

# INDEX

Sr. No.	Content	Page No.
1.	Abstract	6
2.	Introduction 2.1 About skin cancer 2.2 Motivation 2.3 Objectives	7-13
3.	Research Methodology 3.1 Literature Review 3.2 Description of Dataset 3.3 Data Analysis	14-25
4.	Implementation 4.1 Implementing Algorithms 4.2 Image Processing 4.3 Model Architecture 4.4 Result	26-39
5.	Conclusion	40
6.	Bibliography	41-42

## ABSTRACT

The goal of this project is to improve skin cancer detection by developing an integrated system that integrates human expertise, artificial intelligence (AI), and image analysis approaches. The study suggests a cooperative strategy between dermatologists and AI systems and highlights the need of early diagnosis, particularly in the case of melanoma.

Driven by the increased prevalence of skin cancer, the study attempts to increase the accuracy of diagnosis, save lives by early identification, and offer easily available medical treatments. One approach to achieving these objectives is the fusion of human talents and artificial intelligence capabilities.

As part of the project, prior research is examined, data is gathered and pre-processed, multiple data mining algorithms are used, and an image processing model is created. After extensive pre-processing of the dataset, lesion prediction is performed using machine learning methods (KNN, Naïve Bayes, Random Forest, Decision Tree, SVM, and Logistic Regression), with Decision Tree proving to be the most well-suited technique.

A Convolutional Neural Network (CNN) can categorise photos of skin cancer with a high accuracy of 99% when used for image processing. The creation of a web application that enhances skin cancer detection by fusing artificial intelligence (AI) with human expertise is the research's capstone.

In conclusion, by fusing image processing with AI and human intelligence, the research successfully tackles the urgent problem of skin cancer detection, providing a potential option for accurate and accessible healthcare.

## INTRODUCTION

The prevalent and potentially fatal condition referred to as skin cancer affects millions of people worldwide. Early identification of skin cancer is crucial for successful treatment and better patient outcomes [1]. The goal of this research is to advance a comprehensive skin cancer detection system employing image analysis techniques to assist dermatologists and individuals discover potential skin cancer lesions at premature stage.

We must comprehend what skin cancer is? to be able to detect it. The abnormal development and expansion of skin cells is what causes skin cancer [2]. There will be various stages [2]. Although there are many different forms of cancer, basal cell carcinoma (BCC), squamous cell carcinoma (SCC), and melanoma are three most prevalent forms. Each of them is named for a particular cell that causes it [2]. The most prevalent kind, BCC, develops on the hands, neck, and other sun-exposed parts of the body. SCC is kind of like BCC as it can be found on areas that are exposed to sun but can be found in scars, skin ulcers or areas that have been exposed to radiation. Melanoma is not much common as BCC and SCC, but during the past few decades, it has become a substantial public health challenge [3].

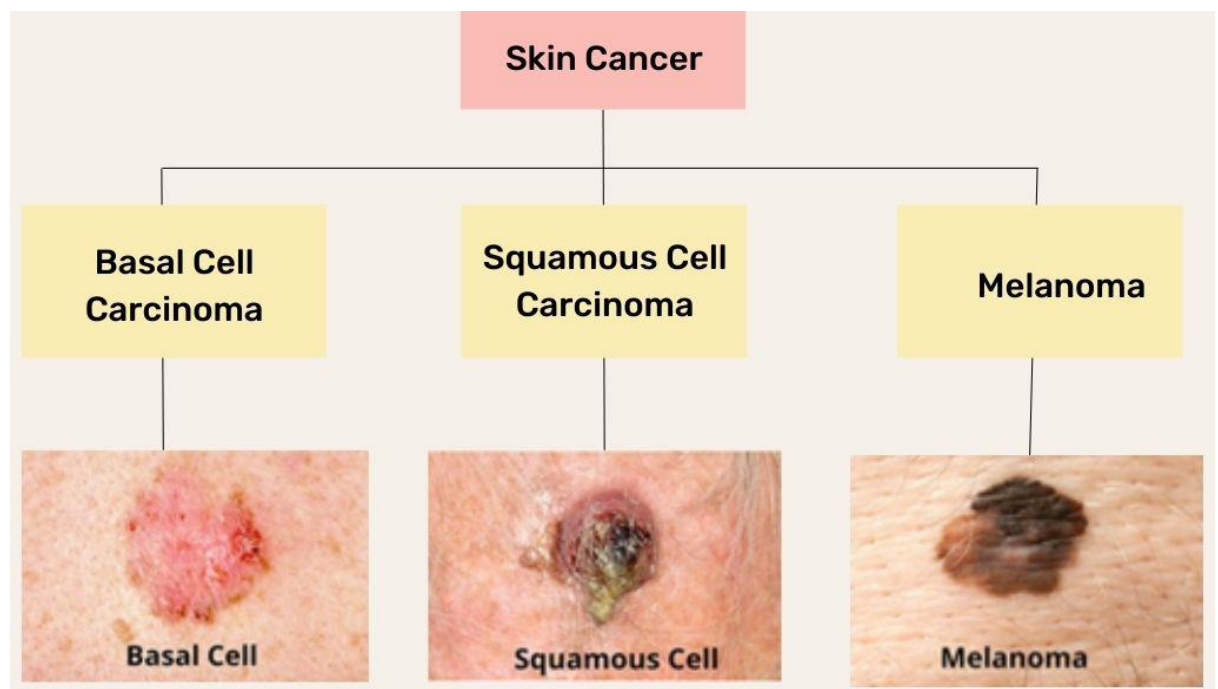


Fig.1: Types of skin cancer

Benign and malignant are terminologies that used to describe the behaviour and nature of growths or tumours in the body. Tumours or growths that are benign are not cancer-causing, avoid spreading to other bodily parts, and do not infiltrate neighbouring tissues. Common examples of benign are moles (nevus), skin cysts, skin tags, etc. Malignant growths or tumours are cancerous and possess the capacity to grow by intruding on nearby tissues to various parts of the body. Melanoma, SCC, and BCC are typical cases of malignancy [4].

The terms Benign and Malignant are further divided into different classes and some of these classes include both benign and malignant lesions. these classes are:

1. bkl (Benign Keratosis-Like Lesions): These are non-cancerous and do not have the potential to become cancerous.
2. nv (Melanocytic Nevi): It is also known as moles, generally come under benign skin lesions. However, in some cases, they may undergo changes that could potentially be cancerous. Regular checking is advised.
3. df (Dermatofibroma): Dermatofibromas are benign skin lesions and are not cancerous.
4. vasc (Vascular Lesions): Vascular lesions are generally come in benign also they are not cancerous.
5. bcc (Basal Cell Carcinoma): It is a general class of skin cancer. It is typically less aggressive than melanoma but still considered malignant.
6. akiec (Actinic Keratosis and Intraepithelial Carcinoma): These are the pre-cancerous lesions. While they are not invasive cancers, they can develop into SCC.
7. mel (Melanoma): It's malignant skin lesion. As studied earlier, it is invasive and if treatment is delayed, the condition may spread to other body areas.

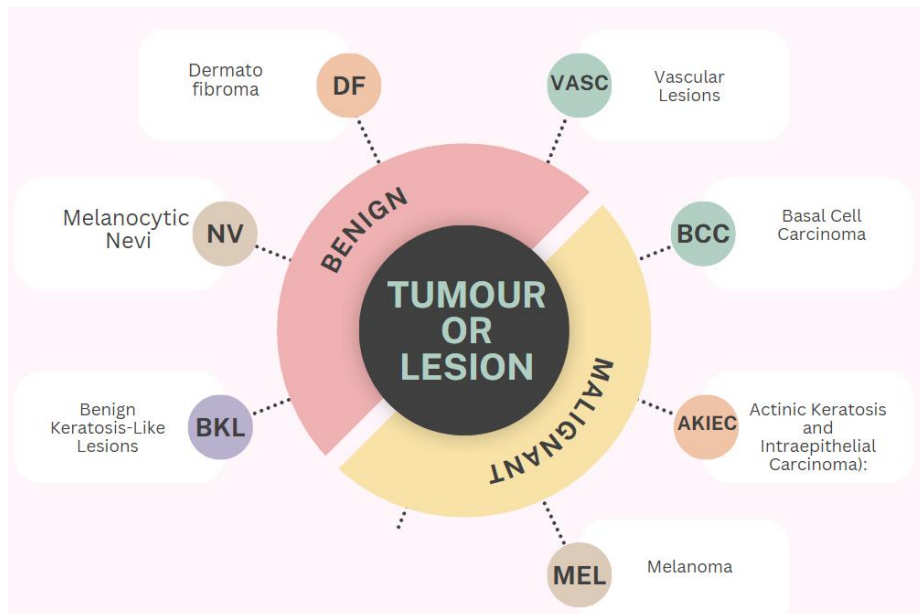


Fig.2: Classification of Tumour or Lesion

The rising incidence rates and mortality associated with melanoma have underscored the need for early detection and prevention efforts [5]. While dermoscopy has proven to enhance diagnosis validity of open-eye examinations [3–6], dermatologists as well as other healthcare professionals who have received training in dermoscopic techniques still struggle to achieve high levels of sensitivity in identifying melanoma [6, 7].

As per the World Health Organization (WHO) report, every year, 2-3 Million people have non-melanoma cancer while 132000 people are diagnosed with melanoma cancer. One in every three persons diagnosed with cancer has skin cancer.

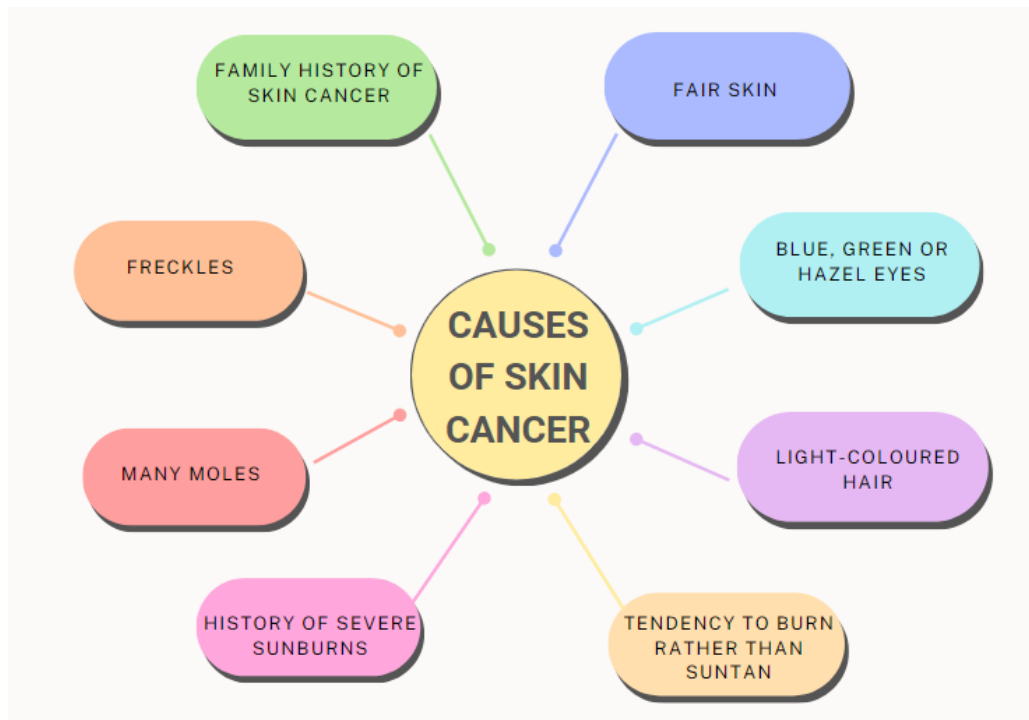


Fig.3: Causes of skin cancer as per the WHO report

Medical diagnostic technology has made amazing progress over the years, yet the difficulty of melanoma diagnosis continues to be a major obstacle. Dermoscopy has been a useful tool for getting an in-depth look at skin lesions(tumours), but its effectiveness has largely hinged on the dermatologist's skill [5-7]. This dependence on human skill raises concerns about variability in diagnosis and the potential for misdiagnosis.

On the other hand, artificial intelligence, particularly deep learning CNNs, has shown great work in providing consistent diagnostic results [1,3,5,7]. By analysing vast datasets of 10,000 dermoscopic images, these AI systems can learn and identify patterns and tiny clues that may be unseen by the human eye. Moreover, they do not suffer from fatigue or inter-operator variability, making them an attractive addition to the dermatologist's toolkit.

The possibility for human expertise and AI to work together is an important factor that deserves attention. AI might be considered as a partner in the diagnostic procedure rather than as a rival. Dermatologists, who bring years of experience and medical knowledge, can work hand-in-hand with AI systems to achieve even greater diagnostic accuracy. This collaboration, which reflects the general trend of man-machine cooperation in healthcare, can go above skin cancer detection to many

medical specialities.

In this research, we want to use skin cancer detection to integrate the best aspects of artificial and human intelligence. By combining the capabilities of image processing with dermatologists' expertise, we want to improve diagnostic accuracy while also paving the way for more practical and generally accessible healthcare solutions. This introduction builds the foundation for thorough study of how this cooperative approach might be applied to eliminate the pressing issue of skin cancer detection.

## MOTIVATION

Understanding the significance of early recognition as well as cure of such condition, which has become into a significant public health concern, became the inspiration for the skin cancer identification. The sources mentioned in the introduction emphasise a number of critical driving forces that served as inspiration for this activity.

During past few decades, there has been ascend in the incidence of skin cancer, especially melanoma. It is cause of a significant amount of cancer-related deaths. Early skin cancer detection considerably increases the likelihood of a successful recovery. Consequently, as this disease spreads, detection techniques get better. Although dermoscopy can be examined with the unaided eye, it can still have difficulties with accurate diagnosis. Even highly skilled dermatologists who are conversant with different dermoscopic algorithms can identify melanoma with a sensitivity of less than 80%. This emphasises necessity for more precise and repeatable diagnostic techniques.

Automatic computer analysis of images has become known as a viable way to help doctors increase the diagnostic precision of skin cancer screening. These systems are able to deliver more objective and reliable evaluations of dermoscopic pictures by employing machine learning techniques. Although past research has mostly partnered human dermatologists versus machine learning algorithms, there is opportunity to investigate a cooperative method. The use of AI technologies with the expertise of human dermatologists has the potential to enhance diagnostic precision and yield superior results.

Effective treatment for skin cancer requires early detection. The references stress that skin cancer in its premature stages has a substantially higher survival probability than in its later stage. Consequently, every advancement in early detection techniques holds potential for life-saving uses. All around the world, individuals are affected by skin cancer. It is not limited to certain regions; it impacts individuals everywhere. The creation of a trustworthy and practical skin cancer testing instrument could have a big impact on global health. The cost of healthcare is an issue in many nations. AI-powered diagnostic automation has the potential to save healthcare costs through better resource allocation and diagnostic efficacy.

In conclusion, we can state that the driving force for this project is the requirement to address the rising incidence of skin cancer, enhance diagnostic precision, save lives by means of early detection, and provide more widely available and economically viable skin cancer treatment options. The combination of human abilities and AI capabilities will improve diagnosis procedures and assist patients, carers, and healthcare systems.



## OBJECTIVES

We want to increase diagnostic accuracy and open the door to more affordable and effective healthcare solutions by leveraging the power of image processing and the knowledge of dermatologists. A handful of the project objectives I hope to achieve are as follows:

- Study previous work on the topic.
- Collect the data to perform various data mining algorithms.
- Pre-process the dataset to get the balanced dataset.
- Apply various data mining algorithms to identify meaningful insights from data.
- Develop an image processing model to identify skin cancer.
- Create an easy-to-use application to identify skin cancer early, minimising the likelihood of late-stage diagnoses.

These goals outline the project's plans to enhance skin cancer detection, benefiting both patients and healthcare providers.

## LITERATURE REVIEW

In research work [1], authors studied HAM10000 dataset. With implementation of CNN algorithms, authors achieved 90% accuracy in skin cancer classification.

In research paper [2], authors proposed Enhanced Image Analysis Technique (EIAT). Authors gathered data and processed on it. Authors performed deep learning algorithms and achieved 94.58% accuracy in skin cancer detection.

In research paper [3], authors created dataset named test-set-300 and implemented deep learning CNN to achieve highly accurate image classification model. The study's findings show that a deep learning CNN with the necessary training can provide a very accurate diagnosis.

In research work [4], authors used ISIC dataset and implemented ANN, SVM, CNN algorithms. CNN provided best results. This methodology detects the lesion in very quick time helping the technicians to perfect their diagnostic skills.

In research work [5], authors used HAM10000 and ISIC dataset and implemented CNN algorithm and fusion method where fusion method provided best results.

In research paper [6], series of image processing operations, involving segmentation, pre-processing, classification, feature extraction, are included. The dermoscopy's ABCD rule is used to describe skin lesions. Skin lesion is classified by comparing the feature parameter with the previous thresholds.

In research work [7], authors implemented various algorithms such as CONV2D, RESNET50, SQUEEZNET, DENSENET201, INCEPTIONV3 on HAM10000. Out of which SQUEEZNET is more accurate.

In research work [8], the authors have implemented algorithms like Probabilistic Neural Network (PNN), Random Forest (RF), Support Vector Machine (SVM), and a combination of SVM and RF for classification. The SVM+RF classifier outperforms other classifiers and provides the best sensitivity, specificity, and accuracy, making it a desirable choice to recognise skin cancer.

In research paper [9], The study focuses on open-source data mining technologies used in healthcare applications, such as Rapid Miner, Orange, Weka, KNIME, and Sisense. Several data mining methods, such as Apriori, decision trees,

neural networks, text mining, and Naïve Bayes, is studied. Deep Neural Network (with 98% accuracy rate) is shown to be the most accurate method.

In research paper [10], authors have explored and obtained health-related data from the database using a three-tier architecture which includes Java applets for database exploration and retrieval. It also presents a data mining method that can be used for research and diagnostics in order to discover association rules between skin cancer factors.

In research article [11], authors use clustering techniques to divide patients into categories that are relevant and non-related, data pre-processing to clean and prepare the data for analysis, and a new MAFIA algorithm to find frequent trends. authors have collected data of 200 persons of different age.

In research article [12], authors use decision tree (J48, ID3) and Naïve Bayes algorithm on skin, breast and lung cancer datasets. ID3 is most accurate with 100% accuracy on all the datasets.

In research paper [13], authors focus on ANNs, CNNs, KNNs, and RBFNs for classification of lesion images. CNN gives more accurate results. International Skin Imaging Collaboration (ISIC) dataset has been used.

In research paper [14], authors have performed various methods on DermIs and DermQuest datasets out of which SVM and Adaboost produces best results.

In research paper [15], authors performed SVM, quadratic discriminant and random forest on ISIC-ISBI 2016 and random forest is most accurate.

**Review Table:**

<b>Year</b>	<b>Authors</b>	<b>Algorithms used</b>	<b>Dataset used</b>	<b>Conclusion</b>
2023	Pooja Nadiger, Ranjana Pavaskar et and all.	Convolutional Neural Network	HAM10000	Achieved 90% accuracy in skin cancer classification.

2022	Arivazhagan N, Mukunthan MA, et and all.	Deep learning	Gathered from various sources.	Achieved 94.58% accuracy in skin cancer detection.
2021	A. Murugan, S. Anu H Nair, et and all.	SVM, RF, PNN, SVM+RF	Not specified	SVM+ RF outperforms other classifiers.
2021	Dildar M, Akram S, et and all.	ANN, CNN, KNN, and RBFN	International Skin Imaging Collaboration (ISIC) dataset	CNN is most accurate.
2021	A. Javaid, M. Sadiq and F. Akram	SVM, quadratic discriminant and random forest.	ISIC-ISBI 2016	Random forest gives best results.
2020	Mohammad Ali Kadampur,et and all.	CONV2D, RESNET50, SQUEEZNET, DENSENET201, INCEPTIONV3.	HAM10000	SQUEEZNET is more accurate.
2020	H. Beenish and M. Fahad	Apriori, decision trees, Naïve Bayes, neural network	Open- source data	Deep Neural Network (DNN) achieves 98% accuracy.

2019	M. Vijayalakshmi	ANN, CNN and SVM.	ISIC	CNN performed more accurately.
2019	Hekler A, Utikal JS et and all.	CNN and fusion method	HAM10000 and ISIC	Fusion method is more accurate.
2018	H.A. Haenssle, C. Fink et and all.	CNN	Test-set-300	CNN model is used to achieve highly accurate image classification.
2017	E. Jana, R. Subban et and all.	ANN, SVM, Adaboost, CNN, BPN.	DermIs and DermQuest	SVM and Adaboost produces best results.
2015	Shivangi Jain, Vandana Jagtap, et and all.	Segmentation, feature extraction, image processing, classification	Not specified	skin lesion can be efficiently segmented using proposed segmentation method.
2013	Ahmed, Kawsar, et and all.	Clustering techniques, MAFA algorithm	Collected data from different sources.	Offers affordable and accessible tool for skin cancer risk analysis.

2013	Priyanga A, Parakram S.	Decision tree (J48, ID3) and Nave Bayes	Not specified	ID3 is most accurate with 100% accuracy
2001	S. M. Chung and Qing Wang	Image segmentation, feature extraction, data mining	Own database	Provides platform for accessing skin cancer related data.

## Dataset Preparation

The initiation of the project involved data collection and pre-processing. In order to achieve best prediction model, pre-processing the dataset is must. The following steps are performed to achieve well quality data:

**1. Accessing the data:** The dataset used for this project is accessed from Kaggle(<https://www.kaggle.com/kmader/skin-cancer-mnist-ham10000>)[16]. This dataset has 2 important .csv files that are needed for model fitting and image processing.

- HAM10000\_metatdata.csv which contains following attributes:

Attributes	Data type	Values	Description
lesion_id	varchar	HAM0000000 - HAM0007628	It is unique for each skin lesion. It is used to distinguish and track individual lesions in the dataset.
image_id	varchar	ISIC0024306 - ISIC0034320	It is specific to each image of a skin lesion. It helps associate an image with a particular lesion.
dx	character	<div> <div> Bcc Akiec Mel </div> <div> } </div> <div> Canc- erous </div> </div> <div> <div> Bkl Df Vasc Nv </div> <div> } </div> <div> Non- Canc- erous </div> </div>	It represents the diagnosis or class of the skin lesion. It provides information about the type of skin condition or disease, such as melanoma, nevus, or other dermatological conditions.
dx_type	character	Histopathology Follow-up Expert consensus Confocal	This attribute describes the method or type of diagnosis for the skin lesion. It can indicate whether the diagnosis was made through expert consensus, follow-up, histopathology, or other means.
Age	float	0,5,10,15,20,25, 30,35,40,45,50, 55,60,65,70,75, 80,85	This attribute records the patients' age who has skin lesion. Age information can be relevant for analysing how skin conditions vary with age.
Sex	character	Male Female	This attribute indicates the gender of the patient with the skin lesion. It can be used to study the gender-based prevalence of skin conditions.
localization	character	Abdomen, Acral, Back, Chest, Ear, Face Foot, Genital, Hand, Lower extremity, Neck Scalp, Trunk, Unknown, Upper extremity	This attribute designates the anatomical site on the body where the skin lesion is located. It provides information about where on the body the lesion occurs, which can be important for diagnosis and research.

- hmnist\_28\_28\_RGB.csv which contains images in RGB format.

## 2. Library Installation:

For data pre-processing different operations on data have to be performed. Python provides some inbuilt libraries. Install the libraries such as pandas, NumPy, seaborn and matplotlib etc.

## 3. Data pre-processing:

In this step, perform different operations on the dataset that can be accessed. The data pre-processing steps are as follows:

### I. Data Exploring:

At this step, we check if there any null value is present in dataset. If yes then there are 2 Approaches for handling the null value- we just ignore or delete that value or we fill that value. Either we fill that null value by using mean or ignore the data value by just dropping it.

```
df.isnull().sum()
: lesion_id      0
  image_id      0
  dx            0
  dx_type       0
  age          57
  sex          0
  localization  0
  dtype: int64

df.dropna(subset=['age'], inplace=True)
df.isnull().sum()
_____
  lesion_id      0
  image_id      0
  dx            0
  dx_type       0
  age           0
  sex           0
  localization  0
  dtype: int64
```

### II. Label Mapping:

In provided dataset, HAM10000\_metadata.csv file contain attributes which are not sufficient to tell whether the lesion is cancerous or not. With the use of label mapping, another column cancer have been created which tells whether the lesion is cancerous or not.

```
label_mapping = {'nv': 0, 'mel': 1, 'bkl': 0, 'bcc': 1, 'akiec': 1, 'vas': 0, 'df': 0}
df['cancer'] = df['dx'].map(label_mapping)
df.head()
```



	lesion_id	image_id	dx	dx_type	age	sex	localization	cancer
0	HAM_0000118	ISIC_0027419	bkl	histo	80.0	male	scalp	0.0
1	HAM_0000118	ISIC_0025030	bkl	histo	80.0	male	scalp	0.0
2	HAM_0002730	ISIC_0026769	bkl	histo	80.0	male	scalp	0.0
3	HAM_0002730	ISIC_0025661	bkl	histo	80.0	male	scalp	0.0
4	HAM_0001466	ISIC_0031633	bkl	histo	75.0	male	ear	0.0

### III. Label Encoding:

The provided data contain some attributes with text datatype. We need to convert the necessary attributes in encoded format. Since the data has so much columns which are unnecessary, we must drop those columns.

Before label encoding and column dropping:

	lesion_id	image_id	dx	dx_type	age	sex	localization	cancer
0	HAM_0000118	ISIC_0027419	2	histo	80.0	male	scalp	0.0
1	HAM_0000118	ISIC_0025030	2	histo	80.0	male	scalp	0.0
2	HAM_0002730	ISIC_0026769	2	histo	80.0	male	scalp	0.0
3	HAM_0002730	ISIC_0025661	2	histo	80.0	male	scalp	0.0
4	HAM_0001466	ISIC_0031633	2	histo	75.0	male	ear	0.0

After label encoding and column dropping:

	dx	age	sex	localization	cancer
0	2	16	1	11	0
1	2	16	1	11	0
2	2	16	1	11	0
3	2	16	1	11	0
4	2	15	1	4	0

### IV. Splitting the dataset:

To train the dataset, it must be split. Splitting the dataset makes it easy to understand whether the predictions are right or not. The data is split into X and Y. X contains the attributes which are independent and Y contains the attribute which is dependent. In this case dx, age, sex, localization is X and cancer is Y.

These X and Y are further split into test and train. Train is used to understand the insights of the data while test is used to make predictions based on the train data.

	dx	age	sex	localization
8225	5	14	1	2
3015	5	10	1	12
6135	5	10	0	12
1268	4	9	1	9
6722	5	8	1	14
...	...	...	...	...
1303	4	10	1	2
4035	5	16	1	12
7271	5	11	0	3
5212	5	10	0	0
3787	5	13	0	0

7966 rows × 4 columns

1	y_train
8225	0
3015	0
6135	0
1268	1
6722	0
...	..
1303	1
4035	0
7271	0
5212	0
3787	0

Name: cancer, Length: 7966,

In this report, I have done all pre-processing steps on dataset. Next I will start to perform different algorithms on dataset such as KNN, SVM, Random forest, decision tree and logistic regression.

### Exploratory Data Analysis (EDA)

Attributes	Values	Male patients percentage	Female patients percentage	Total	Percentage
Lesion(dx)	Cancerous	62.79	37.21	1954	19.49
	Non- Cancerous	51.84	47.45	8061	80.42
Diagnosis (dx_type)	Histopathology	56.57	43.23	5340	53.32
	Follow-up	51.88	48.11	3704	36.98
	Expert consensus	48.66	46.11	902	9.00
	Confocal	34.78	65.21	69	0.6
Age	0-10	51.80	46.98	166	1.6
	11-20	41.05	58.94	246	2.45
	21-30	42.61	57.38	711	7.09
	31-40	45.56	54.25	1738	17.35
	41-50	48.51	51.40	2486	24.82
	51-60	57.22	42.60	1812	18.09
	61-70	64.51	35.84	1487	14.84
	71-80	70.93	29.06	1022	10.20
	81-above	58.21	41.79	347	3.46
Sex	Male			5406	53.97
	Female			4552	45.45
Localization	Abdomen	57.43	42.36	1022	10.20
	Back	61.45	38.45	2192	21.88
	Chest	65.11	34.88	407	4.06
	Ear	46.42	53.57	56	0.55
	Face	53.28	46.71	745	7.43
	Foot	44.82	54.23	319	3.18
	Genital	29.16	70.83	48	0.47
	Hand	36.66	63.33	90	0.89
	Lower extremity	44.53	55.46	2077	20.73
	Neck	56.54	43.45	168	1.67
	Scalp	75.78	24.21	128	1.2
	Trunk	54.34	45.44	1404	14.01
	Unknown	39.31	40.59	234	2.33
	Upper extremity	55.63	44.36	1118	11.16

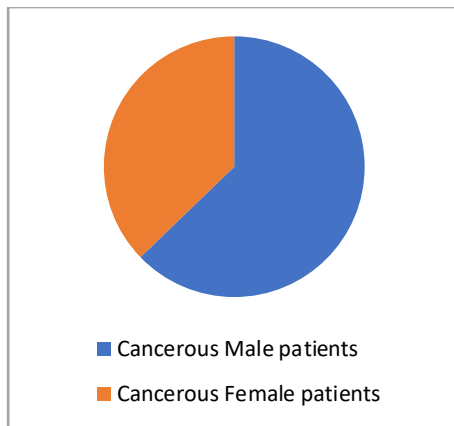


Fig.4: Cancerous male and female patients

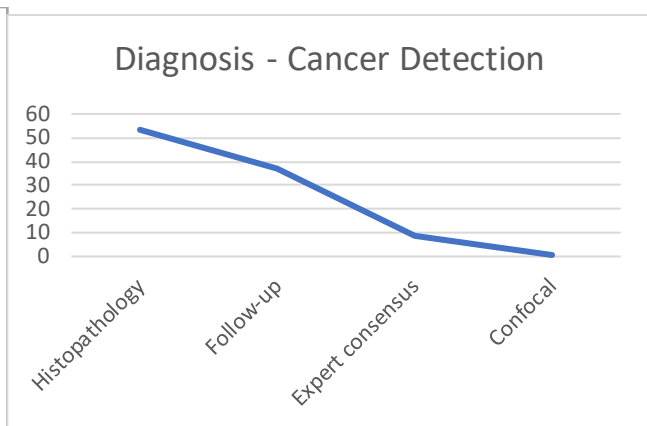


Fig.5: Diagnosis – Cancer detection

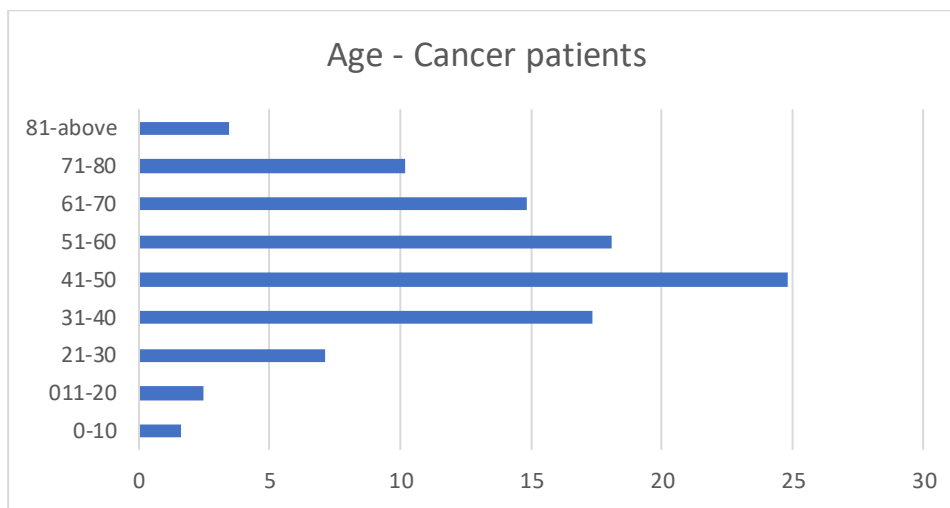


Fig.6: age – cancer patients

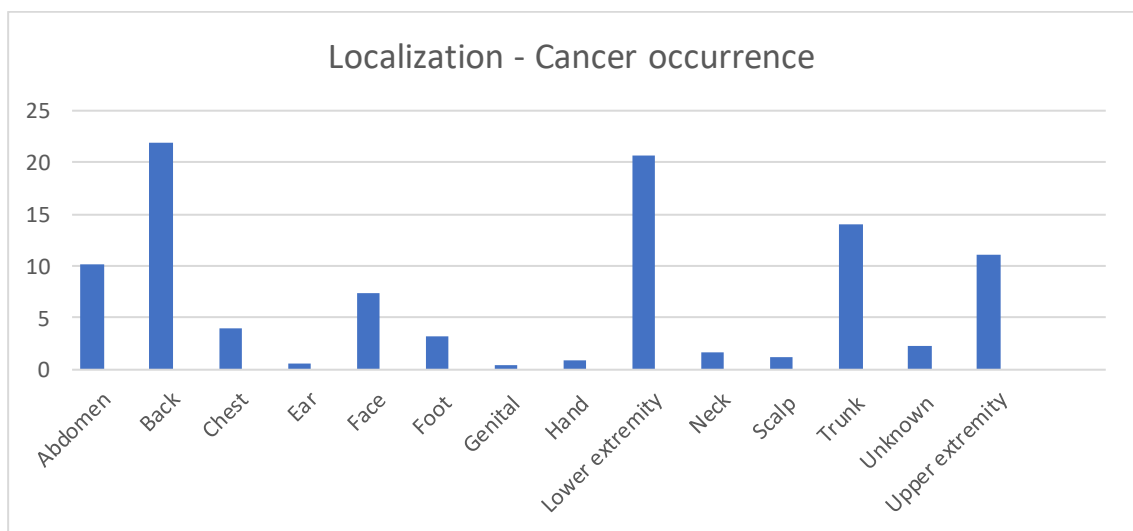


Fig.7: Localization – Cancer occurrence

The above data analysis chart shows that out of 10,015 lesions 19.41% are cancerous while 82.42% are non-cancerous. Most of the lesions are diagnosed with the use of histopathology (53.32%) and then follow up (36.98%) as shown in Fig.4. Data analysis chart displays the distribution of lesion with respect to different age groups. The age group of 41-50 has maximum lesion percentage (24.82%) followed by 51-60 (18.09%), 31-40 (17.35%) and 61-70 (14.84%). Meanwhile age group of 0-10 has lowest lesion percentage which is 1.6%.

The data analysis also draws attention to localization and lesion appearance percentage. Back with 21.88% tops the table followed by lower extremity (20.73%), upper extremity (11.16%) and abdomen (10.20%).

Fig.5 represents the percentage of male patients and female patients who are diagnosed with cancer. Fig.6 which shows the bar chart of age and cancer patient clearly shows that the age group of 41-50 has high lesion percentage. Fig.7 describes the location where the lesion is occurred. Back has maximum lesion occurrence percentage followed by lower extremity and upper extremity.

## Implementation

After pre-processing the dataset, data is ready to be implemented. Now, various algorithms can be performed on the dataset. The algorithms performed are as follows:

### 1. KNN (K-Nearest Neighbours):

KNN is used for foreseeing cancerous lesions by comparing them with lesion type of its k closest neighbours. The lesion is categorized by the algorithm according to the vast majority of group of its surroundings.

```
from sklearn.neighbors import KNeighborsClassifier
```

```
knn = KNeighborsClassifier(n_neighbors=3)
```

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

```
#prediction
```

```
pred=knn.predict(x_test)
```

```
#accuracy
```

```
from sklearn.metrics import accuracy_score
```

```
ac_kn=accuracy_score(y_test, pred)
```

```
print("Accuracy=",ac_kn)
```

```
#confusionmatrix
```

```
from sklearn.metrics import confusion_matrix
```

```
c_m = confusion_matrix(y_test, pred)
```

```
print(c_m)
```

```
#classificationreport
```

```
from sklearn.metrics import classification_report
```

```
print(classification_report(y_test, pred))
```

```
Accuracy= 0.9894578313253012
```

```
[[1567  4  0]
 [  6 389  0]
 [ 11  0 15]]
```

	precision	recall	f1-score	support
0	0.99	1.00	0.99	1571
1	0.99	0.98	0.99	395
2	1.00	0.58	0.73	26
accuracy			0.99	1992
macro avg	0.99	0.85	0.90	1992
weighted avg	0.99	0.99	0.99	1992

### 2. Naïve Bayes:

One supervised machine learning algorithm used for classification tasks,

such as text classification, is the Naïve Bayes classifier.

```
from sklearn.naive_bayes import GaussianNB
gnb = GaussianNB()
pred = gnb.fit(x_train, y_train).predict(x_test)
#accuracy
from sklearn.metrics import accuracy_score
ac_nb=accuracy_score(y_test, pred)
print("Accuracy=",ac_nb)
#confusionmatrix
from sklearn.metrics import confusion_matrix
c_m = confusion_matrix(y_test,pred)
print(c_m)
#classificationreport
from sklearn.metrics import classification_report
print(classification_report(y_test,pred))
```

Accuracy= 0.786144578313253

[[1366	205	0]			
[ 221	174	0]			
[ 0	0	26]]			
		precision	recall	f1-score	support
	0	0.86	0.87	0.87	1571
	1	0.46	0.44	0.45	395
	2	1.00	1.00	1.00	26
	accuracy			0.79	1992
	macro avg	0.77	0.77	0.77	1992
	weighted avg	0.78	0.79	0.78	1992

### 3. Random Forest:

The Random Forest aggregation approach is designed to improve prediction accuracy by combining many decision trees. A Random Forest reduces overfitting and increases resilience in predicting the occurrence of cancerous and non-cancerous lesion by combining the predictions of different trees. It manages complex interactions in the data with effectiveness.

```
from sklearn.ensemble import RandomForestClassifier
rf = RandomForestClassifier(max_depth=2, random_state=0)
rf.fit(x_train, y_train)
#accuracy
from sklearn.metrics import accuracy_score
ac_rf=accuracy_score(y_test, pred)
print("Accuracy=",ac_rf)
#confusionmatrix
from sklearn.metrics import confusion_matrix
c_m = confusion_matrix(y_test,pred)
print(c_m)
# classificationreport
from sklearn.metrics import classification_report
print(classification_report(y_test,pred))
```

```

Accuracy= 0.963855421686747
[[1571  0   0]
 [  46 349  0]
 [  26  0   0]]

```

	precision	recall	f1-score	support
0	0.96	1.00	0.98	1571
1	1.00	0.88	0.94	395
2	0.00	0.00	0.00	26
accuracy			0.96	1992
macro avg	0.65	0.63	0.64	1992
weighted avg	0.95	0.96	0.96	1992

#### 4. Decision Tree:

Decision trees are implemented to build a structure similar to trees in cancer lesion forecasting with every branch representing a choice based on a particular information. Recursively dividing the dataset into subgroups allows for the construction of the tree and the discovery of patterns linked to the incidence of strokes. Decision trees can represent non-linear connections and are accessible.

```

from sklearn.tree import DecisionTreeClassifier
ds = DecisionTreeClassifier(criterion='entropy',max_depth=7)
ds.fit(x_train,y_train)
#accuracy
from sklearn.metrics import accuracy_score
ac_dt=accuracy_score(y_test,pred)
print("Accuracy=",ac_dt)
#confusionmatrix
from sklearn.metrics import confusion_matrix
c_m = confusion_matrix(y_test,pred)
print(c_m)
#classificationreport
from sklearn.metrics import classification_report
print(classification_report(y_test,pred))

```



---

```

Accuracy= 1.0
[[1571  0  0]
 [  0 395  0]
 [  0  0 26]]
      precision    recall  f1-score   support

      0       1.00      1.00      1.00     1571
      1       1.00      1.00      1.00      395
      2       1.00      1.00      1.00       26

 accuracy          1.00          1.00          1.00     1992
 macro avg          1.00          1.00          1.00     1992
 weighted avg          1.00          1.00          1.00     1992

```

## 5. SVM, or support vector machine:

Support Vector Machines (SVM) are used for foreseeing cancerous and non-cancerous lesion by identifying the type of the lesion. The method seeks to minimize classification mistakes while maximizing the gap between groups. With the introduction of kernel-level operations, SVM may identify non-linear correlations and is especially useful for handling large amounts of data.

```

from sklearn.preprocessing import StandardScaler
from sklearn.svm import SVC
from sklearn.metrics import accuracy_score, classification_report

scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(x_train)
X_test_scaled = scaler.transform(x_test)

# Create an SVM model
svm_model = SVC(kernel='linear', C=1.0, random_state=42)

# Train the SVM model
svm_model.fit(X_train_scaled, y_train)

# Make predictions on the test set
y_pred = svm_model.predict(X_test_scaled)

# Evaluate the performance of the model
ac_sv = accuracy_score(y_test, y_pred)

ac_sv

```

---

```
0.7781124497991968
```

## 6. Logistic Regression:

Logistic regression is used for classification based on binary values. It may be used to

determine the probability that a person is at risk of having a cancer. The technique provides probabilities that may be adjusted for identification by modelling the link between input characteristics and the chance of a cancer. Because of its clarity and cleanliness, this linear model is frequently utilized.

```
from sklearn.linear_model import LogisticRegression
# Standardize the features by scaling
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(x_train)
X_test_scaled = scaler.transform(x_test)
# Create a logistic regression model
logreg_model = LogisticRegression(random_state=42)
# Train the logistic regression model
logreg_model.fit(X_train_scaled, y_train)
# Make predictions on the test set
y_pred = logreg_model.predict(X_test_scaled)
# Evaluate the performance of the model
ac_lr = accuracy_score(y_test, y_pred)
ac_lr
0.8237951807228916
```

After comparing accuracies of all the algorithms implemented, the following graph has been obtained:

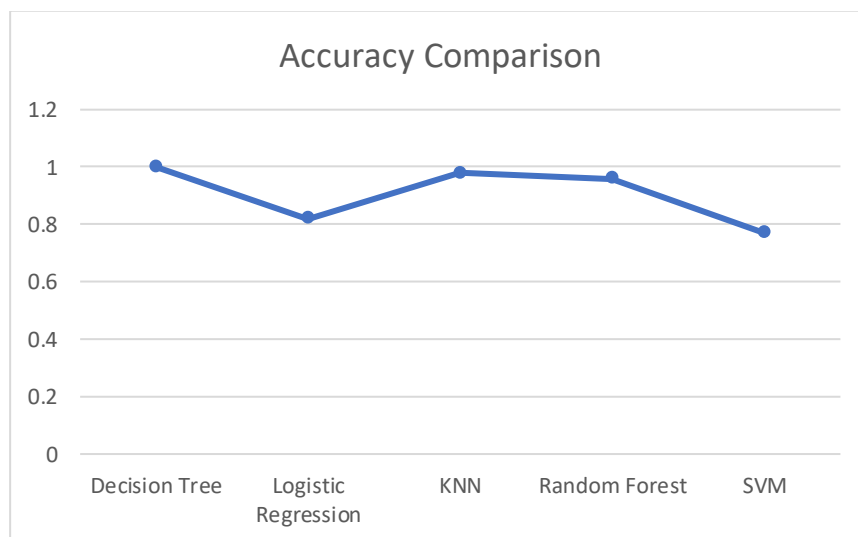


Fig.8: Algorithm accuracy comparison

## Image Processing

Above algorithms used to tell whether the lesion is cancerous or not. Now we will perform image processing to actually identify lesion type. The dataset already contains the .csv file which has all the images into RGB format. So, we will load the dataset and will split for training and testing. Afterwards EDA is performed to handle the imbalance in classes. Imbalanced plot:

```
import seaborn as sns
sns.countplot(train_set['label'])
```

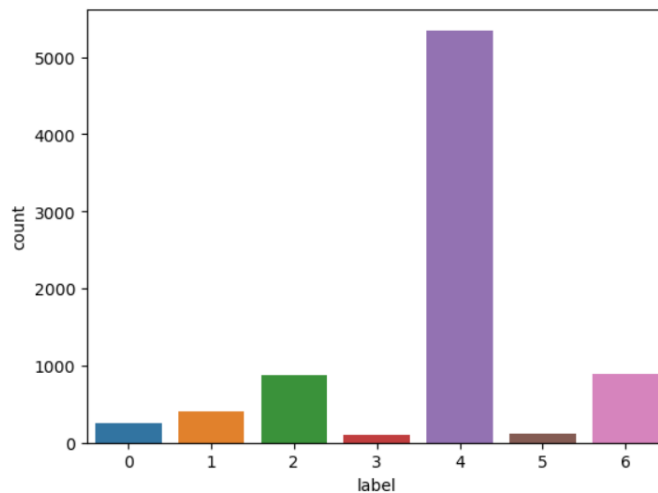


Fig.9: Imbalanced plot

Fig.9 shows imbalanced plot of the dataset where 4<sup>th</sup> numbered class which is melanoma has most number of images.

Balanced plot:

```
from imblearn.over_sampling import RandomOverSampler
oversample = RandomOverSampler()
x_train,y_train = oversample.fit_resample(x_train,y_train)
sns.countplot(y_train)
```

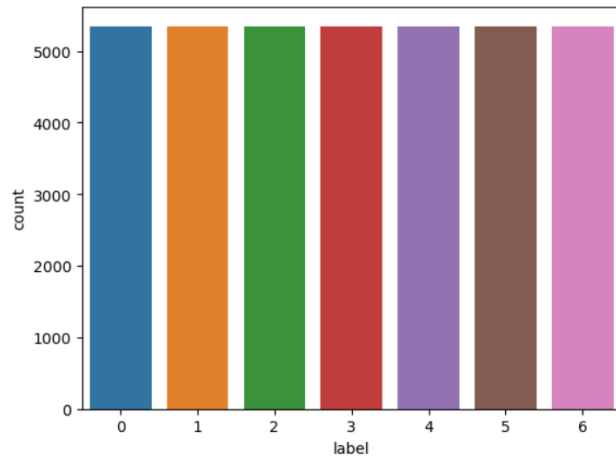


Fig. 10: Balanced plot

Fig.10 shows the balanced plot where each class has same number of data mages.

### Model Architecture

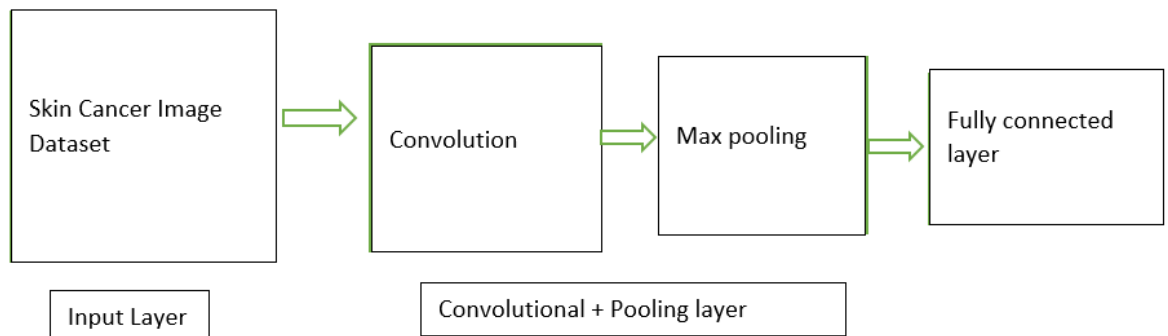


Fig. 11: CNN model

The above architecture is a CNN model used for the skin cancer image classification. It utilizes multiple layers to learn complex patterns and features from the images, which can then be used to classify the images into the various skin cancer types.

The architecture is divided into four main parts:

1. **Input Layer:** This is the first layer of the CNN model. It takes the input images and processes them through the network.
2. **Convolution Layer:** It consists of multiple filters or kernels that slide over the input image. Each filter extracts a specific feature from the image and performs convolution operations on the input data. This helps the model to learn various features and patterns from the images.
3. **Max Pooling Layer:** After the convolution layer, the Max Pooling layer is used to reduce the spatial dimensions of the output. It selects the maximum value from the neighbouring region of the output feature map and down-samples the data, effectively reducing the spatial size.
4. **Convolution + Pooling Layer:** This layer combines the Convolution and Max Pooling layers. Multiple filters are applied to the input image and then the output is passed through the Max Pooling layer to reduce the spatial dimensions.

The CNN model is further extended with fully connected layers, which connect the convolutional layers and form the final output of the model. These layers help the model to combine the features learned from the convolutional layers and classify the input images into the appropriate skin cancer types.

## **Model Building**

Python provides some tools for building neural network models. These tools are as follows:

### **1. Tensorflow:**

- Imagine you have a bunch of data, like numbers in a table. TensorFlow is like a powerful toolbox for doing math operations on these numbers, especially when you have lots of them.
- Think of it as a toolkit that helps you do complex calculations efficiently. It's widely used for building and training machine learning models.

### **2. Keras:**

- Keras is like a friendly interface or a convenient wrapper around TensorFlow. It makes it easier for humans (you!) to build and train neural networks without diving into too many technical details.
- It simplifies the process of defining, configuring, and training deep learning models.

### **3. Sequential:**

- A Sequential model in Keras is a straightforward way to organize the different parts of a neural network, one layer at a time.
- Imagine you're making a sandwich, and each layer (like bread, cheese, and lettuce) in the Sequential model is added one after the other. Similarly, in a Sequential model, layers are stacked in a sequence, making it easy to understand the flow of information through the neural network.

```
from tensorflow.keras.models import Sequential
```

```

from tensorflow.keras.layers import Conv2D, Flatten, Dense, MaxPool2D
import tensorflow as tf
# Time measurement
%time
# Neural Network Model
model = Sequential()
# Convolutional Layers
model.add(Conv2D(16, kernel_size=(3, 3), input_shape=(28, 28, 3), activation='relu',
padding='same'))
model.add(MaxPool2D(pool_size=(2, 2)))
model.add(tf.keras.layers.BatchNormalization())
model.add(Conv2D(32, kernel_size=(3, 3), activation='relu'))
model.add(Conv2D(64, kernel_size=(3, 3), activation='relu'))
model.add(MaxPool2D(pool_size=(2, 2)))
model.add(tf.keras.layers.BatchNormalization())
model.add(Conv2D(128, kernel_size=(3, 3), activation='relu'))
model.add(Conv2D(256, kernel_size=(3, 3), activation='relu'))
# Flatten Layer
model.add(Flatten())
model.add(tf.keras.layers.Dropout(0.2))
# Dense Layers
model.add(Dense(256, activation='relu'))
model.add(tf.keras.layers.BatchNormalization())
model.add(tf.keras.layers.Dropout(0.2))
model.add(Dense(128, activation='relu'))
model.add(tf.keras.layers.BatchNormalization())
model.add(Dense(64, activation='relu'))
model.add(tf.keras.layers.BatchNormalization())
model.add(tf.keras.layers.Dropout(0.2))
model.add(Dense(32, activation='relu'))
model.add(tf.keras.layers.BatchNormalization())
# Output Layer
model.add(Dense(7, activation='softmax'))
# Display Model Summary
model.summary()

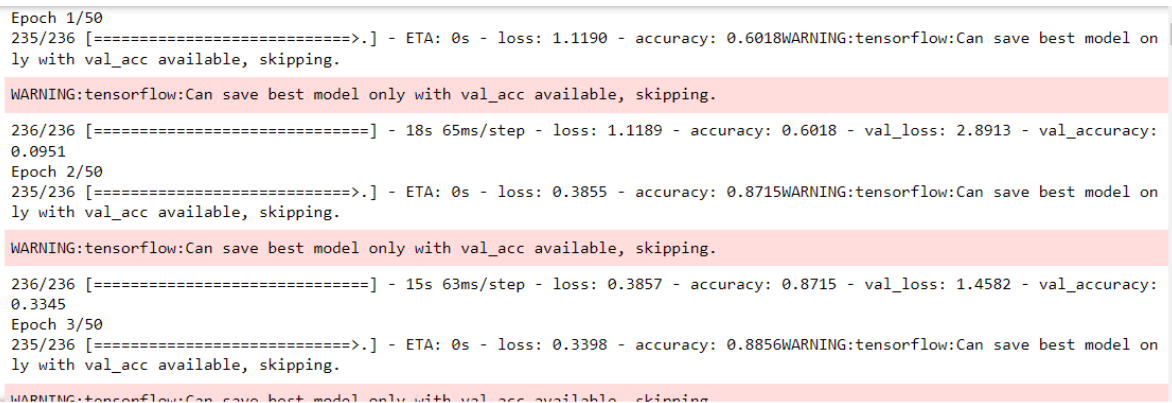
```

Wall time: 0 ns  
Model: "sequential"

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 28, 28, 16)	448
max_pooling2d (MaxPooling2D)	(None, 14, 14, 16)	0
batch_normalization (Batch Normalization)	(None, 14, 14, 16)	64
conv2d_1 (Conv2D)	(None, 12, 12, 32)	4640
conv2d_2 (Conv2D)	(None, 10, 10, 64)	18496
max_pooling2d_1 (MaxPooling2D)	(None, 5, 5, 64)	0
batch_normalization_1 (Batch Normalization)	(None, 5, 5, 64)	256
conv2d_3 (Conv2D)	(None, 3, 3, 128)	73856
conv2d_4 (Conv2D)	(None, 1, 1, 256)	295168
flatten (Flatten)	(None, 256)	0
dropout (Dropout)	(None, 256)	0
dense (Dense)	(None, 256)	65792
batch_normalization_2 (Batch Normalization)	(None, 256)	1024
dropout_1 (Dropout)	(None, 256)	0
dense_1 (Dense)	(None, 128)	32896
batch_normalization_3 (Batch Normalization)	(None, 128)	512
dense_2 (Dense)	(None, 64)	8256
batch_normalization_4 (Batch Normalization)	(None, 64)	256
dropout_2 (Dropout)	(None, 64)	0
dense_3 (Dense)	(None, 32)	2080
batch_normalization_5 (Batch Normalization)	(None, 32)	128
dense_4 (Dense)	(None, 7)	231
=====		
Total params: 504103 (1.92 MB)		
Trainable params: 502983 (1.92 MB)		
Non-trainable params: 1120 (4.38 KB)		

Fitting the model:

```
from datetime import datetime
start_time = datetime.now()
history = model.fit(x_train,
                    y_train,
                    validation_split=0.2,
                    batch_size = 128,
                    epochs = 50,
                    shuffle=True,
                    callbacks=[callback])
end_time = datetime.now()
print('Duration: {}'.format(end_time - start_time))
```



```
Epoch 1/50
235/236 [=====>.] - ETA: 0s - loss: 1.1190 - accuracy: 0.6018WARNING:tensorflow:Can save best model on
ly with val_acc available, skipping.

WARNING:tensorflow:Can save best model only with val_acc available, skipping.

236/236 [=====] - 18s 65ms/step - loss: 1.1189 - accuracy: 0.6018 - val_loss: 2.8913 - val_accuracy:
0.0951
Epoch 2/50
235/236 [=====>.] - ETA: 0s - loss: 0.3855 - accuracy: 0.8715WARNING:tensorflow:Can save best model on
ly with val_acc available, skipping.

WARNING:tensorflow:Can save best model only with val_acc available, skipping.

236/236 [=====] - 15s 63ms/step - loss: 0.3857 - accuracy: 0.8715 - val_loss: 1.4582 - val_accuracy:
0.3345
Epoch 3/50
235/236 [=====>.] - ETA: 0s - loss: 0.3398 - accuracy: 0.8856WARNING:tensorflow:Can save best model on
ly with val_acc available, skipping.

WARNING:tensorflow:Can save best model only with val_acc available, skipping.
```

## Model Evaluation:

```
#plot of accuracy vs epoch
plt.plot(history.history['accuracy'])
plt.plot(history.history['val_accuracy'])
plt.title('model accuracy')
plt.ylabel('accuracy')
plt.xlabel('epoch')
plt.legend(['train', 'val'], loc='upper left')
plt.show()
```



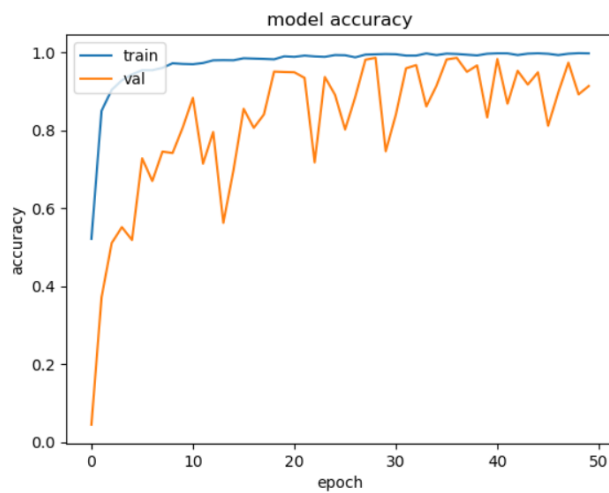


Fig. 12: Plot of accuracy vs epoch

Fig. 12 represents relationship between model's accuracy and epoch.

```
#plot of loss vs epoch
plt.plot(history.history['loss'])
plt.plot(history.history['val_loss'])
plt.title('model loss')
plt.ylabel('loss')
plt.xlabel('epoch')
plt.legend(['train', 'val'], loc='upper left')
plt.show()
```

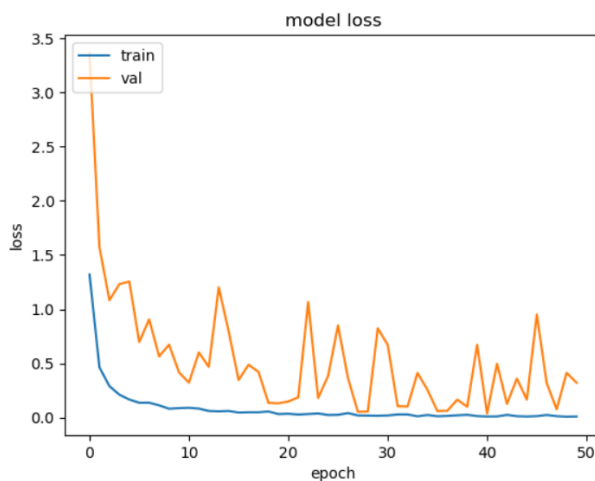


Fig. 13: Plot of loss vs epoch

## RESULT

```
1 #https://pillow.readthedocs.io/en/stable/
2
3 import PIL
4
5 image=PIL.Image.open('C:/Users/Admin/Desktop/skin cancer detection/data/HAM10000_images_part_1/ISIC_0024306.jpg')
6
7 image=image.resize((28,28))
8
9 img=x_test[1]
10
11 img=np.array(image).reshape(-1,28,28,3)
12
13 result=model.predict(img)
14
15 print(result[0])
16
17 result=result.tolist()
18
19 max_prob=max(result[0])
20
21 class_ind=result[0].index(max_prob)
22
23 print(classes[class_ind])
```

1/1 [=====] - 0s 170ms/step  
[3.5341545e-06 1.7356006e-06 1.3786728e-06 2.0972589e-06 9.9996305e-01  
1.6531283e-06 2.6553600e-05]  
( 'nv', ' melanocytic nevi')

Fig. 14: Output of model

The screenshot shows a web application titled "Skin Cancer Detection using Convolutional Neural Network". Below the title, there is a prompt "Select Skin image here". Underneath this prompt, there is a file selection interface consisting of a button labeled "Choose File" and a text input field containing the filename "tester.jpg". Below the file selection area, there is a green button labeled "Submit".

Fig. 15: Input page (home.html)

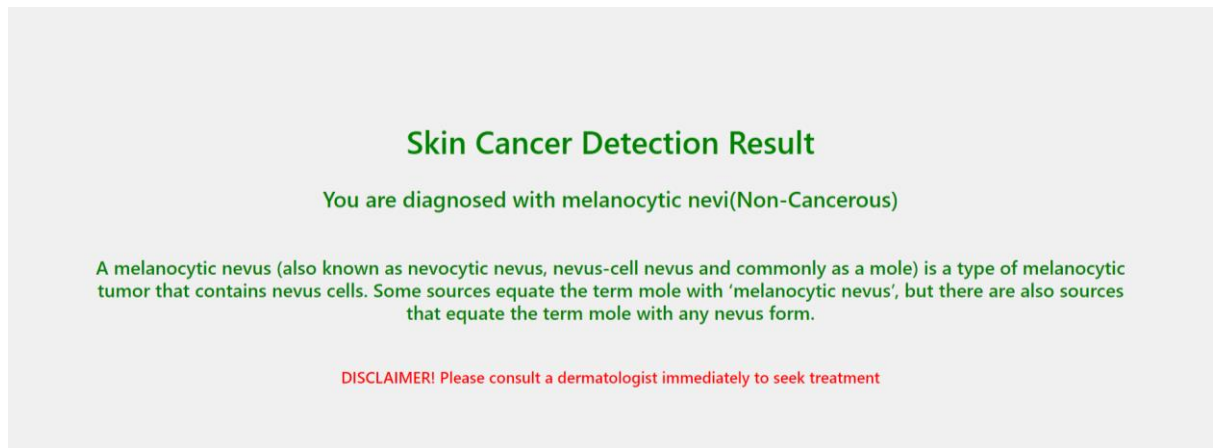


Fig. 16: Output page(result.html)

The data analysis shows that out of 10,015 lesions 19.41% are cancerous while 82.42% are non-cancerous. In which most of the lesions are diagnosed with the use of histopathology (53.32%). Data analysis percentage is different for different data groups The age group of 41-50 has maximum lesion percentage (24.82%) followed by 51-60 (18.09%), 31-40 (17.35%) and 61-70 (14.84%). Meanwhile, age group of 0-10 has lowest lesion percentage which is 1.6%.

After performing all the tasks, the outcomes got are as shown above. KNN provided the accuracy of 98%, Naïve Bayes provided the accuracy of 78%, Random forest provided accuracy of 96%, decision tree provided accuracy of 100%, SVM provided accuracy of 77% and logistic regression provided accuracy of 82%. With all the accuracies, decision tree proves to be the best fitting algorithm.

The CNN model used for image processing provides accuracy of 99% and classifies the images very accurately. The web application works properly and allows user interaction.

## CONCLUSION

By fulfilling the objectives that have been declared earlier this web application can increase diagnostic accuracy and open the door to more affordable and effective healthcare solutions by leveraging the power of image processing and the knowledge of dermatologists.

After studied previous work on the topic and understood what needs to be done. Dataset is collected to perform various data algorithms and got the accuracies of each of them. On the basis of that I can consider decision tree to be the most best fitting algorithm.

Then pre-processing is done then to get the balanced dataset, added the required columns and removed the unessential one. After that various DML algorithms applied to identify meaningful insights from the data.

Afterwards an image processing model is developed to identify skin cancer which gives 99% accuracy. Then created an easy-to-use application to identify skin cancer early, minimizing the likelihood of late-stage diagnoses.

The goals that outline the project's plans to enhance skin cancer detection, benefiting both patients and healthcare providers have been achieved and an easy to use web application with high accuracy have been developed.

## REFERENCES

1. Pooja Nadiger, Ranjana Pavasker, Srushti C, Surabhi Hangal, Dr. Vandana S. Bhat, "Skin Cancer Detection and Classification Using Deep Learning", IJRASET, 2023-05-30
2. Arivazhagan N, Mukunthan MA, Sundaranarayana D, Shankar A, Vinoth Kumar S, Kesavan R, Chandrasekaran S, Shyamala Devi M, Maithili K, Barakkath Nisha U, Abebe TG. "Analysis of Skin Cancer and Patient Healthcare Using Data Mining Techniques." Comput Intell Neurosci. 2022 Sep 26.
3. H.A. Haenssle, C. Fink, R. Schneiderbauer, F. Toberer, T. Buhl, A. Blum, "Man against machine: diagnostic performance of a deep learning convolutional neural network for dermoscopic melanoma recognition in comparison to 58 dermatologists", Volume 29, Issue 8, 2018,
4. M, Vijayalakshmi. (2019). "Melanoma Skin Cancer Detection using Image Processing and Machine Learning. International Journal of Trend in Scientific Research and Development", Volume-3. 780-784. 10.31142/ijtsrd23936.
5. Hekler A, Utikal JS, Enk AH, Hauschild A, Weichenthal M, Maron RC, Berking C, Haferkamp S, Klode J, Schadendorf D, Schilling B, Holland-Letz T, Izar B, von Kalle C, Fröhling S, Brinker TJ; "Collaborators. Superior skin cancer classification by the combination of human and artificial intelligence. Eur J Cancer". 2019 Oct; 120:114-121. doi: 10.1016/j.ejca.2019.07.019. Epub 2019 Sep 10. PMID: 31518967.
6. Shivangi Jain, Vandana jagtap, Nitin Pise, "Computer Aided Melanoma Skin Cancer Detection Using Image Processing, Procedia Computer Science", Volume 48, 2015.
7. Mohammad Ali Kadampur, Sulaiman Al Riyaa, "Skin cancer detection: Applying a deep learning -based model driven architecture in the cloud for classifying dermal cell images", Informatics in Medicine Unlocked, Volume 18, 2020.
8. A. Murugan, S. Anu H Nair, A. Angelin Peace Preethi, K. P. Sanal Kumar, "Diagnosis of skin cancer using machine learning techniques, Microprocessors and

Microsystems”, Volume 81, 2021.

9. H. Beenish and M. Fahad, "Skin Cancer Prediction using Data Mining and its Techniques – A Review," 2020 International Conference on Computing and Information Technology (ICCIT-1441), Tabuk, Saudi Arabia, 2020, pp. 1-4, doi: 10.1109/ICCIT-144147971.2020.9213800.
10. S. M. Chung and Qing Wang, "Content-based retrieval and data mining of a skin cancer image database," Proceedings International Conference on Information Technology: Coding and Computing, Las Vegas, NV, USA, 2001, pp. 611-615, doi: 10.1109/ITCC.2001.918864.
11. Ahmed, Kawsar & Jesmin, Tasnuba & Rahman, Md. (2013). “Early Prevention and Detection of Skin Cancer Risk using Data Mining”. International Journal of Computer Applications. 62. 1-6. 10.5120/10065-4662.
12. Priyanga, A. & Prakasam, S. (2013). “Effectiveness of Data Mining - based Cancer Prediction system (DMBCPS). International Journal of Computer Applications.” 83. 11-17. 10.5120/14483-2791.
13. Dildar M, Akram S, Irfan M, Khan HU, Ramzan M, Mahmood AR, Alsaiani SA, Saeed AHM, Alraddadi MO, Mahnashi MH. “Skin Cancer Detection: A Review Using Deep Learning Techniques. Int J Environ Res Public Health.” 2021 May 20;18(10):5479. doi: 10.3390/ijerph18105479. PMID: 34065430; PMCID: PMC8160886.
14. E. Jana, R. Subban and S. Saraswathi, "Research on Skin Cancer Cell Detection Using Image Processing," 2017 IEEE International Conference on Computational Intelligence and Computing Research (ICCIC), Coimbatore, India, 2017, pp. 1-8, doi: 10.1109/ICCIC.2017.8524554.
15. A. Javaid, M. Sadiq and F. Akram, "Skin Cancer Classification Using Image Processing and Machine Learning," 2021 International Bhurban Conference on Applied Sciences and Technologies (IBCAST), Islamabad, Pakistan, 2021, pp. 439-444, doi: 10.1109/IBCAST51254.2021.9393198.
16. <https://www.kaggle.com/datasets/kmader/skin-cancer-mnist-ham10000>