International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 08 Issue: 07 | July - 2024 SJIF Rating: 8.448 ISSN: 2582-3930

Categorizing Dermatological Malignancies Via Computational Methods

Jahnavi Raghava Singh[1], J Gopi[2],V.Anil Santosh[3], DDD Suri Babu[4]

[1]M.Tech Student, Department of CSE, ISTS Engineering College, Rajahmundry, India
[2]Associate Professor, Department of CSE, ISTS Engineering College, Rajahmundry, India
[3]Associate Professor, Department of CSE, ISTS Engineering College, Rajahmundry, India
[4]Associate Professor, Department of CSE, ISTS Engineering College, Rajahmundry, India

ABSTRACT

In this study, a machine learning model is developed to classify different types of cancer using convolutional neural networks (CNNs) for image processing. The core objective is to achieve a performance level comparable to that of dermatologists. The model is trained on a substantial dataset of medical images, enabling it to learn and recognize various characteristics indicative of different cancer types. By leveraging the power of CNNs, the model can process these images effectively, identifying subtle patterns and features that are often challenging to detect with the naked eye. The training process involves feeding the CNN with labelled images, enabling it to differentiate between benign and malignant cases with high accuracy. Through rigorous testing, the model demonstrates competence on par with experienced dermatologists, both in terms of sensitivity and specificity. This equivalence in performance is particularly significant as it underscores the model's potential to aid in clinical settings, providing reliable second opinions and enhancing diagnostic workflows. A user interface is also developed to allow input images to be analysed by the trained CNN model. This interface not only displays the model's predictions but also provides essential metrics such as confidence scores and probability distributions. These metrics offer valuable insights into the model's decision-making process, aiding clinicians in understanding and trusting the AI's assessments. Overall, the findings suggest that convolutional neural networks hold substantial promise for improving cancer diagnosis. The model's high performance in classification tasks demonstrates its viability as a tool for supporting dermatologists in clinical practice. By reducing diagnostic errors and accelerating the identification process, this technology has the potential to significantly impact patient outcomes and advance the field of medical imaging and diagnostics.

Keywords: Convolutional Neural Networks (CNNs); Cancer Classification; Medical Image Processing; Dermatology AI; Diagnostic Accuracy

1.0 Introduction

Escalating air pollution and ozone depletion have led to a concerning rise in skin cancer cases, surpassing all other cancer types combined. Melanoma, a particularly deadly form, exhibits an alarmingly high mortality rate. The science behind skin cancer revolves around melanin, a pigment produced by melanocytes in the skin's layers. Melanin levels and types vary among individuals, contributing to diverse skin tones. Beyond imparting colour, melanin serves as a protective barrier against the sun's harmful ultraviolet radiation. However, excessive exposure can overwhelm this natural defence mechanism, increasing the risk of skin cancer development. Factors contributing to skin cancer include prolonged exposure to UV rays, numerous or unusual moles, specific skin types, and a family history of melanoma. Although melanoma's mortality rate is high, early diagnosis increases the survival rate to 99% [1], [2]. Diagnosing whether a lesion is benign or malignant can be challenging, even for experienced dermatologists, due to their similar appearances. Dermatologists employ techniques such as the ABCD rule (Atypical, Border, Colour, and Diameter) to improve classification accuracy, but human expertise remains essential [3]. Dermatologists generally discourage frequent use of biopsies. According to the International Skin Imaging Collaboration, the

number of unnecessary culture tests performed varies widely based on several factors, including clinical settings, dermatologist expertise, and the technology used. For example, among young individuals, where melanoma rates are quite low, about 500,000 culture tests are conducted annually to detect roughly 400 melanomas [4]. Advances in computer procedures and machine learning assist dermatologists in the early detection of melanoma, helping to reduce the high costs associated with melanoma detection and the need for unnecessary biopsies. Innovative automatic melanoma detection systems save considerable time, money, and effort. Machine learning has demonstrated the ability to provide melanoma classification with significantly improved accuracy. While advancements in dermatological equipment have improved melanoma classification accuracy, technological progress in machine learning and image processing has led to significant breakthroughs in the diagnosis, detection, and classification of melanoma, enhancing accuracy and reliability considerably. The literature review highlights various approaches used to create computer-aided diagnostic systems for classifying skin cancer using Dermoscopic images. These systems determine whether a lesion is benign or malignant. Hiam Alquran et al. [5] introduced a skin cancer detection system utilizing OTSU Thresholding. After image preprocessing and segmentation, they extracted features such as ABCD (Atypical, Border, Colour, and Diameter), Total Dermoscopic Score (TDS), Circulation, and texture features using the Gray Level Co-occurrence Matrix (GLCM). They employed the SVM classifier with the RBF Kernel algorithm, achieving a classification accuracy of 92.1%. Uzma Bano et al. [6] developed another skin cancer detection system based on image processing techniques. Following preprocessing, the images underwent maximum entropy thresholding for lesion segmentation. They extracted texture features using GLCM and used the SVM classifier, achieving a classification accuracy of 95%.M. Attique Khan et al. [7] created an automatic skin cancer detection and classification system based on normal distribution for image segmentation, following some preprocessing steps. They extracted shape features using the Histogram of Gradients (HOG), texture features using Haralick features, and colour features. For dimensionality reduction, they applied entropy-controlled feature selection. They tested various classifiers to determine classification accuracy and found that the multi-class SVM classifier yielded the best results. They achieved classification accuracies of 97.5% on the PH2 dataset, 97.75% on the ISIC (UDA, MSK-2) dataset, and 93.2% on the combined ISBI 2016-17 dataset. Vijayalakshmi [8] developed an automatic skin cancer detection system utilizing image processing and artificial intelligence with 1000-1500 images from the publicly available ISIC dataset. After initial preprocessing, lesion segmentation was performed using OTSU, Modified OTSU, and watershed segmentation methods. She evaluated performance using multiple classifiers, including the Back Propagation Algorithm (Neural Networks), SVM, and CNN, achieving a maximum accuracy of 85% with the SVM classifier.

Dalila et al. [9] introduced a system for segmenting and classifying melanoma and benign skin lesions using an Ant-Colony based segmentation algorithm. They extracted features related to shape, texture, and colour, and used KNN and ANN classifiers, achieving classification accuracies of 85.22% and 93.60%, respectively. Sumithra R et al. [10] developed a skin cancer diagnosis system based on region-growing segmentation techniques. Following preprocessing and segmentation, they extracted colour, texture, and RGB histogram features. For the classification phase, they employed three classifiers: KNN, SVM, and a combined SVM+KNN approach, achieving classification accuracies of 86%, 87.5%, and 94%, respectively. Many researchers have relied on their datasets collected from hospitals, but these datasets are often too limited for adequate training during the machine learning phase. Some researchers, however, have used the ISIC dataset, which offers a substantial amount of data and is more practical for training purposes. The aim of this research is to discover improved and more effective methods for detecting skin cancer using machine learning techniques. The ultimate goal is to assist doctors in the early detection of skin cancer by providing more reliable and accurate results. The proposed system aims to achieve high accuracy in classifying lesions as benign or malignant, which will significantly aid in efficiently identifying patients.

2.0 Methodology

The research employs Convolutional Neural Networks (CNNs) to classify tumour types based on lesion images, particularly focusing on melanoma skin cancer, known for its potential spread. Recognizing the challenges posed by fine-grained differences in skin lesions, the study emphasizes automated classification through image analysis. To achieve this, CNNs are utilized to process images and extract relevant features, facilitating highly segregated tasks against finely grained object categorization. The methodology involves training the system over these extracted features using machine learning techniques, aiming to develop a robust and accurate model for cancer type detection in patients.

The Functional Requirements Specification delineates essential system operations and activities, catering to a non-technical audience. Users input lesion images via a user-friendly interface without requiring registration. The interface is developed using the Flask framework, facilitating easy creation of web applications with Python. Outputs consist of lesion labels corresponding to input images. Non-functional requirements encompass system reliability, ensuring consistent and accurate outputs. High performance is prioritized, aiming for swift results, while modifiability allows for easy system updates. Software specifications mandate Windows 10 as the operating system, Jupyter Notebook (Anaconda) as the Integrated Development Environment (IDE), and Python as the coding language. The framework employed is Flask, with Keras over TensorFlow utilized for machine learning tasks. These specifications are crucial for optimal system performance, requiring separate installation before the software deployment. TensorFlow, an open-source library, employs data-flow graphs for numerical computation. Initially developed by Google Brain Team, it rapidly evolved since its February 2017 debut, boasting over 21,000 commits and contributions from external collaborators. It's widely adopted across domains, reflecting its significance in machine learning and numerical computation.

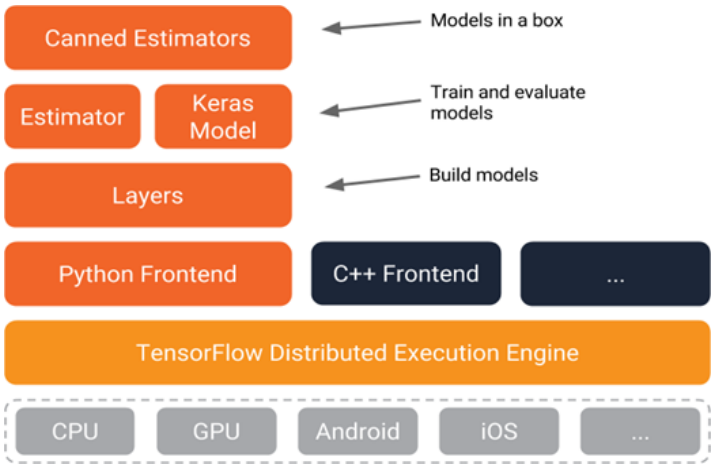


Fig .1TensorFlow architecture

Flask, a Python web framework, spearheaded by Armin Ronacher, utilizes the Werkzeug WSGI toolkit and Jinja2 template engine. Hardware specifications include an i5 processor, 4GB RAM, and 5GB hard disk.

2.1 System Architecture

System architecture outlines a system's structure, behavior, and interactions, facilitating comprehension, design, and communication of complex system functionalities.

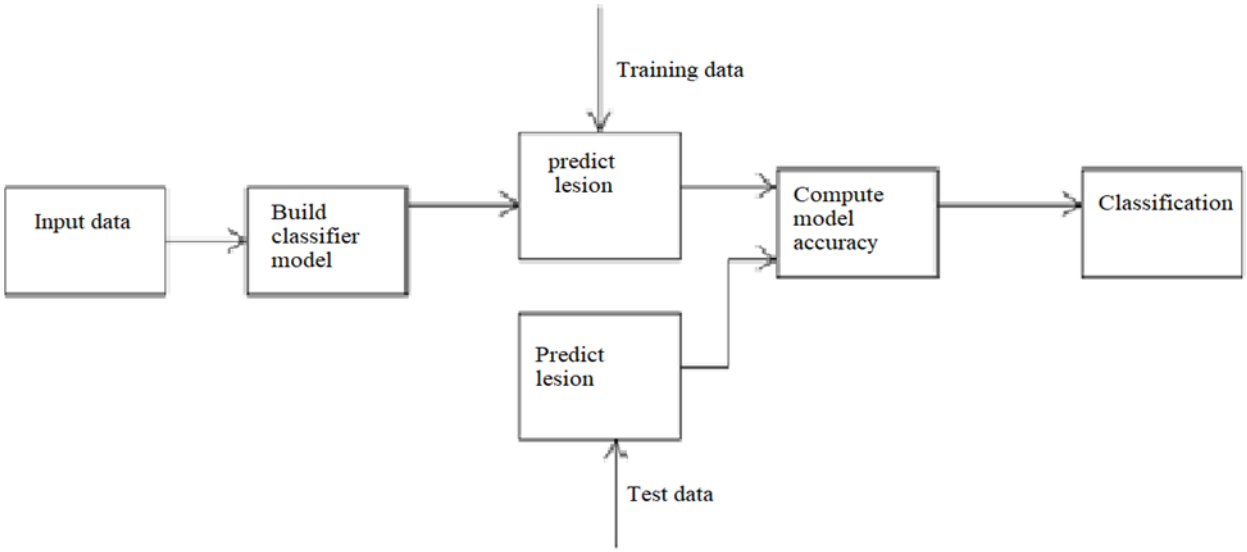


Fig 2 system architecture

2.2 System Design

The Unified Modelling Language (UML) provides a standardized notation for expressing analysis models, crucial for visualizing and constructing software-intensive systems amidst rising complexity. UML serves as a formal language guiding the construction of software systems, facilitating comprehension and timely application delivery. While UML doesn't enforce formal methodologies, it offers various diagrams enhancing system understanding. These include Use Case diagrams, depicting system functionality and user interactions, Sequence diagrams illustrating class interactions over time, and Activity diagrams emphasizing internal process flows. Use Case diagrams particularly aid in articulating high-level system requirements and identifying user-system interactions, enhancing communication among stakeholders during software development's early stages.Within the system design context, UML's structured approach organizes system representation into five views: User Model, Structural Model, Behavioural Model, Implementation Model, and Environmental Model. Each view offers a unique perspective on system architecture and behavior, complemented by corresponding UML diagrams. Use cases within the system, including input dataset, pre-processing data, partitioning dataset, executing the CNN model, and prediction, are outlined, providing a clear delineation of system functionalities. This systematic approach aids in effective system development, ensuring comprehensive understanding and communication among stakeholders.

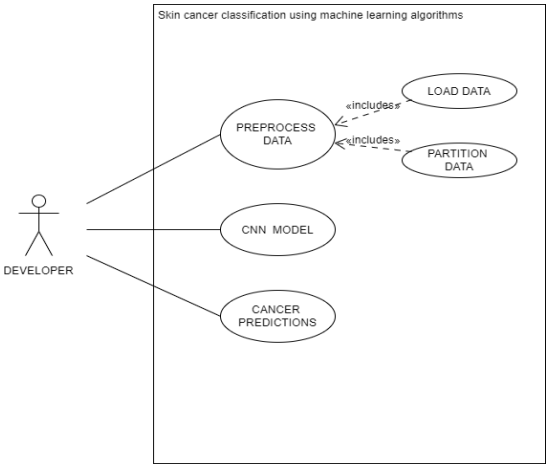


Fig.3 Use-case diagram for developer

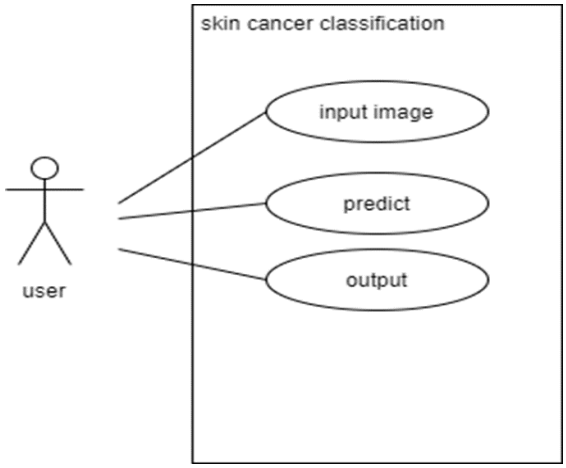


Fig.4 Use-case diagram for user

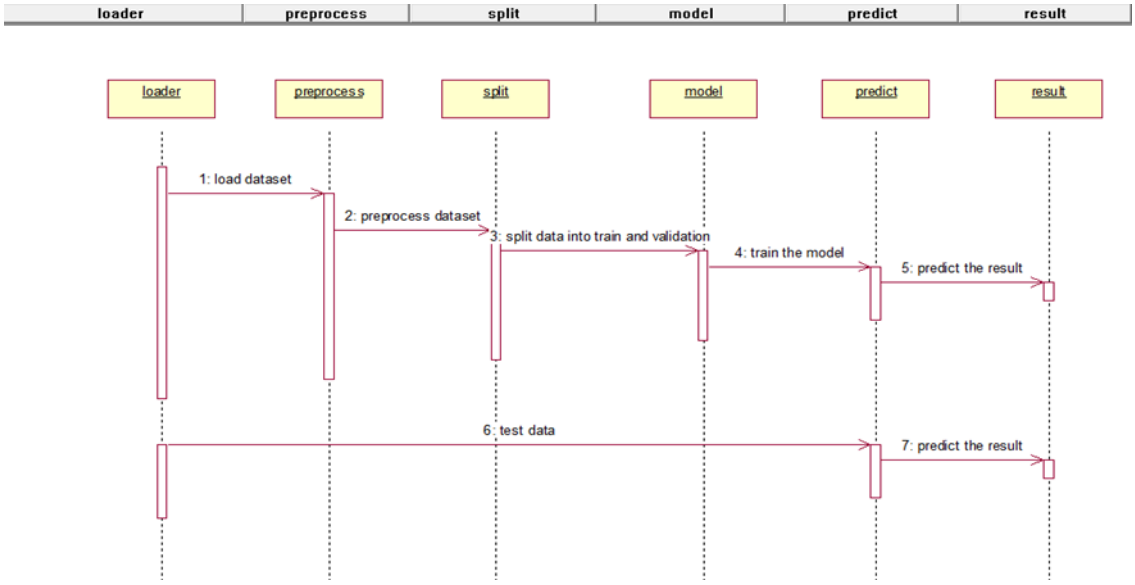


Fig.5 Sequence diagram

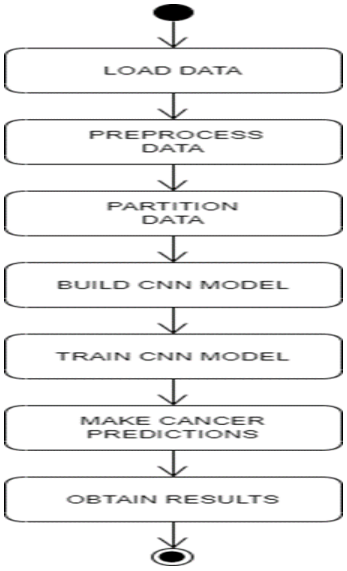


Fig.6 Activity diagram

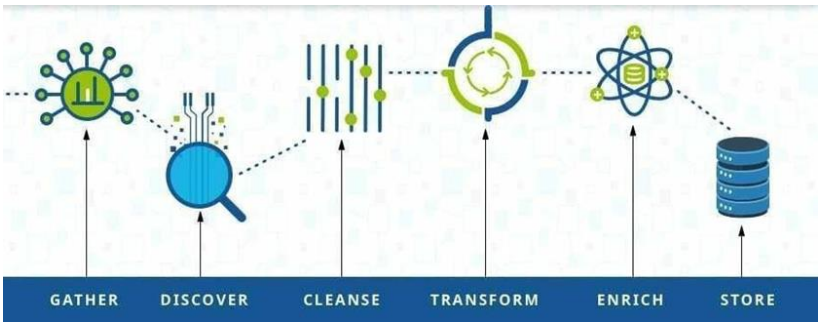


Fig.7 Data Pre-processing

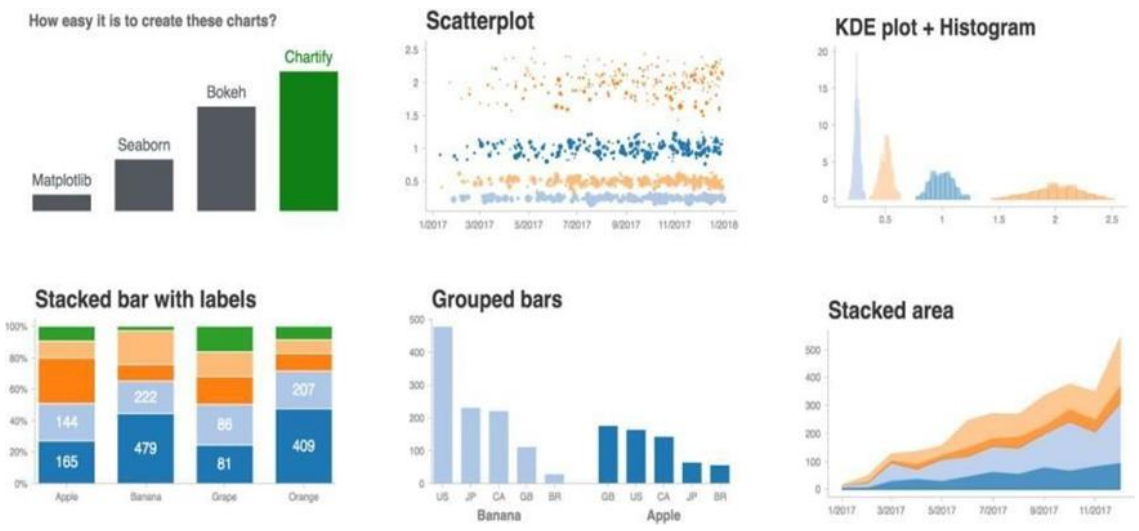


Fig.8 Data Visualization

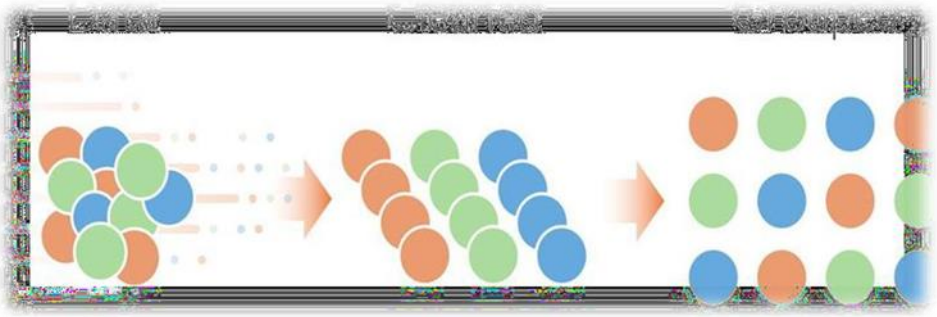


Fig.9 Data Transformation

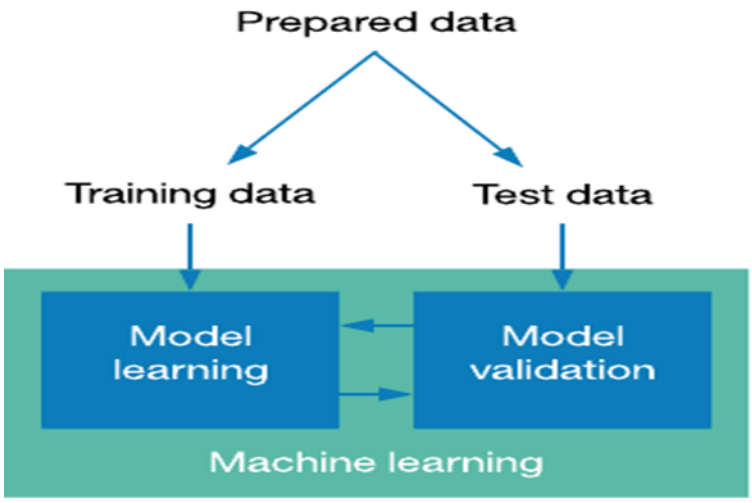


Fig.10 Data Splitting

3.0 Analysis of data

```
In [4]: # Now lets see the sample of tile_df to look on newly made columns
        skin_df.head()
```

Out[4]:

	lesion_id	image_id	dx	dx_type	age	sex	localization	path	cell_type	cell_type_idx
0	HAM_0000118	ISIC_0027419	bkl	histo	80.0	male	scalp	C:/Users/santo\input1\HAM10000_images_part_1\I...	Benign keratosis-like lesions	0
1	HAM_0000118	ISIC_0025030	bkl	histo	80.0	male	scalp	C:/Users/santo\input1\HAM10000_images_part_1\I...	Benign keratosis-like lesions	0
2	HAM_0002730	ISIC_0026769	bkl	histo	80.0	male	scalp	C:/Users/santo\input1\HAM10000_images_part_1\I...	Benign keratosis-like lesions	0
3	HAM_0002730	ISIC_0025661	bkl	histo	80.0	male	scalp	C:/Users/santo\input1\HAM10000_images_part_1\I...	Benign keratosis-like lesions	0
4	HAM_0001466	ISIC_0031633	bkl	histo	75.0	male	ear	C:/Users/santo\input1\HAM10000_images_part_2\I...	Benign keratosis-like lesions	0

Fig.11 Sample data

```
In [5]: skin_df.isnull().sum()

Out[5]: lesion_id       0
        image_id        0
        dx              0
        dx_type         0
        age             4
        sex             0
        localization    0
        path            0
        cell_type       0
        cell_type_idx   0
        dtype: int64
```

Fig 12. Checking null values

```
In [9]: fig, ax1 = plt.subplots(1, 1, figsize= (10, 5))
        skin_df['cell_type'].value_counts().plot(kind='bar', ax=ax1)
Out[9]: <matplotlib.axes._subplots.AxesSubplot at 0x1897b0c20f0>
```

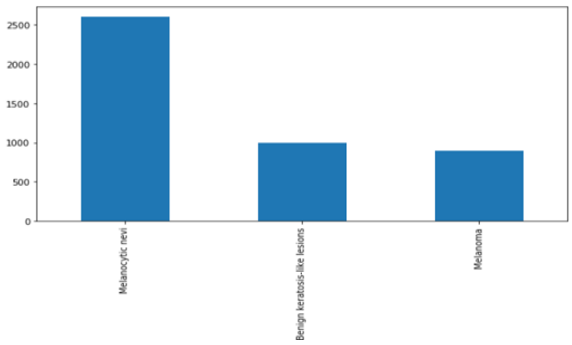
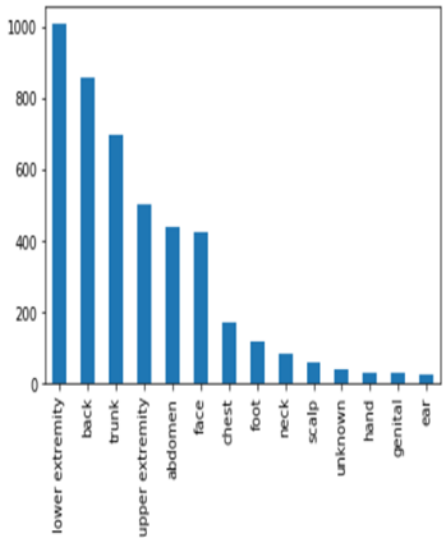


Fig.13 Bar chart of cancer labels

```
In [11]: skin_df['localization'].value_counts().plot(kind='bar')
Out[11]: <matplotlib.axes._subplots.AxesSubplot at 0x1897b0a8668>
```



```
In [10]: skin_df['dx_type'].value_counts().plot(kind='bar')
Out[10]: <matplotlib.axes._subplots.AxesSubplot at 0x18903a485f8>
```

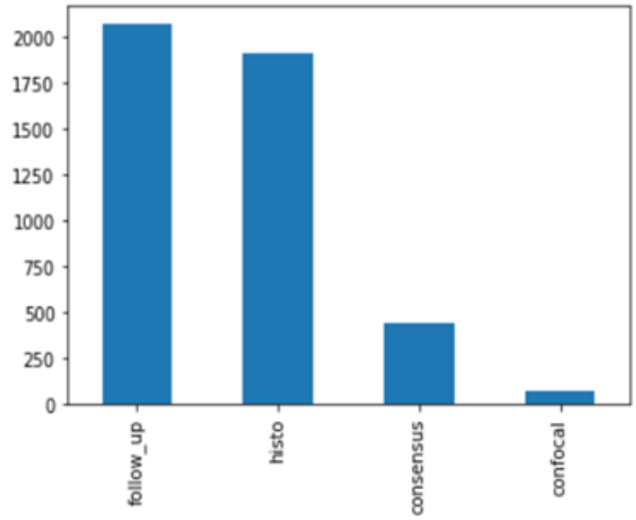


Fig.14 Bar chart for cancer localization

```
In [17]: n_samples = 5
fig, m_axs = plt.subplots(3, n_samples, figsize = (4*n_samples, 3*7))
for n_axs, (type_name, type_rows) in zip(m_axs,
                                         skin_df.sort_values(['cell_type']).groupby('cell_type')):
    n_axs[0].set_title(type_name)
    for c_ax, (_, c_row) in zip(n_axs, type_rows.sample(n_samples, random_state=1234).iterrows()):
        c_ax.imshow(c_row['image'])
        c_ax.axis('off')
fig.savefig('category_samples.png', dpi=300)
```

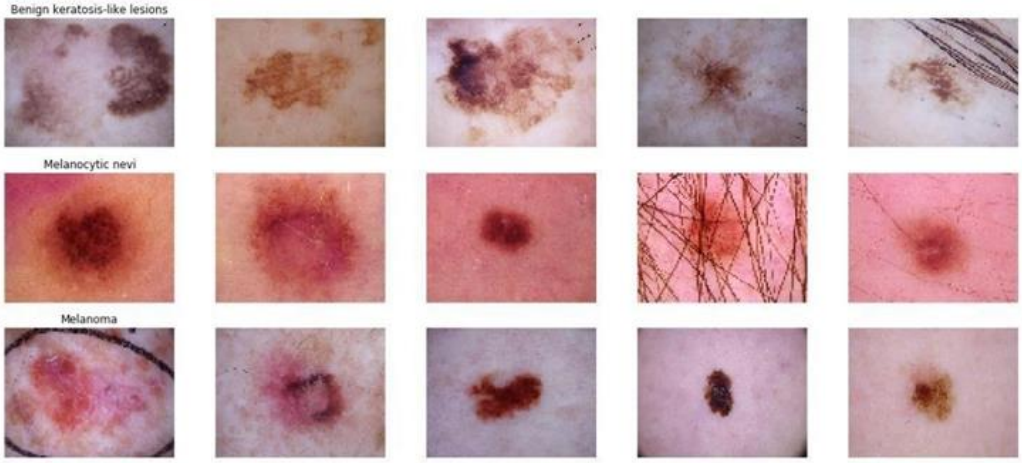


Fig.15 Sample lesion images

Fig.16 details

Layer (type)	Output Shape	Param #
conv2d_1 (Conv2D)	(None, 75, 100, 32)	896
conv2d_2 (Conv2D)	(None, 75, 100, 32)	9248
max_pooling2d_1 (MaxPooling2	(None, 37, 50, 32)	0
dropout_1 (Dropout)	(None, 37, 50, 32)	0
conv2d_3 (Conv2D)	(None, 37, 50, 64)	18496
conv2d_4 (Conv2D)	(None, 37, 50, 64)	36928
max_pooling2d_2 (MaxPooling2	(None, 18, 25, 64)	0
dropout_2 (Dropout)	(None, 18, 25, 64)	0
flatten_1 (Flatten)	(None, 28800)	0
dense_1 (Dense)	(None, 128)	3686528
dropout_3 (Dropout)	(None, 128)	0
dense_2 (Dense)	(None, 3)	387

Total params: 3,752,483
Trainable params: 3,752,483
Non-trainable params: 0

```
Epoch 00014: ReduceLROnPlateau reducing learning rate to 0.0002500000118743628.
Epoch 15/25
44/44 [==============================] - 82s 2s/step - loss: 0.4291 - acc: 0.8186 - val_loss: 0.5005 - val_acc: 0.7900
Epoch 16/25
44/44 [==============================] - 82s 2s/step - loss: 0.4188 - acc: 0.8311 - val_loss: 0.5096 - val_acc: 0.7886
Epoch 17/25
44/44 [==============================] - 82s 2s/step - loss: 0.4091 - acc: 0.8208 - val_loss: 0.5336 - val_acc: 0.7747
Epoch 18/25
44/44 [==============================] - 83s 2s/step - loss: 0.4271 - acc: 0.8169 - val_loss: 0.5088 - val_acc: 0.8011
Epoch 19/25
44/44 [==============================] - 83s 2s/step - loss: 0.4277 - acc: 0.8211 - val_loss: 0.5124 - val_acc: 0.7803
Epoch 20/25
44/44 [==============================] - 82s 2s/step - loss: 0.4037 - acc: 0.8223 - val_loss: 0.4860 - val_acc: 0.7955
Epoch 21/25
44/44 [==============================] - 83s 2s/step - loss: 0.3885 - acc: 0.8417 - val_loss: 0.5113 - val_acc: 0.7886

Epoch 00021: ReduceLROnPlateau reducing learning rate to 0.0001250000059371814.
Epoch 22/25
44/44 [==============================] - 82s 2s/step - loss: 0.3923 - acc: 0.8310 - val_loss: 0.4992 - val_acc: 0.8053
Epoch 23/25
44/44 [==============================] - 82s 2s/step - loss: 0.3944 - acc: 0.8290 - val_loss: 0.5028 - val_acc: 0.7844
Epoch 24/25
44/44 [==============================] - 82s 2s/step - loss: 0.3815 - acc: 0.8385 - val_loss: 0.5135 - val_acc: 0.7900
Epoch 25/25
44/44 [==============================] - 83s 2s/step - loss: 0.3909 - acc: 0.8358 - val_loss: 0.4835 - val_acc: 0.8011
Epoch 1/25
44/44 [==============================] - 180s 4s/step - loss: 0.8630 - acc: 0.6325 - val_loss: 0.6761 - val_acc: 0.6732
Epoch 2/25
44/44 [==============================] - 168s 4s/step - loss: 0.6720 - acc: 0.6876 - val_loss: 0.6326 - val_acc: 0.7038
Epoch 3/25
44/44 [==============================] - 116s 3s/step - loss: 0.6621 - acc: 0.6825 - val_loss: 0.6838 - val_acc: 0.6926
Epoch 4/25
44/44 [==============================] - 84s 2s/step - loss: 0.6292 - acc: 0.7198 - val_loss: 0.5918 - val_acc: 0.7441
Epoch 5/25
44/44 [==============================] - 83s 2s/step - loss: 0.6179 - acc: 0.7393 - val_loss: 0.6262 - val_acc: 0.7441
Epoch 6/25
44/44 [==============================] - 82s 2s/step - loss: 0.5961 - acc: 0.7379 - val_loss: 0.5711 - val_acc: 0.7636
Epoch 7/25
44/44 [==============================] - 83s 2s/step - loss: 0.5706 - acc: 0.7520 - val_loss: 0.5413 - val_acc: 0.7677
Epoch 8/25
44/44 [==============================] - 82s 2s/step - loss: 0.5623 - acc: 0.7621 - val_loss: 0.5362 - val_acc: 0.7677
Epoch 9/25
44/44 [==============================] - 82s 2s/step - loss: 0.5198 - acc: 0.7777 - val_loss: 0.5782 - val_acc: 0.7455
Epoch 10/25
44/44 [==============================] - 81s 2s/step - loss: 0.5228 - acc: 0.7770 - val_loss: 0.5798 - val_acc: 0.7524
```

Fig.17 Model Training

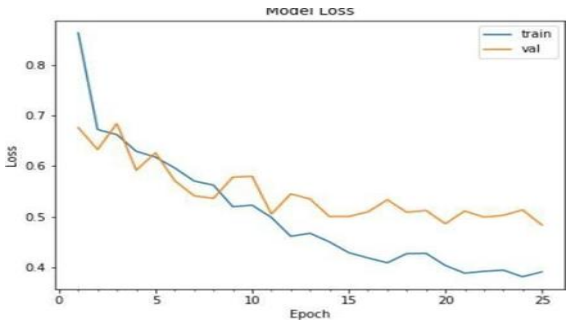


Fig.18 Accuracy curve

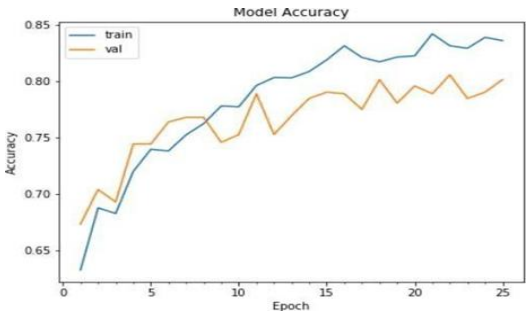


Fig.19 Loss curve

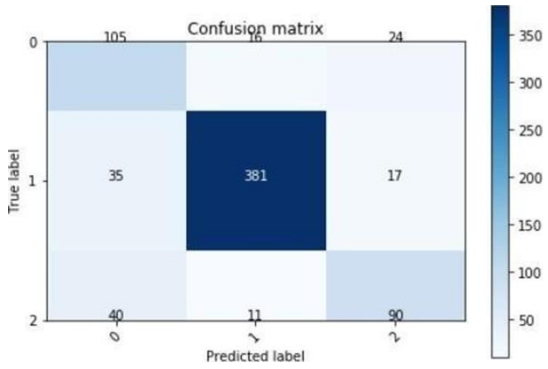


Fig.20 Confusion matrix

3.0 Conclusion

This study investigates the classification of skin cancer types using machine learning algorithms. The dataset utilized is sourced from an international collaboration focused on skin imaging, comprising lesion images paired with corresponding cancer labels. The approach involves constructing a Convolutional Neural Network (CNN) model from scratch, subsequently training it with the provided dataset and evaluating its performance using test data. The CNN model demonstrates a commendable accuracy rate of 84%, indicating its effectiveness in accurately classifying skin cancer types based on lesion images. The utilization of machine learning algorithms in skin cancer classification represents a significant advancement in medical diagnostics. By leveraging the inherent patterns and features present in lesion images, the CNN model can discern subtle differences indicative of various cancer types. This capability holds immense promise for improving diagnostic accuracy and expediting treatment decisions in clinical settings. Furthermore, the development of a user interface using Flask facilitates the deployment of the trained CNN model for practical use. The interface allows users to input lesion images and receive corresponding cancer labels as output. This seamless integration of the CNN model into a user-friendly interface enhances accessibility and usability, enabling clinicians and healthcare professionals to leverage the model's predictive capabilities effectively. The achieved accuracy rate of 84% underscores the robustness and reliability of the CNN model in skin cancer classification tasks. While further refinements and optimizations may be explored to enhance performance, the current results highlight the potential of machine learning algorithms in augmenting diagnostic processes and improving patient outcomes in dermatology.

4.0 References

1.D. Schadendorf et al., "Melanoma," Lancet, vol. 392, no. 10151, pp. 971–984, 2018.

2. S. Gupta and H. Tsao, "Epidemiology of melanoma," Pathol. Epidemiol. Cancer, pp. 591–611, 2016

3. F. Nachbar et al., "The ABCD rule of dermatoscopy: high prospective value in the diagnosis of doubtful melanocytic skin lesions," J. Am. Acad. Dermatol., vol. 30, no. 4, pp. 551–559, 1994.

4. lesions," J. Am. Acad. Dermatol., vol. 30, no. 4, pp. 551–559, 1994. "The International Skin Imaging Collaboration (ISIC)." https://www.isic archive.com/#!/topWithHeader/tightContentTop/about/isicArchive.

5. H. Alquran et al., "The melanoma skin cancer detection and classification using support vector machine," 2017 IEEE Jordan Conf. Appl. Electr. Eng. Comput. Technol. AEECT 2017, vol. 2018 Janua, pp. 1–5, 2017

6. U. B. Ansari and M. E. Student, "Skin Cancer Detection Using Image Processing Tanuja Sarode 2," Int. Res. J. Eng. Technol., vol. 4, no. 4, pp. 2395–56, 2017, [Online]. Available: https://www.irjet.net/archives/V4/i4/IRJET-V4I4702.pdf.

7. M. A. Khan et al., "An implementation of normal distribution based segmentation and entropy controlled features selection for skin lesion detection and classification," BMC Cancer, vol. 18, no. 1, pp. 1–20, 2018.

8. 1–20, 2018. V. M. M, "Melanoma Skin Cancer Detection using Image Processing and Machine Learning," Int. J. Trend Sci. Res. Dev., Volume-3, no. Issue-4, pp. 780–784, 2019.

9. F. Dalila, A. Zohra, K. Reda, and C. Hocine, "Segmentation and classification of melanoma and benign skin lesions," Optik (Stuttg)., vol. 140, pp. 749–761, 2017.

10.R. Sumithra, M. Suhil, and D. S. Guru, "Segmentation and classification of skin lesions for disease diagnosis," Procedia Comput. Sci., vol. 45, no. C, pp. 76–85, 2015.

11. N. V Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: synthetic minority over-sampling technique," J. Artif. Intell. Res., vol. 16, pp. 321–357, 2002.

12. S. Aksoy and R. M. Haralick, "Feature normalization and likelihood-based similarity measures for image retrieval," Pattern Recognit. Lett., vol. 22, no. 5, pp. 563–582, 2001.

13. X. H. Cao and Z. Obradovic, "A robust data scaling algorithm for gene expression classification," 2015 IEEE 15th Int. Conf. Bioinforma. Bioeng. BIBE 2015, no. 1, 2015.

14.G. John, R. Kohavi, and K. Pfleger, "IrreleJohn, G., Kohavi, R., & Pfleger, K. (1994). Irrelevant Features and the Subset Selection Problem. Icml, 121–129. Retrieved from http://machine learning.martinsewell.com/feature selection/JohnKohaviPfleger1994.pdfvant Features and the Subset Selectio," Icml, pp. 121–129, 1994, [Online]. Available: http://machine-learning.martinsewell.com/feature selection/JohnKohaviPfleger1994.pdf.

15.Y. Masoudi-Sobhanzadeh, H. Motieghader, and A. Masoudi-Nejad, "FeatureSelect: A software for feature selection based on machine learning approaches," BMC Bioinformatics, vol. 20, no. 1, pp. 1–17, 2019.

16. Vatsala Singh and Ifeoma Nwogu, "Analysing Skin Lesions in Dermoscopy Images Using Convolutional Neural Networks", 2018 IEEE International Conference on Systems, Man, and Cybernetics

17. Mobeenur Rehman, SharzilHaris Khan, S M Danish Rizvi, Zeeshan Abbas&Adil Zafar, "Classification of Skin Lesion by interference of Segmentation and Convolution Neural Network", 2018 IEEE