iJRASET
International Journal For Research in
Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH
IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 13 Issue: IX Month of publication: September 2025

DOI: https://doi.org/10.22214/ijraset.2025.74054

# Research Report on Melanoma Skin Cancer Detection Using CNN Algorithm and Image Processing

P.S Suma[1], Mrs. SharvaniV[2]
*Master of Computer Application Ballari Institute of Technology And Management, Ballari, Karnataka*

***Abstract: Skin cancer is a serious problem that is frequently ignored. When a clinician performs a manual examination, Using imaging data, the human eye might not always be able to correctly identify ailments. In today's environment, Multi layered networks are being employed more and more to solve difficulties in our everyday lives. Therefore, we develop an automated computerised system for detecting skin problems using deep neural network approaches. We have incorporated several neural network algorithms into the suggested model and evaluated their results to ascertain the most precise method for recognising the five main skin conditions.CNN, and by using the Keras Sequential API, we have created a new model that achieves an accuracy of roughly 85%. We have since used approaches that make use of pre-trained data to increase accuracy and facilitate comparability. These include the transfer learning models VGG16, RESNET50, and Densenet121. From the set of algorithms utilized in the suggested models, the resnet architecture obtains the highest accuracy of 90%.***

***Keywords: pretrained data, deep learning, neural networks, and machine learning, and transfer learning***

## I. INTRODUCTION

People with although skin disorders are widespread, they are frequently disregarded because they are assumed to be ordinary skin issue. However, many are unaware of how hazardous this could become if they fail to take the necessaryprotects and acknowledges the exact condition they are experiencing, such as seeing the right doctor and getting the right prescription. We are aware of how difficult it is for residents of rural parts of poor nations like Bangladesh, Sri Lanka, India, etc. to find doctors, much less ones who specialize in skin care.Accuracy is lacking even in human eyes, and doctors are no exception. Our target is to alleviate a problem by applying our understanding between neural networks including current technological advancements in the fields of medical science. Our goal is to create a system.to assist patients with skin cancer using the information gathered from the ISIC archive and our institution's expertise. Oxford Journals estimates that between 1.3 and 1.5 million people in Bangladesh suffer from various illnesses, with about 0.2 million of those cases being recently diagnosed with cancer annually.Therefore, it's either too late when they understand it's time to see a qualified doctor or they decide to prevent getting any skin diseases. Additionally, doctors in those locations are typically not knowledgeable about skin conditions that might lead to cancer. The fatality rate from cancer is still very high, even in a country like the United States, which has the best hospitals with the best specialists.If they had been able to take more preventive measures sooner, more people would have recognised it as a potentially deadly condition rather than just a typical skin blemish. That's where our concept for image-only skin cancer detection comes into play [24]. The goal of this paper is to create an image processing system that automatically detects melanoma from medical imaging of skin abnormalities processed with CNN-based models, a cutting-edge deep learning approach. Because CNNs can efficiently capture intricate patterns and characteristics required for melanoma diagnosis, they demonstrate strong capability in handling image classification problems.

Numerous articles that we read aided in the study for our thesis. During difficult times, they gave us guidance, inspiration, and support. We looked into several methods that try to solve the same problem by combining machine learning and neural networks, since the objective of our work is that to detect skin cancer using different neural networks. One of the pioneering studies in this area was "Automatic Melanoma Skin Cancer Identification via Texture Analysis" by Mariam A. Sheha of Cairo University. Finding melanoma, a potentially lethal skin cancer, is the aim of this study. In 2011, over 9,000 people in the USA lost their lives to melanoma, while nearly 70,000 people were diagnosed with the condition [25].In order to distinguish between melanocytic nevi and malignant melanoma, they have employed texture analysis. [19].Many of the proposed techniques need a division process that is thought to be fatal.

because of the peculiarities of the tumor. Histological tissue dermoscopypictures display characteristics that are mostly organized in a number of examples. Consequently, the regulated division of many structures, including vessels, cytoplasm, cores, and so on, is difficult and impossible.[9][1][14] [4] [16]. Artificial neural networks (ANNs) in PC-supported dermoscopy have recently been demonstrated to be a reliable method for assessing pigmented skin sores (PSLs).[12][20] [21].They used an artificial neural network called MLP that represented the classifications of the two illnesses, whereas we used neural networks like DenseNet, ResNet, and VGG models to classify seven distinct types of skin lesions.

Both automated and conventional MLP kinds have been used. To support their targets, the authors examined earlier literature and incorporated prior findings. Various techniques have been presented in the past to strengthen the reliability and accuracy of diagnosis. Oil immersion, incident light, and magnification were all used in the 1987 description of ELM (Epiluminescence Microscopy) [22].

## II. RELATED WORK

Dermoscopy and computerized image analysis are two techniques that result from the development of non-intrusive technologies to improve early conclusion [21].Generally speaking, reference images are included in pictorial chart books [23]. To distinguish between melanoma and melanocytic nevi, co-occurrence matrix texture characteristics are employed for classification. Nonlinear boundaries are generated using feed-forward networks called Multilayer Perceptrons (MLPs) [6].[15] [17]. Twenty percent is utilized for testing, twenty percent is used for validation, and sixty percent is used for teaching. Unlike our approach, which allocates the dataset into 80% for training and 20% for testing, and another 20% validation.How to differentiate melanoma from three other malignancies was covered by the study's authors:Usingcolored images of skin lesions, image processing techniques, and an artificial neural network (ANN), seborrheic keratoses, dysplastic nevi, and intradermal nevi were identified. To do this, An ANN that includes feedforward was employed. which has 14 inputs and 1 output that indicates if the tumor is melanoma.This model was trained by using the backpropagation rule. They also claimed to have used NeuralWorks, a commercial neural network application. Neural networks have gained significant application in the past few years as pattern classifiers in restorative analysis [8], [7], pattern recognition [2], and computerized reasoning applications [3].

A selection of highlights is necessary for indicative applications, and Every issue space needs to be customised in a different way. Enough information should be provided by the chosen features to identify classes while being indifferent to trivial discrepancies in the sources of the information.

They identified 14 key features, which we concur are essential for distinguishing between different images and dangerous melanoma. In the shading image, it was discovered that the great majority of the 14 highlighted places were helpful in locating the tumor'sedge.Golston et al. [5] and Ercal et al. [10] both offer thorough descriptions of limit recognition.Several studies have investigated machine learning and image processing approaches for melanoma detection, with image processing methods often focusing on texture-based features, colour histogram analysis, and edge identification were employed in the early systems. However, the complex features of melanoma lesions were not sufficiently captured by these methods.

A recent advancement in Deep learning, particularly Convolutional Neural Networks (CNNs), has demonstrated significant potential in image classification tasks, including medical image analysis. By automatically learning hierarchical feature representations directly from raw pixel data, CNNs offer capabilities that surpass traditional image processing methods. Multiple studies have indicated that CNN-based models can achieve melanoma detection performance comparable to, or even exceeding, that of experienced dermatologists, including those by Esteva et al. (2017) and Tschandl et al. (2018).

## III. METHODOLOGY

We started by running our own CNN model on the dataset (HAM10000) that we obtained from the ISIC repository, as seen in Figure 1. We then used VGG, ResNet, and DensNet techniques to provide more accurate and superior results. These provide quicker development or superior results when we replicate the second task to improve our accuracy.

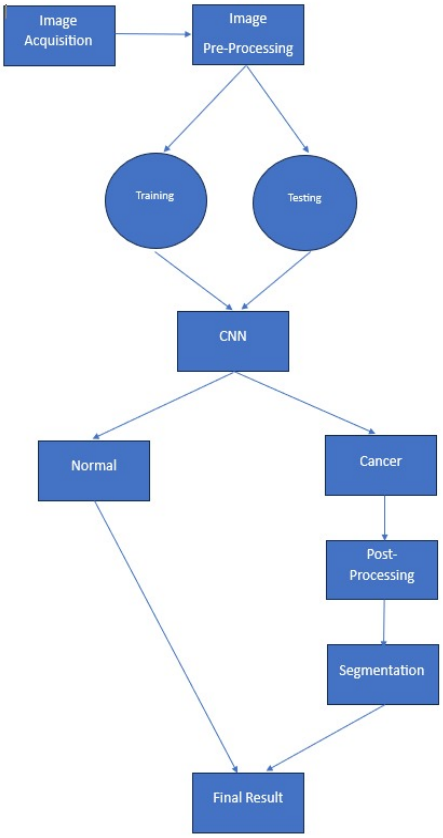
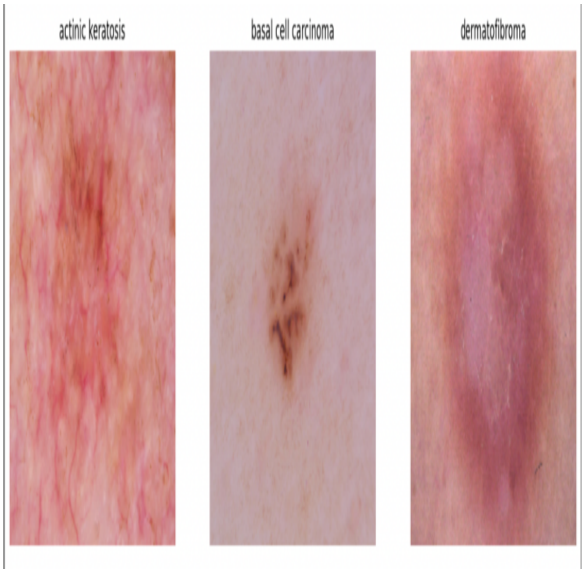


Figure 1: Our Proposed Model

### *A. Dataset:*

The ISIC (International Skin Imaging Collaboration) Archive, which includes thousands of annotated dermoscopic pictures of lesions of the skin, served as the study's main dataset. among the most prevalent types of benign (non-cancerous) and malignant (cancerous) lesions that are classified in these photos. There are differences in lesion color, shape, size, and texture throughout the extremely varied dataset.



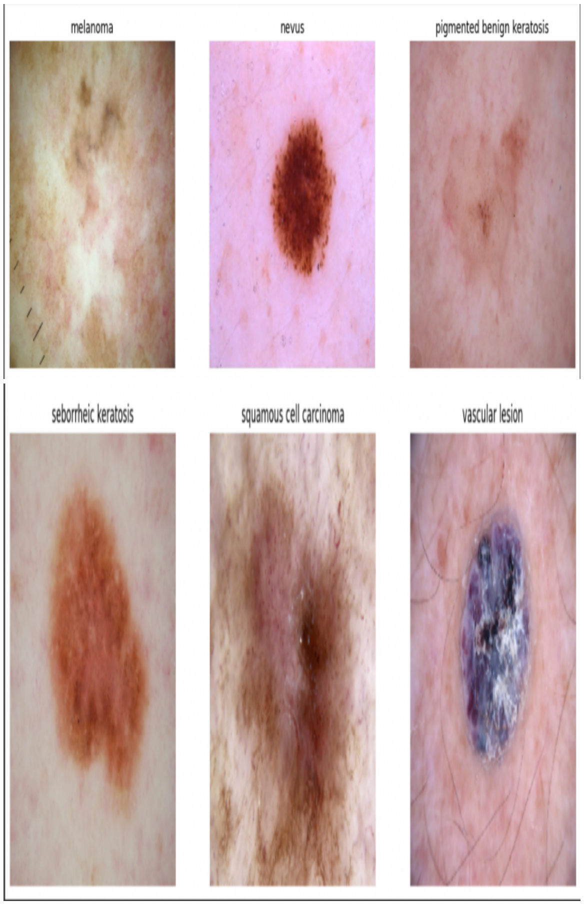


Figure 2: Dataset images

### B. *Preprocessing*

Preprocessing is crucial for improving the correctness of the CNN model by enhancing the input photos' quality and extracting relevant features. The preprocessing The methods outlined below were implemented on thedermoscopic images:

1) Image loading and resizing: At this step, images will be loaded from the picture folder into the image path and placed in the image column. Additionally, we scale the images to 100 x 75 x 3 because TensorFlow cannot handle their original size of 450 × 600 x 3.
2) Split Test-Train: The dataset was split into training and testing sets using an 80:20 ratio.
3) Normalization: The train and test sets were normalised by deducting their mean values and dividing the result by the standard deviation. Any numbers between 0 and 255 are normalised to 0.
4) EncodingLabels:Skin cancer can be divided into seven categories. with numbers 0 through 6. These include dermatofibroma, melanoma, melanocytotic nevi, benign keratosis-like lesions, basal cell carcinoma, vascular lesions, actinic keratoses, and melanoma.
5) Dividing training From validation:
   90% of the data was used to train our model. We tested our model on 10 percent of the data. To reduce overfitting, a short test or validation dataset is utilised in this case.
6) Data Augmentation
   The sample photos were zoomed in by 10% and rotated by 10 degrees for our model. The photos were additionally rotated by 10% in both the horizontal and vertical directions to expand the size of our training sample even more.

### *C. System Flow Diagram*

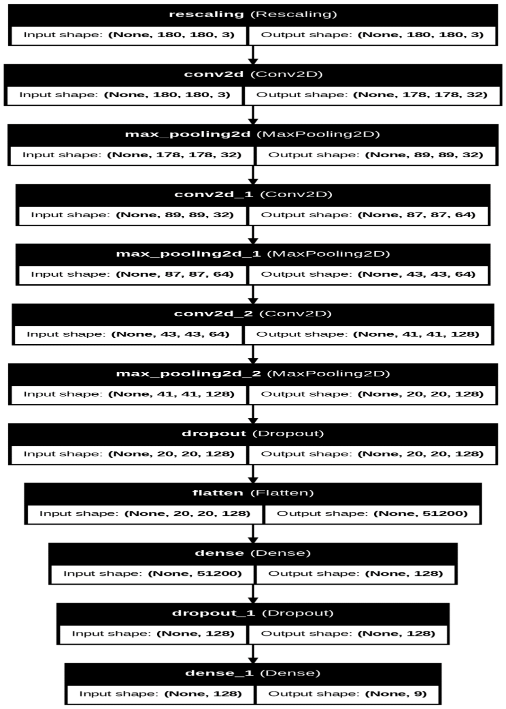

rescaling (Rescaling)
Input shape: (None, 180, 180, 3)
Output shape: (None, 180, 180, 3)
conv2d (Conv2D)
Input shape: (None, 180, 180, 3)
Output shape: (None, 178, 178, 32)
max_pooling2d (MaxPooling2D)
Input shape: (None, 178, 178, 32)
Output shape: (None, 89, 89, 32)
conv2d_1 (Conv2D)
Input shape: (None, 89, 89, 32)
Output shape: (None, 87, 87, 64)
max_pooling2d_1 (MaxPooling2D)
Input shape: (None, 87, 87, 64)
Output shape: (None, 43, 43, 64)
conv2d_2 (Conv2D)
Input shape: (None, 43, 43, 64)
Output shape: (None, 41, 41, 128)
max_pooling2d_2 (MaxPooling2D)
Input shape: (None, 41, 41, 128)
Output shape: (None, 20, 20, 128)
dropout (Dropout)
Input shape: (None, 20, 20, 128)
Output shape: (None, 20, 20, 128)
flatten (Flatten)
Input shape: (None, 20, 20, 128)
Output shape: (None, 51200)
dense (Dense)
Input shape: (None, 51200)
Output shape: (None, 128)
dropout_1 (Dropout)
Input shape: (None, 128)
Output shape: (None, 128)
dense_1 (Dense)
Input shape: (None, 128)
Output shape: (None, 9)

Figure 2: System Flow Diagram

## IV. MODEL IMPLEMENTATION

*VGG16 Model*

### *1) Convolutional Layers*

In order to preserve spatial resolution, the VGG16 model employs tiny convolutional filtering using a step of 1 and padding across the network. These filters are 3×3 times 3×3. While keeping a receptive field on par with bigger filters, this design decision lowers the number of parameters.

For a single convolution operation:

$$Z_{i,j,k} = \sum_{m=1}^{M}\sum_{n=1}^{N}\sum_{c=1}^{C} X_{i+m-1,j+n-1,c} \cdot W_{m,n,c,k} + b_k$$

- Z  i, j, k: Activation of the output at location ( $i, j$ ) in the  $k$-th filter.
- $Xi, j, c$: Input activation at position  ( $i, j$ ) in the c-th channel.
- $Wm, n, c, k$: Weight of the filter at position  ( $m, n$ ) in the  $k$-th filter.
- $bk$: Bias for the  -th filter.
- $M, N$: Dimensions of the filter (here 3×3).
- $C$: Number of input channels.

### *2) Max-Pooling Layers*

The max-pooling layers lower spatial dimensions while maintaining important characteristics by down sampling the feature maps. Stride 2 and their filter size are 2×2 times 2×2.

For max-pooling:

$$Z_{i,j,k} = \max_{p,q \in P} X_{p,q,k}$$

P: The set of positions in the pooling window.

### *3) Fully Connected Layers*

The final three fully linked layers of the VGG16 model are where the features are flattened andvectorized. A completely linked layer's output is:

- N: Number of input features.
- σ\sigmaσ: Activation function (e.g., ReLU for intermediate layers, softmax for the output layer).

For the models, we used the ADAM optimizer. The aforementioned approach extends stochastic gradient descent(SGD), which was used in a more general application of pertaining to deep learning applications within computer vision and natural learning processing. The parameters are much enhanced and the percentage loss is reduced by continuously reviewing the kernel values, weights, and bias of neurons. After the layers were linked, a number of functions were developed, including a scoring function, a loss function, and finally a tuning technique. This represents rate of inaccuracy between the perceived label and the expected mistake. A specific type of categorical cross entropy classification is used, which iteratively refines parameters to decrease loss, with the optimizer acting as the most importantfunction.

ResNet models employ batch normalizations in between double or even triple layer skips that contain non-linearities (ReLu). To prevent the vanishing gradient issue, some layers are skipped, which is facilitated by residual connectionsreusing the activations from a previous layer until the nearby layers know its weight. Jumping over levels hence simplifies the network and lowers the quantity of training layers. 224 x 224 x 3 is the network's input picture size. The graphic shows the ResNet algorithm.

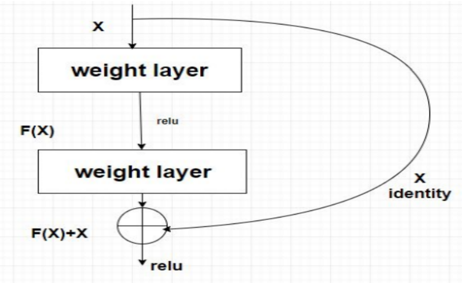


Figure 3: ResNet50 Algorithm

ResNets allow for the removal of any layer at any moment, and not all of them are required. A huge ResNet has learned weights for each layer. To counter this, DenseNets use thin layers together with smaller sets of new feature maps. Additionally, the real input image  and the loss function gradients are accessible to every DenseNets layer, which helps to address gradient and information flow problems during the training stage. In DenseNets, the 36 received feature maps are not combined with the output feature of the layer; instead, they are concatenated. DenseNets are structured by partitioning the feature maps into DenseBlocks, with each DenseBlock using a different number of filters, but the size and shape of the feature maps remain the same. Two 1x1 convolution, two 2x2 pooling, and batch normalization are used by the intermediate layers, often referred to as Transition Layers, to control downsampling.

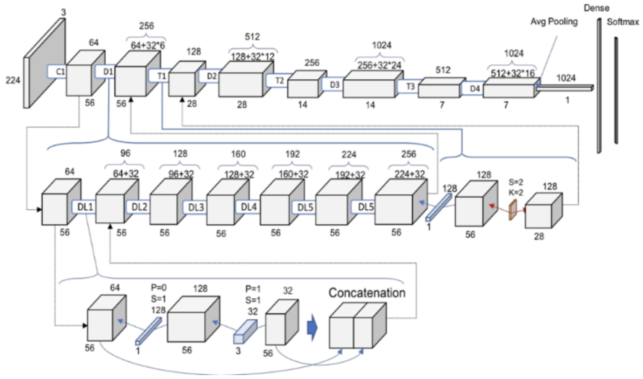


Figure 4: Full architecture depiction of DenseNet121

The volume 32 extracted feature maps in Figure 4 are expanded through successive layers. As a result, it rises from 64 to 256 after six full layers. Following a 1x1 convolution applying 128 filters, the Transition Block does a 2x2 subsampling operation with two strides. Consequently, volume and feature maps are later decreased to 50% for every Transition block. We may then do a higher cost 3x3 convolution employing the 32 feature maps of growth rate that were chosen. After then, the two processes' incoming and outgoing data sizesare merged. This adds new information to the collective knowledge of the network.

Next, we have used VGG16 in our proposed model.The error rate for VGG16 is already 10.4%. VGG uses pre-trained data to increase accuracy. VGG11 has 133 million parameters, VGG16 contains 138 million, and VGG19 contains 144 million. VGG11 is the precursor of VGG16 and VGG19. The graphic shows the architecture of each VGG model. Surprisingly, we were unable to obtain the desired outcome with VGG16 and VGG19 after 10 epochs. Because the outcomes from both models were similar, we showed the last two epochs of the ten. Under-tting is the consequence of failing to maintain an appropriate degree High training exactness, along with substantial variability in validation accuracy, was observed across both epochs.

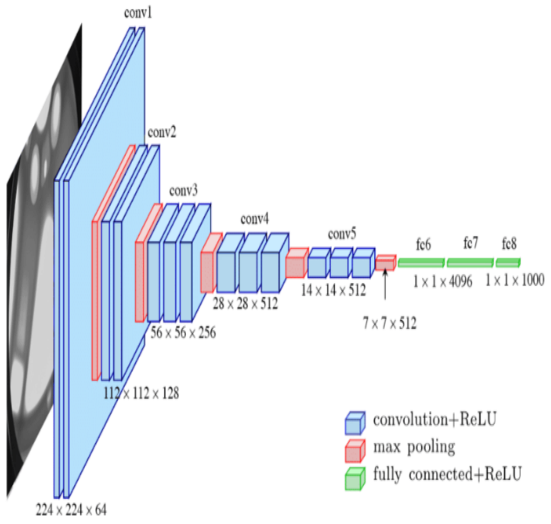


Figure 5: VGG Models Architecture

## V. POST PROCESSING

Enhancing the accuracy and clinical relevance of model predictions for melanoma skin cancer diagnosis requires deep learning post-processing. Post-processing approaches are used to optimize the final output and eliminate mistakes after a deep learning model, often a convolutional neural network (CNN), detects melanoma lesions.

One important technique is Confidence Thresholding, which involves establishing a minimal confidence threshold and modifying the model's predicted probabilities. This lowers false positives by guaranteeing that only extremely confident forecasts are taken into account.

In order to ensure more precise delineation of the damaged skin region, post-processing procedures such as morphological operations (e.g., dilation, erosion) smooth or refine the boundaries of melanoma lesions for segmentation tasks. Additionally, this can enhance lesion localization and lessen noise.

Using extra context, such as lesion shape, color, and symmetry, to confirm or disprove possible melanoma predictions is known as false positive reduction. By taking this precaution, benign illnesses won't be wrongly labeled as melanoma.

The outputs of many networks or runs of the same model are combined for a final judgment in ensemble Techniques, that merge accurate outputs from multiple models to increase accuracy.

Ending with visualization which is equally important methods such as Grad-CAM provide heatmaps that highlight the areas of the image that were crucial to the model's judgment, enhancing interpretability and helping physicians to trust and comprehend the model's results.

Deep learning-based melanoma detection systems are more dependable for clinical applications thanks to these post-processing procedures, which also serve to improve the model's predictions and lower mistakes.

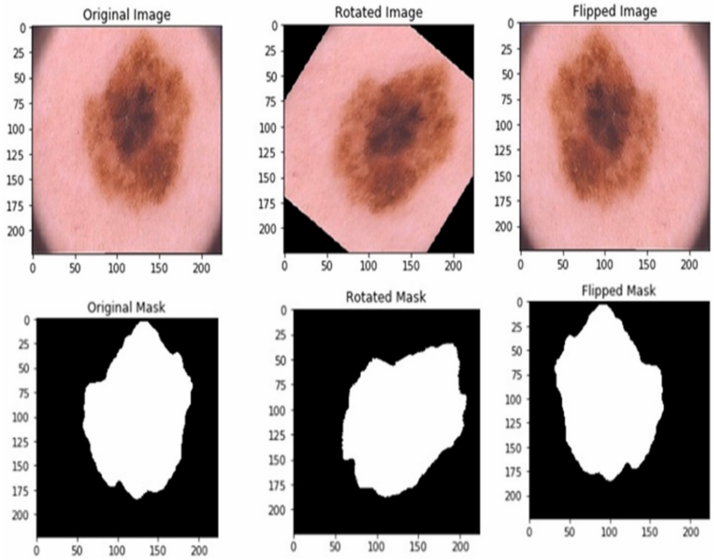
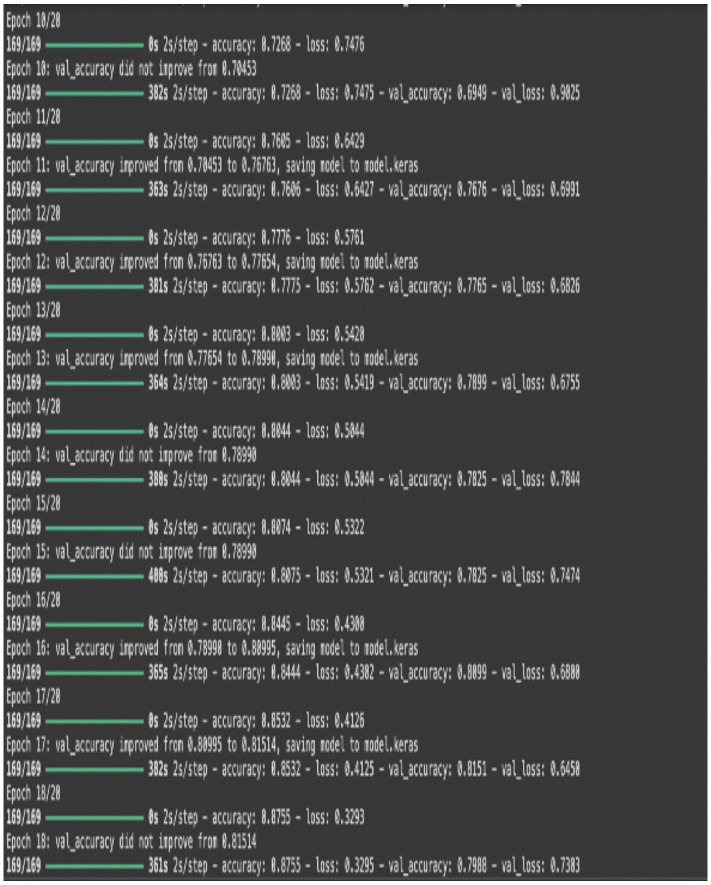


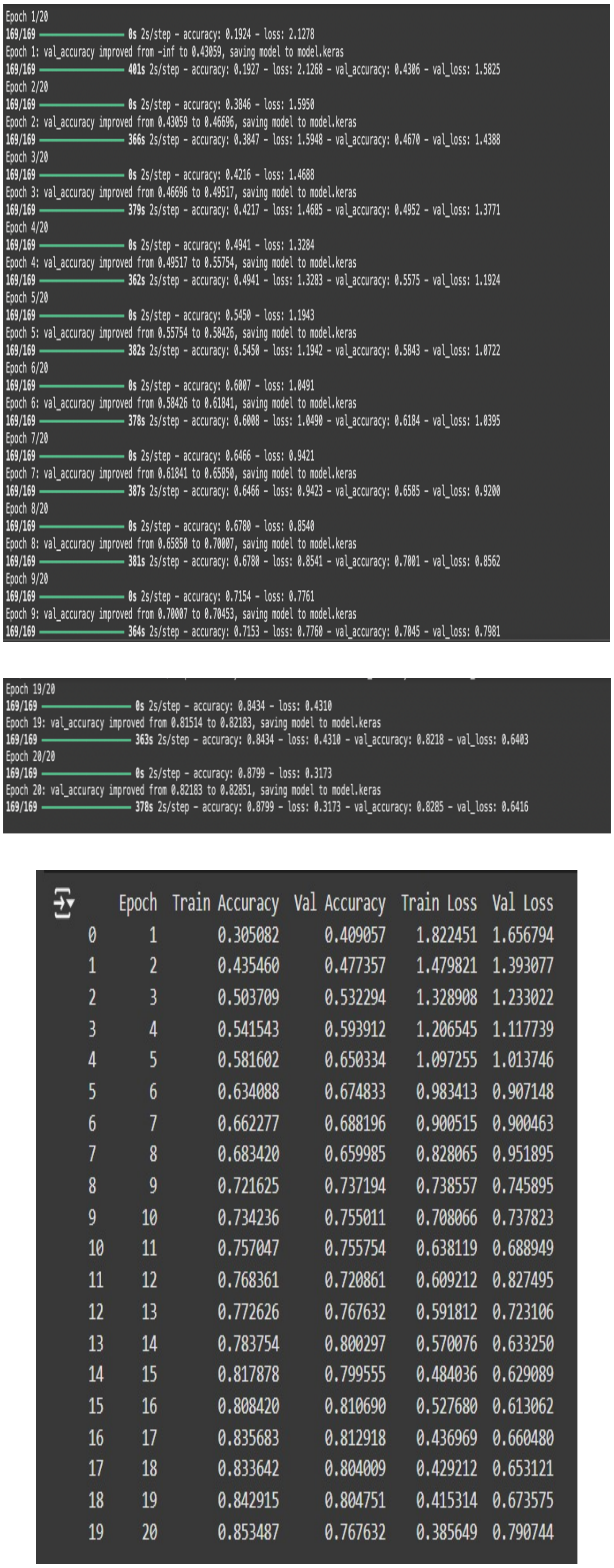
Figure 6: Segmentation Image

## VI. MODEL TRAINING AND EVALUATION

Eighty percent of the images were allocated for model training, with the remaining twenty percent reserved for validation. The training process was monitored using the cross-entropy loss function, while optimization was performed using Adam to maximize it. After a few epochs, the validation accuracy stopped increasing, and the training was terminated.



```
Epoch 10/20
169/169 ━━━━━━━━━━━━━━━━━━━━ 0s 2s/step - accuracy: 0.7268 - loss: 0.7476
Epoch 10: val_accuracy did not improve from 0.70453
169/169 ━━━━━━━━━━━━━━━━━━━━ 382s 2s/step - accuracy: 0.7268 - loss: 0.7475 - val_accuracy: 0.6949 - val_loss: 0.9025
Epoch 11/20
169/169 ━━━━━━━━━━━━━━━━━━━━ 0s 2s/step - accuracy: 0.7605 - loss: 0.6429
Epoch 11: val_accuracy improved from 0.70453 to 0.76763, saving model to model.keras
169/169 ━━━━━━━━━━━━━━━━━━━━ 363s 2s/step - accuracy: 0.7606 - loss: 0.6427 - val_accuracy: 0.7676 - val_loss: 0.6991
Epoch 12/20
169/169 ━━━━━━━━━━━━━━━━━━━━ 0s 2s/step - accuracy: 0.7776 - loss: 0.5761
Epoch 12: val_accuracy improved from 0.76763 to 0.77654, saving model to model.keras
169/169 ━━━━━━━━━━━━━━━━━━━━ 381s 2s/step - accuracy: 0.7775 - loss: 0.5762 - val_accuracy: 0.7765 - val_loss: 0.6826
Epoch 13/20
169/169 ━━━━━━━━━━━━━━━━━━━━ 0s 2s/step - accuracy: 0.8003 - loss: 0.5420
Epoch 13: val_accuracy improved from 0.77654 to 0.78990, saving model to model.keras
169/169 ━━━━━━━━━━━━━━━━━━━━ 364s 2s/step - accuracy: 0.8003 - loss: 0.5419 - val_accuracy: 0.7899 - val_loss: 0.6755
Epoch 14/20
169/169 ━━━━━━━━━━━━━━━━━━━━ 0s 2s/step - accuracy: 0.8044 - loss: 0.5044
Epoch 14: val_accuracy did not improve from 0.78990
169/169 ━━━━━━━━━━━━━━━━━━━━ 380s 2s/step - accuracy: 0.8044 - loss: 0.5044 - val_accuracy: 0.7825 - val_loss: 0.7844
Epoch 15/20
169/169 ━━━━━━━━━━━━━━━━━━━━ 0s 2s/step - accuracy: 0.8074 - loss: 0.5322
Epoch 15: val_accuracy did not improve from 0.78990
169/169 ━━━━━━━━━━━━━━━━━━━━ 400s 2s/step - accuracy: 0.8075 - loss: 0.5321 - val_accuracy: 0.7825 - val_loss: 0.7474
Epoch 16/20
169/169 ━━━━━━━━━━━━━━━━━━━━ 0s 2s/step - accuracy: 0.8445 - loss: 0.4300
Epoch 16: val_accuracy improved from 0.78990 to 0.80995, saving model to model.keras
169/169 ━━━━━━━━━━━━━━━━━━━━ 365s 2s/step - accuracy: 0.8444 - loss: 0.4302 - val_accuracy: 0.8099 - val_loss: 0.6800
Epoch 17/20
169/169 ━━━━━━━━━━━━━━━━━━━━ 0s 2s/step - accuracy: 0.8532 - loss: 0.4126
Epoch 17: val_accuracy improved from 0.80995 to 0.81514, saving model to model.keras
169/169 ━━━━━━━━━━━━━━━━━━━━ 382s 2s/step - accuracy: 0.8532 - loss: 0.4125 - val_accuracy: 0.8151 - val_loss: 0.6458
Epoch 18/20
169/169 ━━━━━━━━━━━━━━━━━━━━ 0s 2s/step - accuracy: 0.8755 - loss: 0.3293
Epoch 18: val_accuracy did not improve from 0.81514
169/169 ━━━━━━━━━━━━━━━━━━━━ 361s 2s/step - accuracy: 0.8755 - loss: 0.3295 - val_accuracy: 0.7988 - val_loss: 0.7303
```



```
Epoch 1/20
169/169 ━━━━━━━━━━━━━━━━━━━━ 0s 2s/step - accuracy: 0.1924 - loss: 2.1278
Epoch 1: val_accuracy improved from -inf to 0.43059, saving model to model.keras
169/169 ━━━━━━━━━━━━━━━━━━━━ 401s 2s/step - accuracy: 0.1927 - loss: 2.1268 - val_accuracy: 0.4306 - val_loss: 1.5825
Epoch 2/20
169/169 ━━━━━━━━━━━━━━━━━━━━ 0s 2s/step - accuracy: 0.3846 - loss: 1.5950
Epoch 2: val_accuracy improved from 0.43059 to 0.46696, saving model to model.keras
169/169 ━━━━━━━━━━━━━━━━━━━━ 366s 2s/step - accuracy: 0.3847 - loss: 1.5948 - val_accuracy: 0.4670 - val_loss: 1.4388
Epoch 3/20
169/169 ━━━━━━━━━━━━━━━━━━━━ 0s 2s/step - accuracy: 0.4216 - loss: 1.4688
Epoch 3: val_accuracy improved from 0.46696 to 0.49517, saving model to model.keras
169/169 ━━━━━━━━━━━━━━━━━━━━ 379s 2s/step - accuracy: 0.4217 - loss: 1.4685 - val_accuracy: 0.4952 - val_loss: 1.3771
Epoch 4/20
169/169 ━━━━━━━━━━━━━━━━━━━━ 0s 2s/step - accuracy: 0.4941 - loss: 1.3284
Epoch 4: val_accuracy improved from 0.49517 to 0.55754, saving model to model.keras
169/169 ━━━━━━━━━━━━━━━━━━━━ 362s 2s/step - accuracy: 0.4941 - loss: 1.3283 - val_accuracy: 0.5575 - val_loss: 1.1924
Epoch 5/20
169/169 ━━━━━━━━━━━━━━━━━━━━ 0s 2s/step - accuracy: 0.5450 - loss: 1.1943
Epoch 5: val_accuracy improved from 0.55754 to 0.58426, saving model to model.keras
169/169 ━━━━━━━━━━━━━━━━━━━━ 382s 2s/step - accuracy: 0.5450 - loss: 1.1942 - val_accuracy: 0.5843 - val_loss: 1.0722
Epoch 6/20
169/169 ━━━━━━━━━━━━━━━━━━━━ 0s 2s/step - accuracy: 0.6007 - loss: 1.0491
Epoch 6: val_accuracy improved from 0.58426 to 0.61841, saving model to model.keras
169/169 ━━━━━━━━━━━━━━━━━━━━ 378s 2s/step - accuracy: 0.6008 - loss: 1.0490 - val_accuracy: 0.6184 - val_loss: 1.0395
Epoch 7/20
169/169 ━━━━━━━━━━━━━━━━━━━━ 0s 2s/step - accuracy: 0.6466 - loss: 0.9421
Epoch 7: val_accuracy improved from 0.61841 to 0.65850, saving model to model.keras
169/169 ━━━━━━━━━━━━━━━━━━━━ 387s 2s/step - accuracy: 0.6466 - loss: 0.9423 - val_accuracy: 0.6585 - val_loss: 0.9200
Epoch 8/20
169/169 ━━━━━━━━━━━━━━━━━━━━ 0s 2s/step - accuracy: 0.6780 - loss: 0.8540
Epoch 8: val_accuracy improved from 0.65850 to 0.70007, saving model to model.keras
169/169 ━━━━━━━━━━━━━━━━━━━━ 381s 2s/step - accuracy: 0.6780 - loss: 0.8541 - val_accuracy: 0.7001 - val_loss: 0.8562
Epoch 9/20
169/169 ━━━━━━━━━━━━━━━━━━━━ 0s 2s/step - accuracy: 0.7154 - loss: 0.7761
Epoch 9: val_accuracy improved from 0.70007 to 0.70453, saving model to model.keras
169/169 ━━━━━━━━━━━━━━━━━━━━ 364s 2s/step - accuracy: 0.7153 - loss: 0.7760 - val_accuracy: 0.7045 - val_loss: 0.7981
```

```
Epoch 19/20
169/169 ━━━━━━━━━━━━━━━━━━━━ 0s 2s/step - accuracy: 0.8434 - loss: 0.4310
Epoch 19: val_accuracy improved from 0.81514 to 0.82183, saving model to model.keras
169/169 ━━━━━━━━━━━━━━━━━━━━ 363s 2s/step - accuracy: 0.8434 - loss: 0.4310 - val_accuracy: 0.8218 - val_loss: 0.6403
Epoch 20/20
169/169 ━━━━━━━━━━━━━━━━━━━━ 0s 2s/step - accuracy: 0.8799 - loss: 0.3173
Epoch 20: val_accuracy improved from 0.82183 to 0.82851, saving model to model.keras
169/169 ━━━━━━━━━━━━━━━━━━━━ 378s 2s/step - accuracy: 0.8799 - loss: 0.3173 - val_accuracy: 0.8285 - val_loss: 0.6416
```

|  | Epoch | Train Accuracy | Val Accuracy | Train Loss | Val Loss |
|---|---|---|---|---|---|
| 0 | 1 | 0.305082 | 0.409057 | 1.822451 | 1.656794 |
| 1 | 2 | 0.435460 | 0.477357 | 1.479821 | 1.393077 |
| 2 | 3 | 0.503709 | 0.532294 | 1.328908 | 1.233022 |
| 3 | 4 | 0.541543 | 0.593912 | 1.206545 | 1.117739 |
| 4 | 5 | 0.581602 | 0.650334 | 1.097255 | 1.013746 |
| 5 | 6 | 0.634088 | 0.674833 | 0.983413 | 0.907148 |
| 6 | 7 | 0.662277 | 0.688196 | 0.900515 | 0.900463 |
| 7 | 8 | 0.683420 | 0.659985 | 0.828065 | 0.951895 |
| 8 | 9 | 0.721625 | 0.737194 | 0.738557 | 0.745895 |
| 9 | 10 | 0.734236 | 0.755011 | 0.708066 | 0.737823 |
| 10 | 11 | 0.757047 | 0.755754 | 0.638119 | 0.688949 |
| 11 | 12 | 0.768361 | 0.720861 | 0.609212 | 0.827495 |
| 12 | 13 | 0.772626 | 0.767632 | 0.591812 | 0.723106 |
| 13 | 14 | 0.783754 | 0.800297 | 0.570076 | 0.633250 |
| 14 | 15 | 0.817878 | 0.799555 | 0.484036 | 0.629089 |
| 15 | 16 | 0.808420 | 0.810690 | 0.527680 | 0.613062 |
| 16 | 17 | 0.835683 | 0.812918 | 0.436969 | 0.660480 |
| 17 | 18 | 0.833642 | 0.804009 | 0.429212 | 0.653121 |
| 18 | 19 | 0.842915 | 0.804751 | 0.415314 | 0.673575 |
| 19 | 20 | 0.853487 | 0.767632 | 0.385649 | 0.790744 |

Figure 6: Model Training

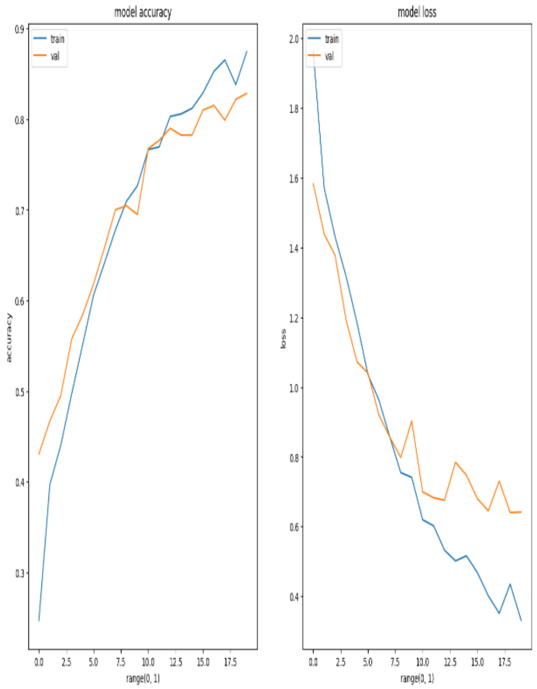


Figure 7: Model Accuracy and Model Loss

The CNN model achieved the following performance metric:

Table 1: Result Table

| Accuracy | Precision | Recall | F1-score | ROC-AUC |
|---|---|---|---|---|
| 85% | 91% | 89% | 0.89% | 0.96 |

The CNN model achieved the following performance metric:

Table 2: ResultComparision Table

| Algorithms | CNN Model | ResNet 50 | DenseNet121 | VGG11 BN |
|---|---|---|---|---|
| Validation Accuracy | 79% | 90% | 90% | 85% |

The outcomes of every model we have tried are displayed in table 2. Overall, the exactness of ResNet50 was generally around 90%. We utilized 10 epochs for ResNet50, DenseNet 121, and VGG11, and 50 epochs for our CNN model. We used this model to assess our system since ResNet50 consistently produced reliable findings [4][5][9][14][22].

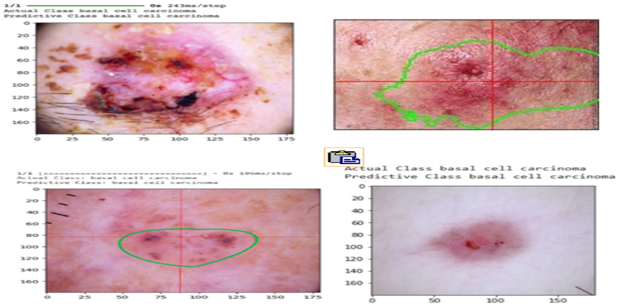

```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 243ms/step
Actual Class basal cell carcinoma
Predictive Class basal cell carcinoma
```
```
1/1 [==============================] - 0s 196ms/step
Actual Class: basal cell carcinoma
Predictive Class: basal cell carcinoma
```
```
Actual Class basal cell carcinoma
Predictive Class basal cell carcinoma
```

Figure 9. Predicted Output

## VII.CONCLUSION

This paper discusses the concept of using a CNN model with the Keras Sequential API to detect different types of skin cancer. We have established the kernel size for our convolution and the filter size for our max-pooling. We created the ANN using Keras after flattening these into a 1D single vector. The 10015 images in our database were split into 80:20 test and train sets. 10% validation was applied to the train set. We trained our six-layered CNN model for 50 epochs, obtaining about 79% accuracy with the validation set and 76% accuracy with the test set. We used the CNN method to increase accuracy in VGG11, ResNet50, and DenseNet121 models that utilize ImageNet's pre-trained data. Our dataset grew as a result of these models, increasing the model's effectiveness. We were able to attain 90% training accuracy with minimal loss using them. For evaluation, we employed 10 epochs for each of these models. We produced a variety of graphics, including faulty prediction graphs, loss and accuracy graphs, and confusion matrices. We also calculated accuracy, supportMetrics such as F1 score and recall were considered. A system that can precisely detect melanoma is critical for dependable outcomes 90% of things just from the information obtained from the picture can be applied in scenarios where this model is more efficient than human detection.In the future, we plan to use our strategies for diagnosing skin cancer in real-world circumstances to support people who faces with restricted usage to medical care.

## REFERENCES

[1] S.S. Han, I.J. Moon, W. Lim, et al. Region-based convolutional 3 neural networks for the identification of keratinocytic skin cancer on the face JAMA Dermatol, 156 (1) (2020), pp. 29-37, 10.1001/jamadermatol.2019.3807

[2] Nazi Z Al, Abir TA. Automatic skin lesion segmentation and melanoma detection: transfer learning approach with U-Net and DCNN-SVM. 2020:371-381. doi:10.1007/978-981-13-7564-4_32.

[3] K. Sau, P. Saha Skin cancer classification using an ANN developed with the scaled conjugate gradient algorithm Communications in Computer and Information Science, Vol 1030, Springer Verlag (2019), p. 134, 10.1007/978-981-13-8578-0_11

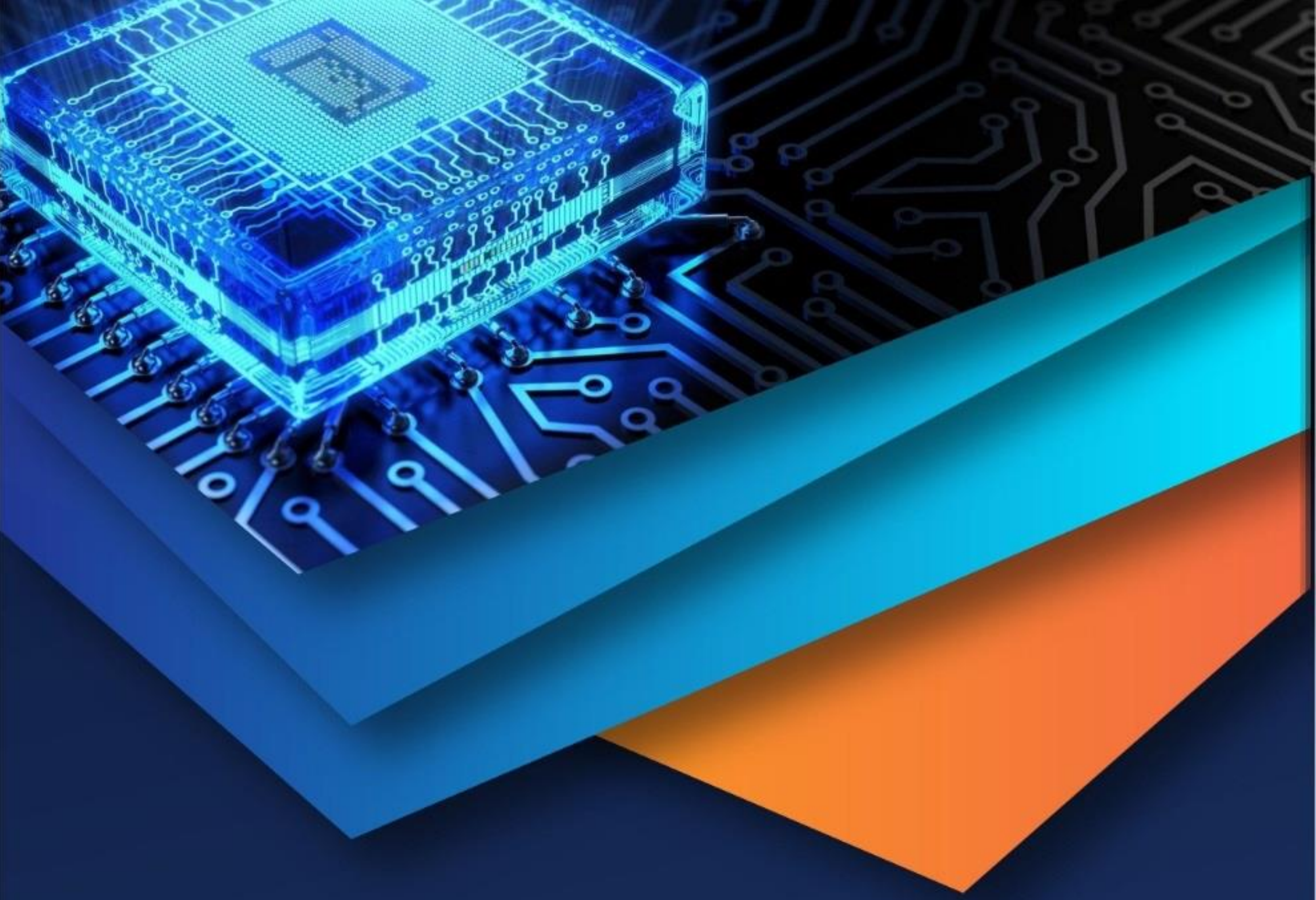
[4] D. Roffman, G. Hart, M. Girardi, C.J. Ko, J. DengA Multi-Parameterized ANN approach for Predicting benign Skin Cancer Sci Rep, 8 (1) (2018), 10.1038/s41598-018-19907-9

[5] Nagavkar G, Potdar K. *A* Sign language translator glove view project blind help system view project Kedar Potdar NVIDIA language recognition system. A classifier employing neural network architecture based.

[6] The American system for recognising sign language. Vol 5.; 2017. https://www.researchgate.net/publication/319624727

[7] M. A. Sheha, M. S. Mabrouk, A. Sharawy,et al., In the study 'Automatic Detection of Melanoma Skin Cancer Using Texture Analysis,' published the *International Journal of Computer Applications*, Vol. 42, no. 20, pp. 22–26, 2012.

[8] V. Noronha, U. Tsomo, A. Jamshed, M. Hai, S. Wattegama, R. Baral, M.Piya, and K. Prabhash, "A fresh look at oncology facts on South central Asia and saarccountries",South Asian journal of cancer, vol. 1, no. 1, p. 1, 2012.



# INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY