Novel Approach with Deep Learning Models For Melanoma Skin Cancer Detection

Shankar M. Patil,
Professor, Smt Indira Gandhi College of Engineering,
Navi Mumbai, Maharashtra, India
smpatil2k@gmail.com

Sushopti Gawade,
Professor Vidyalankar Institute of Technology,
Wadala, Maharashtra, India
sushoptiekrishimitra@gmail.com

Sunil S. Chavan,
Professor, Smt Indira Gandhi College of Engineering,
Navi Mumbai, Maharashtra, India chavanss@hotmail.com

Umakant Gohatre,
Assistant Professor, Smt Indira Gandhi College of Engineering,
Navi Mumbai, Maharashtra, India
umakantbhaskar@gmail.com

***Abstract*: Artificial intelligence methodologies with deep learning models proved very effective in many areas. This research focuses on using deep learning models for the detection of skin cancer melanoma in patients at an early stage. The growth of a malignant melanocytic tumor is the primary cause of melanoma, a deadly form of skin cancer. The most dangerous type of cancer, known as melanoma, is brought on by melanocytes, which produce pigment. Seventy-five percent of skin cancer fatalities are caused by melanoma. Only 15% of patients who have survived over the previous five years, according to a review of the survival rate, have received chronic treatment. The manual system is the issue. The mole has a maximum of six colors, thus the image is troublesome since it has a variety of tints and is hard for humans to distinguish. As a result, we created a model utilizing flask and a pre-trained deep-learning model to handle this issue. The suggested model is optimized using the SGD, RMSprop, Adam, Agagrad, Adadelta, and Nadam optimizers. According to testing, the suggested CNN model with the Adam optimizer performs the best at categorizing the dataset of skin cancer lesions. Also it gives the result specifically to the seven types of the skin cancer with their possible chances in percentage. It achieved a training accuracy of 99.73 % and a testing accuracy of 96.53, which is better than the previous results. In order to lower the mortality rate, this research intends to identify an early treatment for people with skin cancer.**

Keywords— Patients, Melanoma, deep learning models, skin cancer, malignant tumor.

I. Introduction

Melanoma is a dangerous type of skin cancer that develops when melanocytes—the pigment-producing cells that give skin its color—transform malignantly. It mostly affects skin areas that are exposed to ultraviolet (UV) light, which increases the likelihood that it may spread quickly. Among cases of skin cancer, melanoma is the most common cause of death. Merely 15% of patients have seen a lasting remission after therapy in the chronic stage over the last five years, indicating that the outlook is still dire despite advancements in treatment. The continuous increase in the prevalence of melanoma over the past 25 years exacerbates this problem and emphasizes the urgent need for an early detection system.

Traditional detection techniques have inherent drawbacks, especially when it comes to picking up on minute differences in mole pigmentation, which can range up to six different colors. Furthermore, important diagnostic characteristics such as asymmetry, uneven borders, and abrupt transitions require accurate lesion boundary delineation, which is frequently difficult for human observers.

A suggested solution leverages machine learning (ML) and artificial intelligence (AI) to meet this urgent need and get over inherent constraints, with the promise of better diagnostic efficacy than traditional manual methods. Making use of deep learning's widespread availability and remarkable performance—which includes an average 98% accuracy rate—becomes an appealing option for early-stage ailment identification. Through the use of advanced techniques like Flask, Python, and Deep Learning frameworks, the research hopes to make this novel methodology possible and enable automated melanoma diagnosis with improved efficiency and accuracy.

II. Literature Survey

A color improvement using the RGB color channel was proposed by Jinen Daghrir et al.[1] Additionally, the CNN approach, Morphological Snakes, and 2-D Gaussian derivatives (2D-GOD) are employed for skin lesion segmentation as well as hair identification. The automated melanoma detection methods have not yet looked into it. Due to a shortage of tagged data, their algorithms cannot be trained.

S. M. Patil, et al.[2], A pretrained DL model-based method for early skin cancer detection is suggested. A Flask website is created in this effort to enable users to post dermatological photos and guess the class to which they belong.

In addition to working on DE-ANN and pre-processing, Manoj Kumar et al. [3] also employed fuzzy C- Means clustering to picture improvement.

According to a study by Ahmad Naeem et al. [4] that identifies current research trends, difficulties, and possibilities for melanoma diagnosis and looks at available solutions, CNN is a superior approach for melanoma detection. Along with the ABCD rule, pre-processing includes image segmentation using the Otsu thresholding method, SVM, and pre-processing with SVM classification [5].

Patil S.M., Malik A.K. [6] It has become essential to establish methods for resource use in order to make informed judgements due to the quest for improved performance and reliability.

Chopade, & Gawade (2022) [7] worked on a paper where they used machine learning to predict end-semester results. They used algorithms like SVM, KNN and DT. The SVM algorithm's accuracy was about 78 %.

Shankar M. Patil and Dr. Praveen Kumar [8]The first and most significant step is data collection and analysis because it is essential for decision support in all industries.

Yashodhan Ketkar and Sushopti Gawade (2022) [9] build algorithms that allow users with basic knowledge to access and understand machine learning technology easily. The model with more weight will be selected for a particular task. The performance of all models is stored so it can be used for future evaluation of the system.

According to Dang N. H. Thanh et al.[10], picture segmentation and pre-processing are carried out to assess the skin lesions score using the ABCD rule.

Yashodhan Ketkar, and Sushopti Gawade (2022) [11] built a method that allowed users to train the model that will be most appropriate for the challenge. The researchers claim that it can also be used by medical experts to identify unusual arrhythmias in patients.

The authors, Marathe, Gawade, and Kanekar (2021) [12], employed supervised learning techniques. to identify diabetes and heart disease early. Marathe et al. came to the conclusion that the model functioned as expected.

Verma & Gawade (2021) [13] predicted crop growth rates using machine learning systems. They were successful in gaining useful insights on the subject. Verma and Gawade came to the conclusion that applying machine learning to farming would reduce complexity while enhancing productivity.

Sushopti Gawade et al. (2023) [14] used various deep learning models and CNN algorithms for bone cancer disease detection.

The paper by S. Satre et al.[15] presents an investigation into the development of an Online Exam Proctoring System utilizing Artificial Intelligence (AI) technology. the paper elucidates the efficacy and feasibility of leveraging AI for online exam proctoring, offering insights into its potential applications and implications in the educational landscape.

Shankar M. Patil et al.[16] the study endeavors to devise an automated system capable of accurately categorizing fruits based on their condition, such as ripeness or spoilage. Through the application of machine learning algorithms.

Uma Sharma et al.[17] explore the realm of fake news detection employing machine learning algorithms. Focusing on the pervasive issue of misinformation dissemination, the study aims to develop an automated system capable of discerning between genuine news articles and fabricated or misleading content by leveraging machine learning techniques.

Temidayo Oluwatosin Omotehinwa, David Opeoluwa Oyewola, Emmanuel Gbenga Dada[18] focuses on breast cancer diagnosis with tree structured Parzen Estimator .

Sandeep W. and Dileepkumar S.[19] used machine learning algorithms to detect lung cancer from medical images. B. Jaishankar, Ashwini A.M., Vidyabharathi D., L. Raja[20] In this paper, a novel epilepsy seizure prediction approach is designed using deep learning. The proposed model is applied to the Electroencephalogram (EEG) recordings collected from Children's Hospital Boston (CHB-MIT).

III. Experimental Design

A. System Architecture

The system architecture is outlined comprehensively in this section, comprising both a conceptual diagram and a block diagram, depicted respectively in Figures 1 and 2.

The overall structure and functions of the system are visualized in the conceptual diagram. It probably shows the system's high-level elements and how they interact with one another. This schematic acts as a road map, providing an overview of how various components work together to achieve the system's goals. Usually, it concentrates on the connections and interactions between different modules or subsystems while abstracting away complex technological specifics.

The block diagram presents different components as separate blocks or modules and provides a more thorough explanation of the system architecture. Every block in the system represents a particular function or group of related functions. Block connections show how data, control signals, or other pertinent information moves across a system. Understanding the internal workings of the system and how various parts interact to produce desired results is made easier with the help of this graphic. It offers a detailed viewpoint that makes analysis, design, and troubleshooting tasks easier.

The block diagram presents different components as separate blocks or modules and provides a more thorough explanation of the system architecture. Every block in the system represents a particular function or group of related functions. Block connections show how data, control signals, or other pertinent information moves across a system. Understanding the internal workings of the system and how various parts interact to produce desired results is made easier with the help of this graphic. It offers a detailed viewpoint that makes analysis, design, and troubleshooting tasks easier.

B. Conceptual Block Diagram

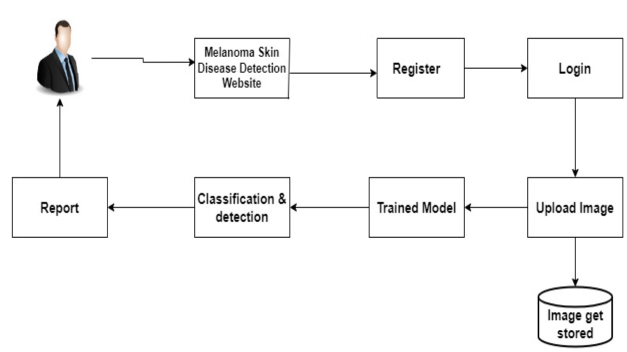


Fig. 1. Conceptual Diagram for Melanoma skin cancer detection

C. Block Diagram:

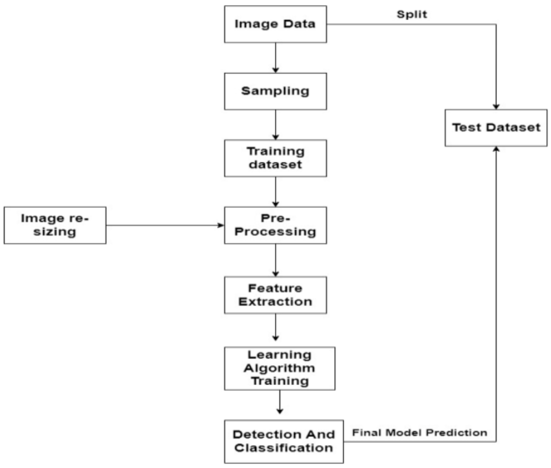


Fig. 2. Block Diagram for Melanoma skin cancer detection

IV. SYSTEM METHODOLOGY

A. Technology used (CNN)

The block diagram shows the system architecture in depth and distinguishes different parts as separate blocks or modules. Within the system, each block encapsulates a single function or a cohesive group of related functions. Data processing, computing jobs, communication protocols, and other operational facets essential to the system's operation are examples of these functions.

Block interconnections illustrate how information flows across the system, including data, control signals, and other relevant data. These connections delineate the pathways through which data flows, facilitating coordination and collaboration across diverse components. By visualizing these linkages, stakeholders can gain a better understanding of the dynamic interactions and dependencies that support the system's functionality.

This makes it easier to comprehend how the system functions inside and how different components work together to produce desired results.

This graphical representation helps one understand the internal workings of the system and how different elements work together to accomplish desired outcomes. It offers a well-organized summary that supports system analysis, design, and troubleshooting. The block diagram is a useful tool for engineers and developers to find possible bottlenecks, improve performance, or accurately diagnose problems..

To sum up, the block diagram provides a thorough understanding of the system's architecture, helping stakeholders to understand the nuances of its functioning and supporting well-informed decision-making at every stage of the development process.

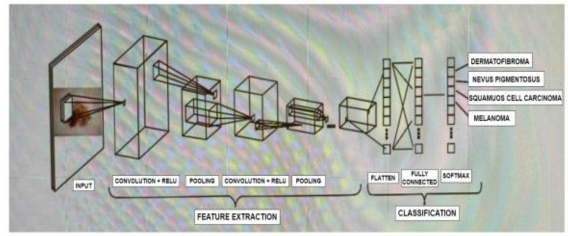


Fig. 3. Architecture of CNN

1) Convolution Layer:

The convolutional layer is the primary layer of CNN. The output layer's outcome is obtained from the input layer in this layer by filtering under particular circumstances. The neurons that make up this layer are shaped like cubical blocks.

Max-pooling layer: Following each convolution layer, the pooling layer performs the subsequent process. To reduce the size of the neurons, these layers are used. These are tiny rectangular grids that gather a tiny chunk of the convolutional layer and filter it to provide a result from that block.

The layer that fetches the most pixels from the block is max pooling, which is the most often utilized layer.

Fully Connected layers: A convolutional neural network's (CNN) A fully connected layer is the last layer. created by the attachment of all neurons before it. Given that it is fully linked like an artificial neural network, it lowers the spatial information. It is made up of neurons, starting with input neurons and ending with output neurons.

B. Implementation Platform:

Microsoft's free source code editor Visual Studio Code works with Linux, macOS, and Windows operating systems. It has several capabilities, including as code rewriting tools, snippets, integrated Git functionality, debugging support, intelligent code completion, and syntax highlighting. Users can also further personalize the editor by adding extensions to improve functionality and changing keyboard keys, themes, and preferences.

Create Jupyter notebook papers with the flexible web-based interactive computational platform called Jupyter Notebook. Initially introduced as IPython Notebooks, this platform offers an adaptable setting for carrying out data analysis, scientific investigations, and instructional tasks.

Depending on the context, the term "notebook" in the Jupyter context refers to several entities. It could refer to the Jupyter web application for creating and editing notebooks, the Jupyter document format itself, or the Jupyter Python web server that enables notebook interactions.

JSON files are used to organize Jupyter Notebook documents, which are normally saved with the ".ipynb" extension. These files are composed of a series of cells that can each contain either executable code or Markdown text along with the input/output that corresponds to it. This enables users to incorporate rich media such as narrative explanations, mathematics, code snippets, and visualizations into a single document.

Moreover, Jupyter notebooks have a versioned structure, which makes it possible for users to monitor modifications and work together efficiently. When working on collaborative projects or revisiting and upgrading studies over time, this versioning feature is quite helpful.

To sum up, Jupyter Notebook offers an effective way to create and distribute interactive documents that combine text, code, and visualizations seamlessly. This makes it possible to conduct repeatable research, create interactive teaching resources, and collaborate on data exploration projects.

Figure 4. illustrates a system composed of three layers. The top layer, serving as the input layer, contains the datasets utilized for training. Before passing data to the hidden layers, the input layer applies weighting to the information. Subsequently, a pattern is extracted from the data within the input layer to identify neurons in the hidden layers, aiding in feature separation. Finally, the output layers, constructed upon the extracted pattern, classify relevant classes.

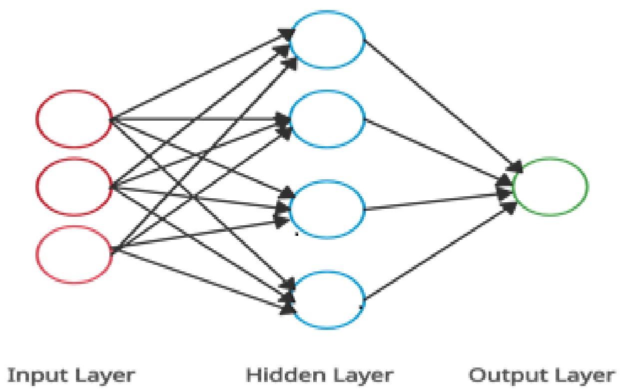
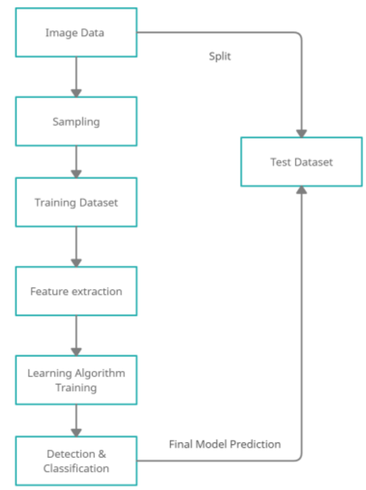


Fig. 4. Layer of Convolution Neural Network

In data preparation, One of the main issues with computational vision is the size of the images. Countless data points might be present in the input. The input feature dimension is up to 28x28x3 if the supplied photographs are of size 10,000 pixels. Depending on the number of hidden units involved, processing images with dimensions like 1024x1024x3 successfully for a deep neural network, especially a Convolutional Neural Network (CNN), demands a large amount of CPU power. Three channels, or RGB (Red, Green, and Blue), make up an image and produce three-dimensional data. However, when reading these images into the network, processing limitations frequently require that they be converted into single-channel representations. Within the neural network architecture, this dimensionality reduction promotes effective processing and aids in the management of computational complexity.

The CNN is fed preprocessed input as shown in Figures 5 and 6. There are several layers in CNN. The list of them is as follows:



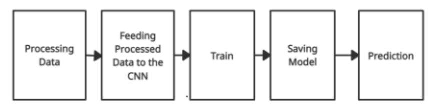


Fig. 5. Procedures for theMelanoma skin cancer detection system .

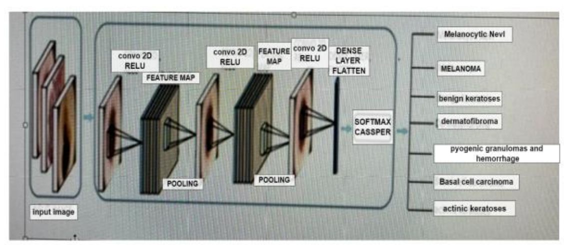


Fig. 6. Architecture of Convolution Neural Networks

1) Convolution Layer

By placing the filter window around each pixel in this layer and extracting the pixels from the provided image, the patch is analysed to see if the image lesion is skin cancer or not. To determine the category of each pixel, filters are moved around the picture and calculations are made. In our proposed method, an initial 28x28 pixel RGB enlargement of the supplied image is performed., and after that, a 3x3 pixel filter is applied. We try to relocate the filter to every pixel in the image before multiplying each pixel value by the matching filter pixel value and all of the other pixels' pixel values. After all, values have been added together, the result is recorded in the centre pixel by multiplying it by how many pixels there are in the outcome matrix. The source image will be padded if the patch pixels extend outside the picture's borders. Reflection in all four corner directions, as well as in all four corner directions, is necessary for padding (left, right, up, and down). The properties of the image are extracted by this layer [2].

2) Activation Function

The activation layer of our Convolutional Neural Network (CNN) model was designed with Rectified Linear Units (ReLu). This layer keeps values from building up to zero by eliminating negative values and substituting them with zeros. When the input value surpasses a predefined threshold, the ReLu activation function activates the node; otherwise, the pixel is set to zero.

3) Pooling Layer

The output of the activation layer is sent into the pooling layer, where we must choose between a 2x2 or 3x3 window size. Our model has a window that is 2x2. After that, drag each window onto the picture and add the window's highest value to the result matrix. As a result, the matrix's size can be reduced. The image can be further shrunk if needed, and then the convolution, activation, and pooling layers can be applied once again. This procedure can be repeated until the image is the desired size.

4) Dropout Layer

In order to randomly deactivate connections between hidden units (neurons within hidden layers), dropout is applied during the training phase. After each update, the outgoing edges of these units are set to zero. We use dropouts in our methods to help reduce overfitting, a frequent problem in machine learning when the model overfits to the training data and underperforms in terms of generalization.

5) Batch Normalization

To normalize the CNN model's inputs, batch normalization is applied to either the activations of the layer that came before it or the inputs themselves. Training can go more quickly if the number of epochs needed is reduced or increased as a result of this normalization process. Furthermore, by reducing generalization errors, batch normalization offers regularization and improves the model's overall performance.

6) Dense Layer

The Dense layer, sometimes referred to as the completely connected layer, receives the output from the Pooling layer. A fully connected network compares the positions of the input picture vector and the high-value elements in the training model in this layer. It computes the values of each outcome by dividing the sum of the input image by the values of the high-value element positions. It does this by aggregating the values of the input image and the high-value elements at corresponding positions, dividing the result by the sum of the high-value element vectors in the trained model, and so on. Then, the group with the best accuracy value is allocated to the provided image by referencing one of the seven forms of skin cancer [2].

7) Flatten layer

By reducing the multidimensional input tensors to a single dimension, this layer effectively feeds data into each and every neuron in the neural network model. In this layer, the matrix is transformed into a one-dimensional array. The picture has been reduced in size.

An optimizer can reduce losses by adjusting the weights and learning rate of a neural network. To handle sparse gradients on noisy problems, we used the Adam optimizer, an approach to replacement optimisation using stochastic gradient descent[2], in our deep learning models. We chose a learning rate of 0.001, which is a floating-point number and a tensor composed of it. When using Keras optimizer schedules, the learning rate is set to 0.001 by default. The Learning Rate Schedule is a callable that takes no parameters and produces the actual number to use. [2].

C. Dataset and Experimental Environment

The dataset used in this study is the HAM10000 dataset, which is publicly available on Kaggle at the following link: https://www.kaggle.com/datasets/kmader/skin-cancer-mnist-ham10000/data.

Recognizing pigmented skin lesions automatically using neural networks poses a challenge due to the scarcity of publicly accessible dermatoscopic image datasets, compounded by the small size of these images. To address this challenge effectively and accurately identify skin lesion images, particularly those indicative of melanoma, the study leverages the HAM10000 dataset, which comprises 10,015 images of skin lesions.

The model developed for this task utilizes deep learning techniques implemented in Python programming language. Various Python libraries were employed, including Numpy, OS, Tensorflow, Keras, and pandas, to facilitate data manipulation, model construction, and evaluation. The development environment for writing the Python code was Jupyter Notebook, an open-source web-based interactive application widely used for data analysis and machine learning tasks.

Following the model's construction and training, it was deployed using the Flask Python framework to enable the display of results on a website. Flask is a lightweight and extensible web framework for Python, commonly used for developing web applications. By deploying the model with Flask, users can interact with the trained model through a user-friendly web interface, facilitating convenient access and utilization of the melanoma detection system.

V. Results and Discussion

In our quest for maximizing accuracy in our model, we conducted a thorough evaluation of different types of optimizers. Optimizers play a crucial role in adjusting the parameters of a neural network during training to minimize the loss function and improve performance. Among the various optimizers tested, Adam consistently exhibited the highest accuracy across multiple training iterations. As a result, we made the strategic decision to adopt the Adam optimizer for our model.

To illustrate the comparative performance of different optimizers, we have provided a table below. This table highlights the accuracy achieved by each optimizer across various training iterations. By visually comparing the performance metrics of different optimizers, we were able to discern Adam's superiority and confidently select it as the optimizer for our model. This meticulous analysis ensures that our model is equipped with the most effective optimization strategy, ultimately leading to improved accuracy and performance in melanoma detection.

TABLE I. COMPARISON TABLE FOR DIFFERENT OPTIMIZERS.

Optimizers	No of epochs	Accuracy at initial epoch	Final Accuracy (final epoch)
RMSProp	51	0.6362	0.9972
Nadam	51	0.5797	0.9987
SGD	51	0.2895	0.9454
Agagrad	51	0.3271	0.9479
Adadelta	51	0.1497	0.4892
Adam	51	0.9919	0.9983

Table I presents a comparative analysis of various optimizers utilized within a machine learning model, alongside pertinent metrics including the number of epochs, initial epoch accuracy, and final epoch accuracy.

Each row in the "Optimizers" section corresponds to a distinct optimizer employed during model training, encompassing RMSProp, Nadam, SGD, Adagrad, Adadelta, and Adam.

The "Number of Epochs" column denotes the total number of training iterations completed for each optimizer. In this scenario, all optimizers underwent training for 51 epochs.

"Accuracy at Initial Epoch" signifies the model's accuracy at the onset of training, providing insights into its performance at the initial stage.

The "Final Accuracy" column reflects the model's accuracy upon completion of training, specifically during the final epoch. This metric offers a comprehensive assessment of the model's performance following the designated training period.Based on the results as shown in Table 5.1,

In our study, we conducted a detailed analysis of various optimizers within the context of our machine learning model. RMSProp, one of the optimizers evaluated, demonstrated noteworthy performance, starting with an initial accuracy of 0.6362 and achieving a final accuracy of 0.9972. Similarly, Nadam exhibited competitive results, commencing with an initial accuracy of 0.5797 and reaching a final accuracy of 0.9987. In contrast, SGD displayed a lower initial accuracy of 0.2895, which gradually improved to a final accuracy of 0.9454. Agagrad, starting with an initial accuracy of 0.3271, steadily progressed to a final accuracy of 0.9479, showcasing consistent improvement throughout training.

Adadelta, despite its lowest initial accuracy of 0.1497, demonstrated limited improvement, concluding with a final accuracy of 0.4892. However, it is noteworthy that the Adam optimizer showcased exceptional performance, boasting the highest initial accuracy of 0.9919 and maintaining strong performance throughout training, culminating in a final accuracy of 0.9983.

The outcomes of our study emphasize the pivotal role of optimizers in augmenting the accuracy of the model. Notably, the Adam optimizer emerged as the most effective performer in this comparative analysis, showcasing its exceptional capability in optimizing the model's performance.

In essence, the table presents a comprehensive comparison of various optimizers, delineating their respective impacts on enhancing the model's accuracy during the training phase. Through this comparative analysis, it becomes evident that the Adam optimizer consistently yielded promising results, underscoring its efficacy in fine-tuning the model parameters and improving predictive accuracy.

In summary, the tabulated data serves as a valuable tool for assessing and comparing the performance of different optimizers. It provides insights into their effectiveness in optimizing the model's accuracy, with the Adam optimizer standing out as a particularly potent choice in this regard. This finding underscores the importance of optimizer selection in achieving optimal model performance and underscores the potential of Adam optimizer in enhancing the efficacy of machine learning models.

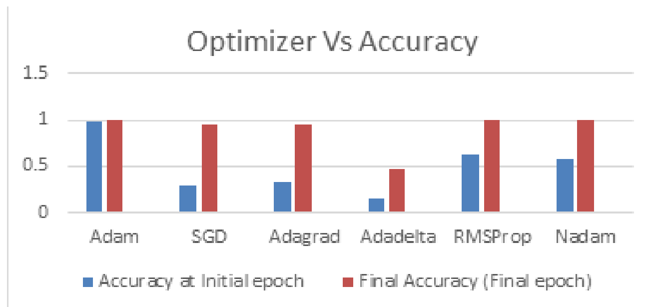


Fig. 7. Optimizer vs Accuracy

Epochs:

In order to verify for more accuracy, we investigated several epoch counts for the Adam optimizer. There was no discernible difference in the loss when we saw the same accuracy after about 40 training epochs. We'll have to stop iterating at 51 as a result. A comparison chart for various epoch counts is shown below in table II.

TABLE II. COMPARISON CHART FOR VARIOUS EPOCH COUNTS

Epochs No	accuracy at the start of time	Final precision
35	0.5858	0.9915
50	0.9909	0.9973
70	0.9966	0.9983

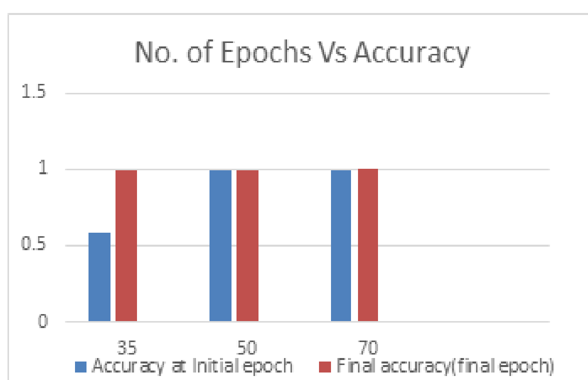


Fig. 8. Epoch count versus accuracy

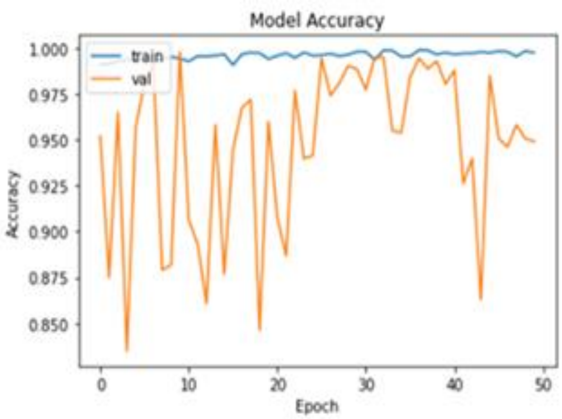


Fig. 9. Accuracy vs. Epoch

This section evaluates accuracy, precision, recall, specificity, and the f1 score.

Figure 9 reflects accuracy and the epoch. Accuracy increases when the epoch is pushed forward.

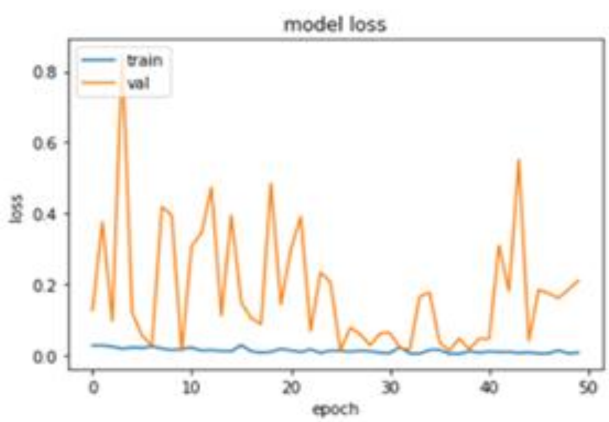


Fig. 10. Figure 5.4: Loss vs. Epoch.

The loss is shown as a function of epoch in Figure 10. As the epoch becomes bigger, the loss gets smaller. Additionally, the website helps consumers and experts by accurately identifying ailments in the most straightforward manner.

Fig. 11. Home page of Melanoma skin cancer detection website

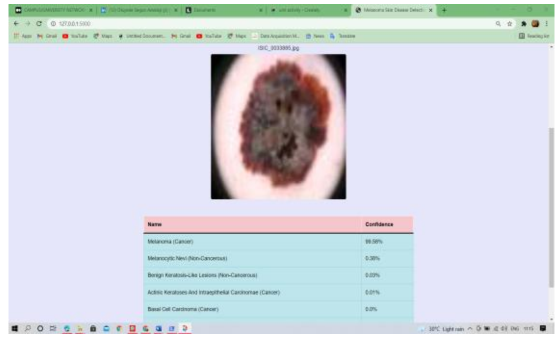


Fig. 12. Result of Melanoma skin cancer detection skin lesion image

VI. Conclusion and Future Scope

A. Conclusion

The conclusion drawn from this study underscores the critical importance of early detection in mitigating the risks associated with melanoma, the deadliest form of skin cancer. Traditional diagnostic methods often fall short in terms of speed and accuracy, necessitating the integration of advanced imaging techniques to enhance diagnostic precision. In response to this need, medical experts have devised various methodologies aimed at facilitating the early detection of melanoma.

This research presents a novel approach to skin cancer detection, centering on the utilization of Convolutional Neural Networks (CNN) as a fundamental component for accurately categorizing skin lesions into benign, atypical, or malignant classes. Through rigorous optimization employing a spectrum of optimizers such as SGD, RMSprop, Adam, Adagrad, Adadelta, and Nadam, the efficacy of the proposed CNN model is evaluated.

The evaluation results reveal that when paired with the Adam optimizer, the CNN model surpasses alternative optimization techniques in effectively categorizing skin cancer lesion datasets. Furthermore, the model yields comprehensive results, providing insights into seven distinct types of skin cancer alongside their corresponding probabilities expressed as percentages. With training accuracy reaching an impressive 99.73% and testing accuracy standing at 96.53%, the proposed model demonstrates superior performance compared to existing techniques documented in the technical literature.

The findings of this research suggest promising potential for the proposed model as a valuable tool in the realm of early-stage skin cancer diagnosis. Its utilization can empower users to promptly seek medical advice, thus potentially improving patient outcomes and reducing the mortality rate associated with melanoma. Overall, this study contributes to the ongoing efforts aimed at leveraging advanced technologies to enhance the accuracy and efficiency of skin cancer detection methodologies.

B. Future Scope

Improving the accuracy of both training and testing processes, coupled with the integration of advanced technology into a dedicated smartphone application, has the potential to enhance the reliability and utility of skin cancer identification methods.

References

[1] J. Daghrir, L. Tlig, M. Bouchouicha, and M. Sayadi, "Melanoma skin cancer detection using deep learning and classical machine learning techniques: A hybrid approach," 2020 5th International Conference on Advanced Technologies for Signal and Image Processing (ATSIP), 2020, pp. 1-5, doi: 10.1109/ATSIP49331.2020.9231544.

[2] S. M. Patil, B. S. Rajguru, R. S. Mahadik and O. P. Pawar, "Melanoma Skin Cancer Disease Detection Using Convolutional Neural Network," 2022 3rd International Conference for Emerging Technology (INCET), 2022, pp. 1-5, doi: 10.1109/INCET54531.2022.9825381.

[3] Kumar, Manoj & Alshehri, Mohammed & AlGhamdi, Rayed & Sharma, Purushottam & Deep, Vikas. (2020). A DE-ANN Inspired Skin Cancer Detection Approach Using Fuzzy C-Means Clustering. Mobile Networks and Applications. 25. 10.1007/s11036-020-01550-2.

[4] A. Naeem, M. S. Farooq, A. Khelifi and A. Abid, "Malignant Melanoma Classification Using Deep Learning: Datasets, Performance Measurements, Challenges and Opportunities," in IEEE Access, vol. 8, pp. 110575-110597, 2020, doi: 10.1109/ACCESS.2020.3001507.

[5] Lokesh Singh, Rekh Ram Janghel, Satya Prakash Sahu,Designing a Retrieval-Based Diagnostic Aid using Effective Features to Classify Skin Lesion in Dermoscopic Images,Procedia Computer Science,Volume 167,2020,Pages 2172-2180,ISSN 1877-0509, https://doi.org/10.1016/j.procs.2020.03.267.

[6] Patil S.M., Malik A.K. (2019) Correlation Based Real-Time Data Analysis of Graduate Students Behaviour. In: Santosh K., Hegadi R. (eds) Recent Trends in Image Processing and Pattern Recognition. RTIP2R 2018. Communications in Computer and Information Science, vol 1037. Springer, Singapore. https://doi.org/10.1007/978-981-13-9187-3_62.

[7] Yashodhan Ketkar, Sushopti Gawade.- "Detection of arrhythmia using weightage-based supervised learning system for COVID-19". https://doi.org/10.1016/j.iswa.2022.200119 .

[8] Shankar M. Patil and Dr. Praveen Kumar, "Data Mining Model For Effective Data Analysis Of Higher Education Students Using MapReduce", IJERMT, 2017.

[9] Chopade and Gawade- "Multimedia Teaching Learning Methodology and Result Prediction System Using Machine Learning" Journal of Engineering Education Transformations, Volume No 35, January 2022, Special issue, eISSN 2394-1707

[10] Thanh, D., Prasath, V.B.S., Hieu, L. et al. Melanoma Skin Cancer Detection Method Based on Adaptive Principal Curvature, Colour Normalisation and Feature Extraction with the ABCD Rule. J Digit Imaging 33, 574–585 (2020). https://doi.org/10.1007/s10278-019-00316-x

[11] Marathe, Gawade, & Kanekar- "Prediction of heart disease and diabetes using naive Bayes algorithm ".International Journal of Scientific Research in Computer Science, Engineering and Information Technology Volume 7, pp. 447-453 Issue 3 Accepted : 25 May 2021, Published : 31 May 2021, doi : https://doi.org/10.32628/CSEIT217399

[12] Verma & Gawade- "A machine learning approach for prediction system and analysis of nutrients uptake for better crop growth in the hydroponics system".2021 international conference on artificial intelligence and smart systems (ICAIS), IEEE (2021), pp. 150-156, doi: 10.1109/ICAIS50930.2021.9395956.

[13] Yashodhan Ketkar, Sushopti Gawade.- "Detection of arrhythmia using weightage-based supervised learning system for COVID-19".https://doi.org/10.1016/j.health.2022.100117 Healthcare Analytics Volume 2, November 2022, 100117

[14] Sushopti Gawade, Ashok Bhansali, Kshitij Patil, Danish Shaikh. "Application of the convolutional neural networks and supervised deep-learning methods for osteosarcoma bone cancer detection", Healthcare Analytics, 2023.

[15] S. Satre, Shankar M. Patil, T. Mane, V. Molawade, T. Gawand and A. Mishra, "Online Exam Proctoring System Based on Artificial Intelligence," 2023 International Conference on Signal Processing, Computation, Electronics, Power and Telecommunication (IConSCEPT), Karaikal, India, 2023, pp. 1-6, doi: 10.1109/IConSCEPT57958.2023.10170577.

[16] Shankar M. Patil, Heena Shaikh, Yashshree Wagh, Siddhesh Shinde, "Classification of Affected Fruits using Machine Learning", International Journal of Creative Research Thoughts (IJCRT), Volume 8, Issue 6 June 2020, ISSN: 2320-2882.

[17] Uma Sharma, Sidharth Saran, Shankar M. Patil, "Fake News Detection Using Machine Learning Algorithms", International Journal of Creative Research Thoughts (IJCRT), Volume 8, Issue 6 June 2020, ISSN: 2320-2882.

[18] Temidayo Oluwatosin Omotehinwa, David Opeoluwa Oyewola, Emmanuel Gbenga Dada "A Light Gradient-Boosting Machine algorithm with Tree-Structured Parzen Estimator for breast cancer diagnosis", Healthcare Analytics, 2023

[19] Sandeep Wadekar, Dileep Kumar Singh," A modified convolutional neural network framework for categorizing lung cell histopathological image based on residual network" , Healthcare Analytics, 2023

[20] B. Jaishankar, Ashwini A.M., Vidyabharathi D., L. Raja,"A novel epilepsy seizure prediction model using deep learning and classification,Healthcare Analytics, 2023