tf.keras.layers.MaxPooling2D())

model.add(tf.keras.layers.Conv2D(64,kernel_size=3,activation="relu"))
model.add(tf.keras.layers.MaxPooling2D())

model.add(tf.keras.layers.Conv2D(128,kernel_size=3,activation="relu"))
model.add(tf.keras.layers.MaxPooling2D())

model.add(tf.keras.layers.Dropout(0.2))
model.add(tf.keras.layers.BatchNormalization())

model.add(tf.keras.layers.Flatten())

model.add(tf.keras.layers.Dense(128,activation="relu"))
model.add(tf.keras.layers.Dense(128,activation="relu"))
model.add(tf.keras.layers.Dense(32,activation="relu"))

model.add(tf.keras.layers.Dense(1,activation="sigmoid"))
```

Figure 5.6: Model architecture

```
model.summary()
```

```
Model: "sequential_1"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 sequential (Sequential)     (None, 128, 128, 3)       0

 conv2d (Conv2D)             (None, 126, 126, 32)      896

 max_pooling2d (MaxPooling2  (None, 63, 63, 32)        0
 D)

 conv2d_1 (Conv2D)           (None, 61, 61, 64)        18496

 max_pooling2d_1 (MaxPoolin  (None, 30, 30, 64)        0
 g2D)

 conv2d_2 (Conv2D)           (None, 28, 28, 128)       73856

 max_pooling2d_2 (MaxPoolin  (None, 14, 14, 128)       0
 g2D)

 dropout (Dropout)           (None, 14, 14, 128)       0

 batch_normalization (Batch  (None, 14, 14, 128)       512
 Normalization)

 flatten (Flatten)           (None, 25088)             0

 dense (Dense)               (None, 128)               3211392

 dense_1 (Dense)             (None, 128)               16512
```

Figure 5.7: Model summary

Next, I compile the model for training by defining its optimization algorithm, loss function, and evaluation metrics. It utilizes the Adam optimizer for efficient gradient descent, Binary Focal Cross entropy as the loss function, which is good at handling class imbalance in binary classification tasks, and accuracy as the metric for evaluating model performance.

```
model.compile(optimizer=tf.keras.optimizers.Adam(),
              loss=tf.keras.losses.BinaryFocalCrossentropy(),
              metrics=["accuracy"])
```

Figure 5.8: Model Compile

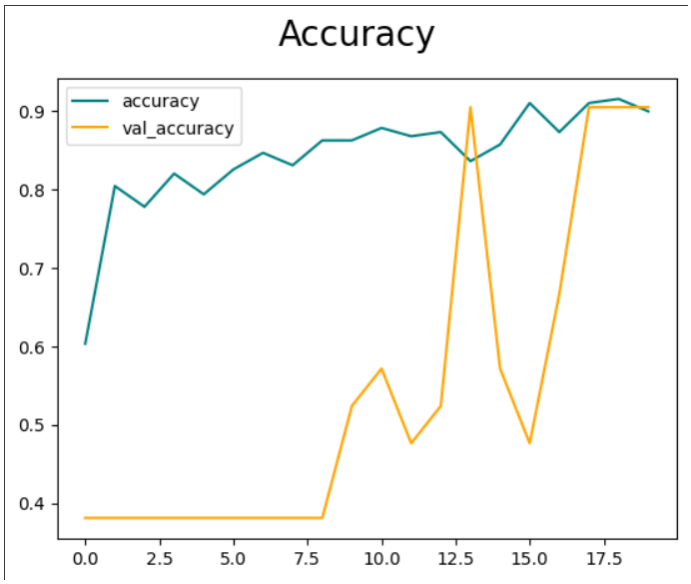
I used separate validation data to validate the model's performance while training it for 20 epochs on the training dataset.

```
history=model.fit(train_data,epochs=20,
                  validation_data=validation_data)
```

```
Epoch 1/20
6/6 [==============================] - 9s 1s/step - loss: 0.1962 - accuracy: 0.6032 - val_loss: 0.2183 - val_accuracy: 0.3810
Epoch 2/20
6/6 [==============================] - 6s 943ms/step - loss: 0.1249 - accuracy: 0.8042 - val_loss: 1.1677 - val_accuracy: 0.3810
Epoch 3/20
6/6 [==============================] - 7s 1s/step - loss: 0.1419 - accuracy: 0.7778 - val_loss: 0.2906 - val_accuracy: 0.3810
Epoch 4/20
6/6 [==============================] - 7s 1s/step - loss: 0.1324 - accuracy: 0.8201 - val_loss: 0.7157 - val_accuracy: 0.3810
Epoch 5/20
6/6 [==============================] - 6s 993ms/step - loss: 0.1055 - accuracy: 0.7937 - val_loss: 0.3486 - val_accuracy: 0.3810
Epoch 6/20
6/6 [==============================] - 6s 953ms/step - loss: 0.0938 - accuracy: 0.8254 - val_loss: 0.2274 - val_accuracy: 0.3810
Epoch 7/20
6/6 [==============================] - 6s 934ms/step - loss: 0.0995 - accuracy: 0.8466 - val_loss: 0.2774 - val_accuracy: 0.3810
Epoch 8/20
6/6 [==============================] - 6s 937ms/step - loss: 0.0962 - accuracy: 0.8307 - val_loss: 0.3768 - val_accuracy: 0.3810
Epoch 9/20
6/6 [==============================] - 6s 938ms/step - loss: 0.0910 - accuracy: 0.8624 - val_loss: 0.3750 - val_accuracy: 0.3810
Epoch 10/20
6/6 [==============================] - 6s 963ms/step - loss: 0.0673 - accuracy: 0.8624 - val_loss: 0.1831 - val_accuracy: 0.5238
Epoch 11/20
6/6 [==============================] - 7s 1s/step - loss: 0.0654 - accuracy: 0.8783 - val_loss: 0.1596 - val_accuracy: 0.5714
Epoch 12/20
6/6 [==============================] - 6s 938ms/step - loss: 0.0661 - accuracy: 0.8677 - val_loss: 0.2876 - val_accuracy: 0.4762
Epoch 13/20
6/6 [==============================] - 6s 912ms/step - loss: 0.0703 - accuracy: 0.8730 - val_loss: 0.2071 - val_accuracy: 0.5238
Epoch 14/20
6/6 [==============================] - 6s 971ms/step - loss: 0.0747 - accuracy: 0.8360 - val_loss: 0.1048 - val_accuracy: 0.9048
Epoch 15/20
```

Figure 5.9: Model Train

Next, I plot the training and validation accuracy and loss over the epochs.



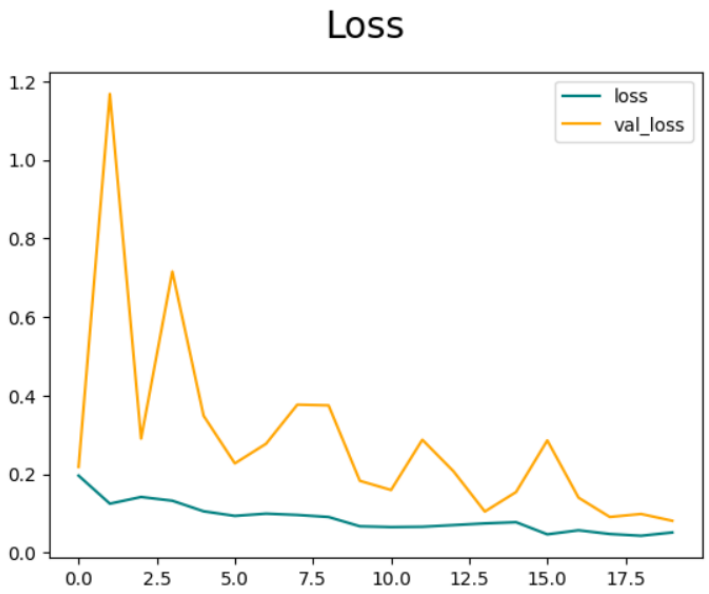


Figure 5.10: Training and validation accuracy and loss over the epochs

Next, evaluates the trained model's performance using precision, recall, and binary accuracy metrics on the test dataset. It initializes instances of metric classes for precision, recall, and binary accuracy.

```
#evaluate the model
precision=tf.keras.metrics.Precision()
recall=tf.keras.metrics.Recall()
accuracy=tf.keras.metrics.BinaryAccuracy()
```

```
for batch in test_data.as_numpy_iterator():
  x,y =batch
  yhat=model.predict(x)
  precision.update_state(y,yhat)
  recall.update_state(y,yhat)
  accuracy.update_state(y,yhat)
```

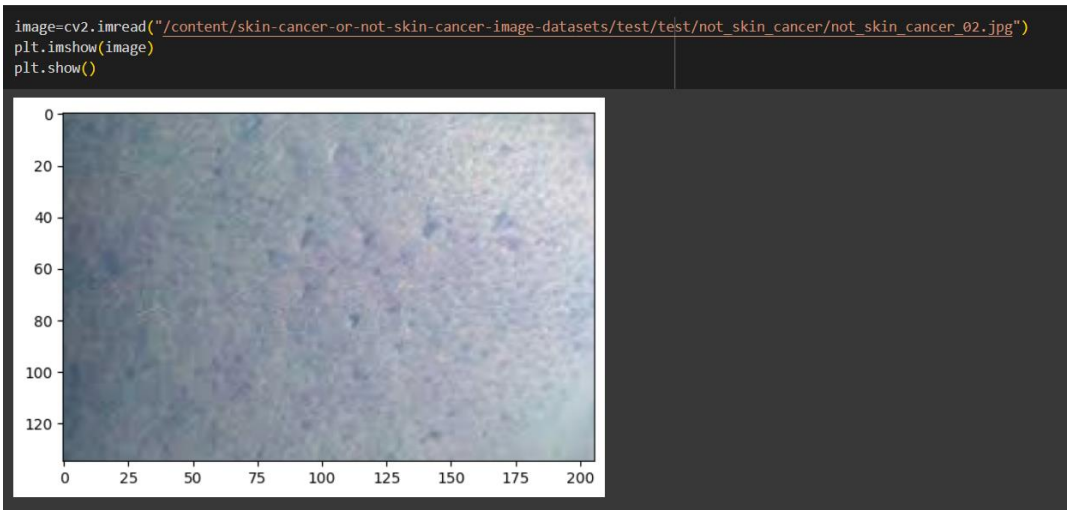
```
1/1 [==============================] - 0s 446ms/step
```

```
precision.result()
recall.result()
accuracy.result()
```

```
<tf.Tensor: shape=(), dtype=float32, numpy=0.90909094>
```

Figure 5.11: Precision, recall, and binary accuracy result

Finally, I test an image using a trained model for skin cancer classification. The model prediction y_hat is based on the defined threshold of 0.5. If the prediction probability is greater than or equal to 0.5, it prints the class name corresponding to index 1 in the class_names list, which represents skin_cancer. Otherwise, it prints the class name corresponding to index 0, which represents not_skin_cancer. This approach effectively maps the model's output to the appropriate class label based on the threshold. Based on the prediction result y_hat of 0.479, the model's output probability suggests that the image is classified as not containing skin cancer.



```
np.expand_dims(scaled_image,0).shape
```

```
(1, 128, 128, 3)
```

```
y_hat=model.predict(np.expand_dims(scaled_image,0))
```

```
1/1 [==============================] - 0s 28ms/step
```

```
y_hat
```

```
array([[0.47974414]], dtype=float32)
```

```
class_names
```

```
['not_skin_cancer', 'skin_cancer']
```

```
if y_hat >=0.5:
  print(class_names[1])
else:
    print(class_names[0])
```

```
not skin cancer
```

Figure 5.12: Predict the label

Predict the skin lesion

This study employs logistic regression to predict whether a skin lesion has been biopsied, a critical decision-making process in dermatological practice. Initially, essential libraries for data manipulation and machine learning are imported. Following data loading, cleaning procedures are undertaken to ensure data quality. Exploratory data analysis reveals insights into the distribution of key features associated with skin lesions, particularly focusing on the prevalence of biopsied lesions. Subsequently, categorical variables are encoded into numerical labels using LabelEncoder to prepare for logistic regression modeling.

	smoke	drink	age	pesticide	gender	skin_cancer_history	cancer_history	has_piped_water	has_sewage_system	fitspatrick	...	diameter_1	diameter_2	dia
1	0	0	55	0	0	1	1	1	1	3.0	...	6.0	5.0	
4	0	1	79	0	1	1	0	0	0	1.0	...	5.0	5.0	
6	0	1	52	0	0	0	1	1	1	3.0	...	15.0	10.0	
7	0	0	74	1	0	0	0	0	0	1.0	...	15.0	10.0	
9	0	1	58	1	0	1	1	1	1	1.0	...	9.0	7.0	

5 rows × 21 columns

Figure 5.13: Label Encoding.

Then I separated the dataset into input features (x) and the target variable (y) for the machine learning model.

```
x = df.drop(columns = ['biopsed'] )
y =df['biopsed']
```

Figure 5.14:Separated the dataset.

The train_test_split function from the scikit-learn library split the dataset into training and testing sets for both the input features (x) and the target variable (y).

```
x_train, x_test, y_train, y_test = train_test_split(x,y, test_size = 0.2, random_state = 0,stratify=y)
```

```
print(x_train.shape)
print(x_test.shape)
```

```
(1195, 20)
(299, 20)
```

Figure 5.15: Train test split

Then instantiate a logistic regression model object and fit the model to the training data.

```
model=LogisticRegression()
```

```
model.fit(x_train,y_train)
```

Figure 5.16: Model building

The logistic regression model's training on the designated training data and subsequent prediction on the test set, the accuracy of the model is evaluated. The model successfully predicts

the biopsied status of skin lesions with an accuracy of 88.29%. This indicates that the model accurately classifies approximately 88.29% of the skin lesions as either biopsied or non-biopsied based on the provided features. Moreover, to assess the model's performance comprehensively, both training and testing accuracies are calculated. The training accuracy stands at 88.95%, suggesting that the model performs consistently well on the data used for training. Similarly, the testing accuracy, which aligns closely with the training accuracy at 88.29%, underscores the model's capability to generalize effectively to unseen data.

```
y_Pred = model.predict(x_test)
y_Pred
```

```
array([1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1,
       0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 0, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       0, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0,
       1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1])
```

```
accuracy = accuracy_score(y_test, y_Pred)
print('Accuracy:', accuracy)
```

```
Accuracy: 0.882943143812709
```

```
#calculating the classification accuracies
print("Training Accuracy", model.score(x_train, y_train))
print("Testing Accuracy", model.score(x_test, y_test))
```

```
Training Accuracy 0.8895397489539749
Testing Accuracy 0.882943143812709
```

Figure 5.17: Predict the test data

Chapter 6 – Discussion and Recommendations

6.1 Discussion

The prediction of skin cancer presence has garnered significant attention, particularly concerning the identification of key features associated with individual skin lesions. In Section 2.3 of this study, reference is made to research conducted by the International Journal of Advances in Engineering and Management (IJAEM) concerning the ISIC_MSK-2 dataset, which comprises both benign and malignant skin lesions, confirmed via biopsy, encompassing melanocytic and non-melanocytic lesions. The researchers employed Convolutional Neural Networks (CNNs) in their analysis, leveraging established architectures such as VGG16 and RESNet50.

Moreover, the literature suggests avenues for further investigation, particularly in the realm of patient stratification based on their risk of developing skin cancer.

6.2 Recommendations

Skin cancer represents a significant public health challenge, with increasing incidence rates globally, particularly in regions with high levels of ultraviolet (UV) radiation exposure such as the United Kingdom. Melanoma, in particular, poses a considerable threat due to its aggressive nature and potential for metastasis. However, the encouraging aspect is that a vast majority of skin cancer cases are preventable through the adoption of sun-safe practices. This paper aims to outline evidence-based recommendations for minimizing the risk of skin cancer by protecting the skin from UV radiation.

Recommendations:

- Sunscreen Application: The regular application of broad-spectrum sunscreen with a sun protection factor (SPF) of 30 or higher is essential. Sunscreen should be applied generously to all exposed skin at least 15 minutes before sun exposure and reapplied every two hours or immediately after swimming or sweating.
- Seeking Shade: Minimize direct sun exposure, especially during peak hours between 10 a.m. and 4 p.m. Seek shade under umbrellas, trees, or other forms of shelter to reduce UV exposure.
- Protective Clothing: Wear protective clothing, including wide-brimmed hats, long-sleeved shirts, and sunglasses with UV protection. Clothing with a tight weave offers better protection against UV radiation.
- Avoid Tanning Beds: Avoid the use of tanning beds and sunlamps, as they emit harmful UV radiation that increases the risk of skin cancer, including melanoma.
- Regular Skin Checks: Perform regular self-examinations of the skin to detect any changes in moles, freckles, or other skin lesions. Seek prompt medical attention if any suspicious changes are observed.
- Professional Skin Screenings: Schedule regular skin cancer screenings with a dermatologist, especially for individuals with a history of sunburns, extensive UV exposure, or a family history of skin cancer. For more effective sun protection, select clothing with an ultraviolet protection factor (UPF) label.

6.3 Conclusion

Through the integration of logistic regression and Convolutional Neural Networks (CNN), this study aimed to identify essential features associated with skin lesions and accurately distinguish between skin cancer and non-skin cancer lesions. By incorporating a wide range of factors including demographic information, environmental exposures,

lesion characteristics, and medical history, the logistic regression model provided insights into the complex interplay of these variables in skin lesion diagnosis.

The CNN model, on the other hand, effectively utilized image data to classify skin lesions as cancerous or non-cancerous based on visual features extracted from the lesions. By leveraging the power of deep learning, the CNN demonstrated high accuracy in lesion classification, particularly in cases where visual cues played a significant role in diagnosis.

The combined approach of logistic regression and CNN allowed for a holistic analysis of skin lesion classification, integrating both clinical and visual information to improve diagnostic accuracy. By identifying key features associated with skin cancer and non-skin cancer lesions, this study contributes to a better understanding of the factors influencing lesion diagnosis and highlights the importance of multidimensional approaches in dermatological practice.

Appendices

01.

```
import numpy as np
import tensorflow as tf
import matplotlib.pyplot as plt
import time
import opendatasets as od
```

02.

```
od.download("https://www.kaggle.com/datasets/mdismielhossenabir/skin-cancer-or-not-skin-cancer-image-datasets")
#nethmisarandi
#d0e64c7acad746d87d71f87a47044b21
```

```
Please provide your Kaggle credentials to download this dataset. Learn more: http://bit.ly/kaggle-creds
Your Kaggle username: nethmisarandi
Your Kaggle Key: ··········
Downloading skin-cancer-or-not-skin-cancer-image-datasets.zip to ./skin-cancer-or-not-skin-cancer-image-datasets
100%|          | 3.21M/3.21M [00:00<00:00, 51.4MB/s]
```

+ Code + Text

```
batch_size=32
image_size=(128,128)   #input image size
```

```
train_data_dir="/content/skin-cancer-or-not-skin-cancer-image-datasets/train/train"
test_data_dir="/content/skin-cancer-or-not-skin-cancer-image-datasets/test/test"
```

03.

```
train_data=tf.keras.utils.image_dataset_from_directory(train_data_dir,batch_size=batch_size,image_size=image_size,subset="training",
                                                       validation_split=0.1,seed=42)
```

```
Found 210 files belonging to 2 classes.
Using 189 files for training.
```

```
validation_data=tf.keras.utils.image_dataset_from_directory(train_data_dir,batch_size=batch_size,image_size=image_size,subset="validation",
                                                       validation_split=0.1,seed=42)
```

```
Found 210 files belonging to 2 classes.
Using 21 files for validation.
```

```
test_data=tf.keras.utils.image_dataset_from_directory(test_data_dir,batch_size=batch_size,image_size=image_size)
```

```
Found 11 files belonging to 2 classes.
```

04.

```
class_names=train_data.class_names
class_names
```

```
['not_skin_cancer', 'skin_cancer']
```

```
for image_batch,label_batch in train_data.take(1):
  print(image_batch.shape)
  print(label_batch.shape)
```

```
(32, 128, 128, 3)
(32,)
```

05.

```
plt.figure(figsize=(10,4))
for image,label in train_data.take(1):
  for i in range(10):
    ax=plt.subplot(2,5,i+1)
    plt.imshow(image[i].numpy().astype('uint8'))
    plt.title(class_names[label[i]])
    plt.axis('off')
```

06.

```
#feature scaling
train_data=train_data.map(lambda x,y:(x/255,y))
validation_data=validation_data.map(lambda x,y:(x/255,y))
test_data=test_data.map(lambda x,y:(x/255,y))
```

```
for image,label in train_data.take(1):
  for i in range(1):
    print(image)
```

```
tf.Tensor(
[[[[0.6668658  0.54137564 0.5335325 ]
   [0.66939336 0.5456955  0.53665745]
   [0.659758   0.5460325  0.5293505 ]
   ...
   [0.45951286 0.40261948 0.40261948]
   [0.3920037  0.33377758 0.32712927]
   [0.38019302 0.3213695  0.30980393]]

  [[0.6691487  0.5366877  0.53112745]
   [0.66939336 0.54255766 0.5338861 ]
   [0.6585367  0.5388174  0.5239298 ]
   ...
   [0.44168186 0.38360476 0.37822163]
   [0.39947242 0.34087992 0.33109382]
   [0.4151692  0.35862857 0.34694088]]

  [[0.6719114  0.53219604 0.52814615]
```

07.

```
data_augmentation=tf.keras.Sequential(
    [tf.keras.layers.RandomFlip("horizontal",input_shape=(128,128,3)),
     tf.keras.layers.RandomRotation(0.2),
     tf.keras.layers.RandomZoom(0.2),]
)
```

08.

```
#model building
model=tf.keras.models.Sequential()

model.add(data_augmentation)

model.add(tf.keras.layers.Conv2D(32,kernel_size=3,activation="relu"))
model.add(tf.keras.layers.MaxPooling2D())

model.add(tf.keras.layers.Conv2D(64,kernel_size=3,activation="relu"))
model.add(tf.keras.layers.MaxPooling2D())

model.add(tf.keras.layers.Conv2D(128,kernel_size=3,activation="relu"))
model.add(tf.keras.layers.MaxPooling2D())

model.add(tf.keras.layers.Dropout(0.2))
model.add(tf.keras.layers.BatchNormalization())

model.add(tf.keras.layers.Flatten())

model.add(tf.keras.layers.Dense(128,activation="relu"))
model.add(tf.keras.layers.Dense(128,activation="relu"))
model.add(tf.keras.layers.Dense(32,activation="relu"))

model.add(tf.keras.layers.Dense(1,activation="sigmoid"))
```

09.

```
model.summary()
```

```
Model: "sequential_1"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 sequential (Sequential)     (None, 128, 128, 3)       0

 conv2d (Conv2D)             (None, 126, 126, 32)      896

 max_pooling2d (MaxPooling2  (None, 63, 63, 32)        0
 D)

 conv2d_1 (Conv2D)           (None, 61, 61, 64)        18496

 max_pooling2d_1 (MaxPoolin  (None, 30, 30, 64)        0
 g2D)

 conv2d_2 (Conv2D)           (None, 28, 28, 128)       73856

 max_pooling2d_2 (MaxPoolin  (None, 14, 14, 128)       0
 g2D)

 dropout (Dropout)           (None, 14, 14, 128)       0

 batch_normalization (Batch  (None, 14, 14, 128)       512
 Normalization)

 flatten (Flatten)           (None, 25088)             0

 dense (Dense)               (None, 128)               3211392

 dense_1 (Dense)             (None, 128)               16512

 dense_2 (Dense)             (None, 32)                4128
```

10.

```
model.compile(optimizer=tf.keras.optimizers.Adam(),
              loss=tf.keras.losses.BinaryFocalCrossentropy(),
              metrics=["accuracy"])
```

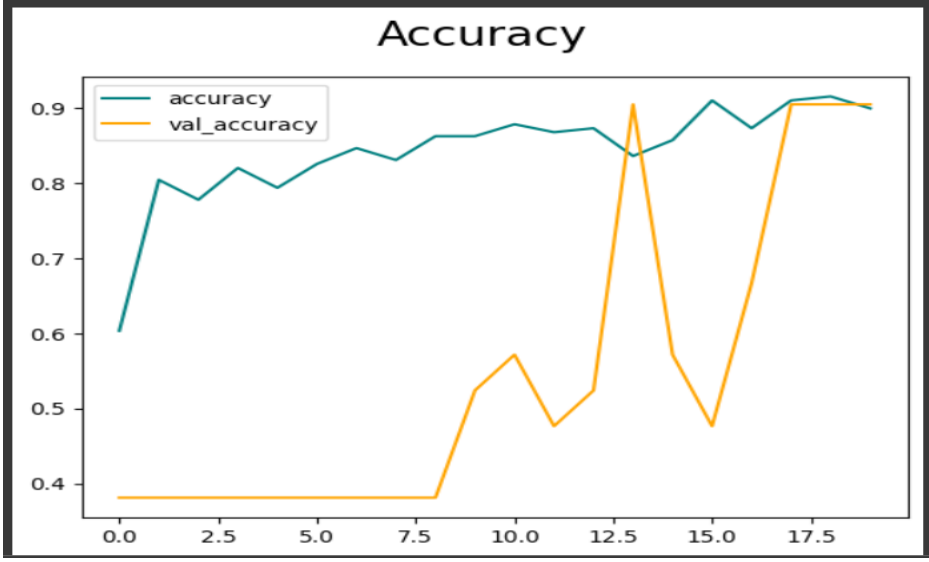
11.

```
history=model.fit(train_data,epochs=20,
                  validation_data=validation_data)
```

```
Epoch 1/20
6/6 [==============================] - 9s 1s/step - loss: 0.1962 - accuracy: 0.6032 - val_loss: 0.2183 - val_accuracy: 0.3810
Epoch 2/20
6/6 [==============================] - 6s 943ms/step - loss: 0.1249 - accuracy: 0.8042 - val_loss: 1.1677 - val_accuracy: 0.3810
Epoch 3/20
6/6 [==============================] - 7s 1s/step - loss: 0.1419 - accuracy: 0.7778 - val_loss: 0.2906 - val_accuracy: 0.3810
Epoch 4/20
6/6 [==============================] - 7s 1s/step - loss: 0.1324 - accuracy: 0.8201 - val_loss: 0.7157 - val_accuracy: 0.3810
Epoch 5/20
6/6 [==============================] - 6s 993ms/step - loss: 0.1055 - accuracy: 0.7937 - val_loss: 0.3486 - val_accuracy: 0.3810
Epoch 6/20
6/6 [==============================] - 6s 953ms/step - loss: 0.0938 - accuracy: 0.8254 - val_loss: 0.2274 - val_accuracy: 0.3810
Epoch 7/20
6/6 [==============================] - 6s 934ms/step - loss: 0.0995 - accuracy: 0.8466 - val_loss: 0.2774 - val_accuracy: 0.3810
Epoch 8/20
6/6 [==============================] - 6s 937ms/step - loss: 0.0962 - accuracy: 0.8307 - val_loss: 0.3768 - val_accuracy: 0.3810
Epoch 9/20
6/6 [==============================] - 6s 938ms/step - loss: 0.0910 - accuracy: 0.8624 - val_loss: 0.3750 - val_accuracy: 0.3810
Epoch 10/20
6/6 [==============================] - 6s 963ms/step - loss: 0.0673 - accuracy: 0.8624 - val_loss: 0.1831 - val_accuracy: 0.5238
Epoch 11/20
6/6 [==============================] - 7s 1s/step - loss: 0.0654 - accuracy: 0.8783 - val_loss: 0.1596 - val_accuracy: 0.5714
Epoch 12/20
6/6 [==============================] - 6s 938ms/step - loss: 0.0661 - accuracy: 0.8677 - val_loss: 0.2876 - val_accuracy: 0.4762
Epoch 13/20
6/6 [==============================] - 6s 912ms/step - loss: 0.0703 - accuracy: 0.8730 - val_loss: 0.2071 - val_accuracy: 0.5238
Epoch 14/20
6/6 [==============================] - 6s 971ms/step - loss: 0.0747 - accuracy: 0.8360 - val_loss: 0.1048 - val_accuracy: 0.9048
Epoch 15/20
```

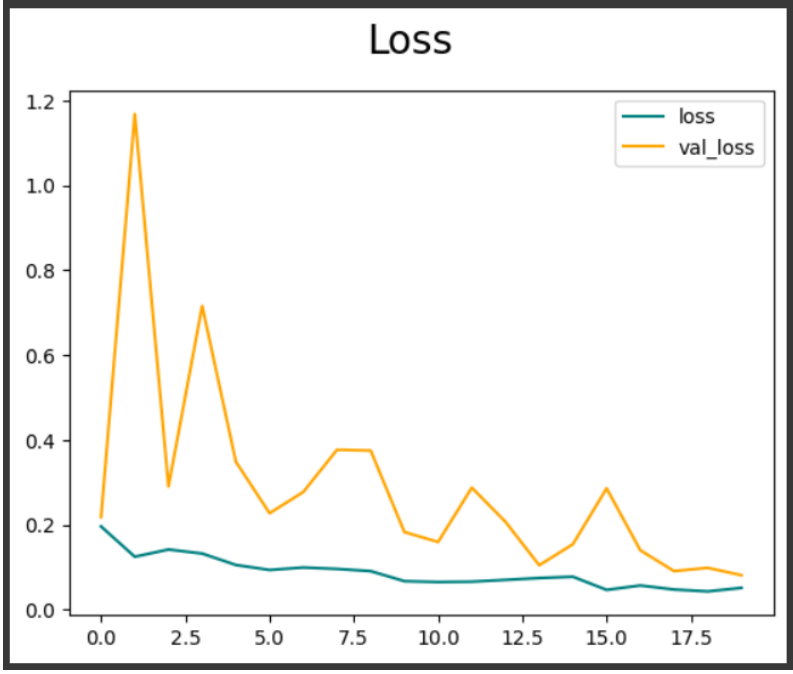
12.

```
fig=plt.figure()
plt.plot(history.history["accuracy"],color="teal",label="accuracy")
plt.plot(history.history["val_accuracy"],color="orange",label="val_accuracy")
fig.suptitle("Accuracy",fontsize=20)
plt.legend()
plt.show()
```



13.

```
fig=plt.figure()
plt.plot(history.history["loss"],color="teal",label="loss")
plt.plot(history.history["val_loss"],color="orange",label="val_loss")
fig.suptitle("Loss",fontsize=20)
plt.legend()
plt.show()
```



14.

```
#evaluate the model
precision=tf.keras.metrics.Precision()
recall=tf.keras.metrics.Recall()
accuracy=tf.keras.metrics.BinaryAccuracy()
```

```
for batch in test_data.as_numpy_iterator():
  x,y =batch
  yhat=model.predict(x)
  precision.update_state(y,yhat)
  recall.update_state(y,yhat)
  accuracy.update_state(y,yhat)
```

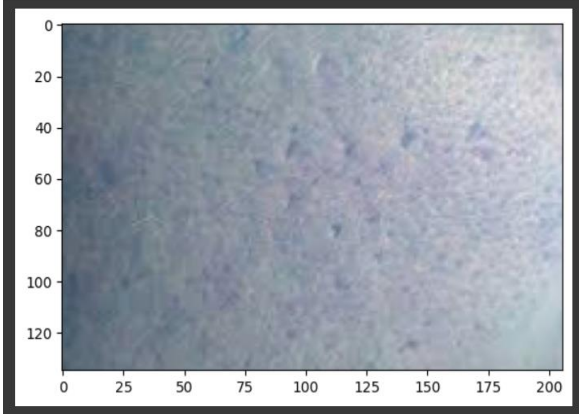
```
1/1 [==============================] - 0s 446ms/step
```

```
precision.result()
recall.result()
accuracy.result()
```

```
<tf.Tensor: shape=(), dtype=float32, numpy=0.90909094>
```

15.

```
image=cv2.imread("/content/skin-cancer-or-not-skin-cancer-image-datasets/test/test/not_skin_cancer/not_skin_cancer_02.jpg")
plt.imshow(image)
plt.show()
```




```
resized_image=tf.image.resize(image,image_size)
scaled_image=resized_image/225
```

16.

```
resized_image=tf.image.resize(image,image_size)
scaled_image=resized_image/225
```

```
scaled_image
```

```
<tf.Tensor: shape=(128, 128, 3), dtype=float32, numpy=
array([[[0.40225926, 0.4688414 , 0.5044815 ],
        [0.3880382 , 0.45470485, 0.49026042],
        [0.3830961 , 0.4497628 , 0.48737848],
        ...,
        [0.85939807, 0.8686168 , 0.9085532 ],
        [0.91122913, 0.9067951 , 0.9512291 ],
        [0.8964433 , 0.89203584, 0.933353  ]],

       [[0.42086807, 0.48444444, 0.52381945],
        [0.40931952, 0.47598618, 0.5122709 ],
        [0.40659708, 0.47326374, 0.5095485 ],
```

17.

```
np.expand_dims(scaled_image,0).shape
```

```
(1, 128, 128, 3)
```

```
y_hat=model.predict(np.expand_dims(scaled_image,0))
```

```
1/1 [==============================] - 0s 28ms/step
```

```
y_hat
```

```
array([[0.47974414]], dtype=float32)
```

```
class_names
```

```
['not_skin_cancer', 'skin_cancer']
```

18.

```
if y_hat >=0.5:
  print(class_names[1])
else:
    print(class_names[0])
```

```
not_skin_cancer
```

19.

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sb
import os
sb.set()

from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.preprocessing import OneHotEncoder, StandardScaler
from sklearn.compose import make_column_transformer
from sklearn.pipeline import make_pipeline
from sklearn.metrics import r2_score
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score
from sklearn import metrics
from sklearn.metrics import roc_curve
from sklearn.metrics import recall_score, confusion_matrix, precision_score, f1_score, accuracy_score, classification_report
from sklearn.preprocessing import MinMaxScaler
```

20.

```
data=pd.read_csv("metadata.csv")
```

```
data.head()
```

	patient_id	lesion_id	smoke	drink	background_father	background_mother	age	pesticide	gender	skin_cancer_history	...	diameter_2	diagnostic	itch
0	PAT_1516	1765	NaN	NaN	NaN	NaN	8	NaN	NaN	NaN	...	NaN	NEV	FALSE
1	PAT_46	881	False	False	POMERANIA	POMERANIA	55	False	FEMALE	True	...	5.0	BCC	TRUE
2	PAT_1545	1867	NaN	NaN	NaN	NaN	77	NaN	NaN	NaN	...	NaN	ACK	TRUE
3	PAT_1989	4061	NaN	NaN	NaN	NaN	75	NaN	NaN	NaN	...	NaN	ACK	TRUE
4	PAT_684	1302	False	True	POMERANIA	POMERANIA	79	False	MALE	True	...	5.0	BCC	TRUE

5 rows × 26 columns

21.

```
data.shape
```

```
(2298, 26)
```

22.

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2298 entries, 0 to 2297
Data columns (total 26 columns):
 #   Column               Non-Null Count  Dtype  
---  ------               --------------  -----  
 0   patient_id           2298 non-null   object 
 1   lesion_id            2298 non-null   int64  
 2   smoke                1494 non-null   object 
 3   drink                1494 non-null   object 
 4   background_father    1480 non-null   object 
 5   background_mother    1476 non-null   object 
 6   age                  2298 non-null   int64  
 7   pesticide            1494 non-null   object 
 8   gender               1494 non-null   object 
 9   skin_cancer_history  1494 non-null   object 
 10  cancer_history       1494 non-null   object 
 11  has_piped_water      1494 non-null   object 
 12  has_sewage_system    1494 non-null   object 
 13  fitspatrick          1494 non-null   float64
 14  region               2298 non-null   object 
 15  diameter_1           1494 non-null   float64
 16  diameter_2           1494 non-null   float64
 17  diagnostic           2298 non-null   object 
 18  itch                 2298 non-null   object 
 19  grew                 2298 non-null   object 
 20  hurt                 2298 non-null   object 
 21  changed              2298 non-null   object 
 22  bleed                2298 non-null   object 
 23  elevation            2298 non-null   object 
 24  img_id               2298 non-null   object 
 25  biopsed              2298 non-null   bool   
dtypes: bool(1), float64(3), int64(2), object(20)
memory usage: 451.2+ KB
```

23.

```
data.isnull().sum()
```

```
patient_id              0
lesion_id               0
smoke                 804
drink                 804
background_father      818
background_mother      822
age                     0
pesticide             804
gender                804
skin_cancer_history   804
cancer_history        804
has_piped_water       804
has_sewage_system     804
fitspatrick           804
region                  0
diameter_1            804
diameter_2            804
diagnostic              0
itch                    0
grew                    0
hurt                    0
changed                 0
bleed                   0
elevation               0
img_id                  0
biopsed                 0
dtype: int64
```

24.

```
data.drop(columns=['patient_id'],inplace=True)
```

```
data.drop(columns=['lesion_id'],inplace=True)
```

```
data.drop(columns=['img_id'],inplace=True)
```

```
data.drop(columns=['background_father'],inplace=True)
```

```
data.drop(columns=['background_mother'],inplace=True)
```

25.

```
data.dropna(inplace=True)
```

```
data.isnull().sum()
```

```
smoke                 0
drink                 0
age                   0
pesticide             0
gender                0
skin_cancer_history   0
cancer_history        0
has_piped_water       0
has_sewage_system     0
fitspatrick           0
region                0
diameter_1            0
diameter_2            0
diagnostic            0
itch                  0
grew                  0
hurt                  0
changed               0
bleed                 0
elevation             0
biopsed               0
dtype: int64
```

```
data.shape
```

```
(1494, 21)
```

26.

```
import pandas as pd

# Assuming 'data' is your DataFrame
for col in data.columns:
    if data[col].dtype == 'object' or data[col].dtype == 'bool':
        unique_values = pd.unique(data[col])
        print(f'{col} : {unique_values}\n')
```

```
smoke : [False True]

drink : [False True]

pesticide : [False True]

gender : ['FEMALE' 'MALE']

skin_cancer_history : [True False]

cancer_history : [True False]

has_piped_water : [True False]

has sewage system : [True False]
```

27.

```
from sklearn.preprocessing import LabelEncoder
#creating an encoder
le=LabelEncoder()
```

```
from sklearn.preprocessing import LabelEncoder

def object_to_int(dataframe_series):
    if dataframe_series.dtype == 'object':
        dataframe_series = LabelEncoder().fit_transform(dataframe_series)
    elif dataframe_series.dtype == 'bool':
        dataframe_series = dataframe_series.astype(int)
    return dataframe_series
```

```
df = data.apply(lambda x: object_to_int(x))
```

28.

```
df.head()
```

	smoke	drink	age	pesticide	gender	skin_cancer_history	cancer_history	has_piped_water	has_sewage_system	fitspatrick	...	diameter_1	diameter_2	dia
1	0	0	55	0	0	1	1	1	1	3.0	...	6.0	5.0	
4	0	1	79	0	1	1	0	0	0	1.0	...	5.0	5.0	
6	0	1	52	0	0	0	1	1	1	3.0	...	15.0	10.0	
7	0	0	74	1	0	0	0	0	0	1.0	...	15.0	10.0	
9	0	1	58	1	0	1	1	1	1	1.0	...	9.0	7.0	

5 rows × 21 columns

29.

```
x = df.drop(columns = ['biopsed'] )
y =df['biopsed']
```

30.

```
x_train, x_test, y_train, y_test = train_test_split(x,y, test_size = 0.2, random_state = 0,stratify=y)
```

```
print(x_train.shape)
print(x_test.shape)
```

```
(1195, 20)
(299, 20)
```

31.

```
model=LogisticRegression()
```

```
model.fit(x_train,y_train)
```

32.

```
y_Pred = model.predict(x_test)
y_Pred
```

```
array([1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1,
       0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 0, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       0, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0,
       1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1])
```

```
accuracy = accuracy_score(y_test, y_Pred)
print('Accuracy:', accuracy)
```

```
Accuracy: 0.882943143812709
```

```
#calculating the classification accuracies
print("Training Accuracy", model.score(x_train, y_train))
print("Testing Accuracy", model.score(x_test, y_test))
```

```
Training Accuracy 0.8895397489539749
Testing Accuracy 0.882943143812709
```

References

https://www.kaggle.com/datasets/mahdavi1202/skin-cancer

https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8945332/

https://ijaem.net/issue_dcp/Prediction%20of%20Skin%20Cancer%20using%20Machine%20Learning.pdf

https://www.sciencedirect.com/science/article/pii/S2352914819302047

https://ieeexplore.ieee.org/document/9393198

https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8160886/

https://sist.sathyabama.ac.in/sist_naac/documents/1.3.4/b.e-ece-batchno-181.pdf

https://www.mdpi.com/2075-4418/13/11/1911