Library Progress International
Vol.44, No.1s, January-June 2024: P.42-54
DOI: https://doi.org/10.52710/lpi.44.1s.4

Print version ISSN 0970 1052
Online version ISSN 2320 317X

Original Article
Available online at www.bpasjournals.com

Revolutionizing Skin Cancer Classification: Unveiling the Potential of Information Science and Transfer Learning Architectures for Enhanced Diagnosis

Shailja Pandey

Author's Affiliation:
Department of Computer Science and Engineering, School of Engineering, Babu Banarasi Das University, Lucknow, India
E-mail: cuteshailja4@gmail.com

How to cite this article: S. Pandey (2024). Assessing Research Quality through Scopus and WoS Insights: Case Study of MSIT, *Library Progress International,* 44(1s), 42-54.

ABSTRACT
The deadliest kind of skin cancer is malignant melanoma. To help doctors make more precise diagnoses of skin malignancies, dermoscopy uses noninvasive high-resolution imaging. Melanoma is a malignant skin cancer that grows rapidly and aggressively. Malignant melanoma continues to rank among the world's most rapidly expanding malignancies because of this trait. After spreading to other organs or tissues, the likelihood of a positive response to therapy drops to 5%, and the likelihood of survival after 10 years drops to around 10%. There is currently no therapeutic option that involves surgical removal after it has metastasized. Therefore, it is of the utmost importance to diagnose malignant melanoma early. The false negative ratio for melanoma is the highest of all skin malignancies. One of the most popular and effective approaches to medical picture processing right now is Deep Neural Networks (DNN). While Deep Learning has shown promising results, there are still obstacles to overcome when applying it to these kinds of problems. These include issues like data volatility, noise sensitivity, and insufficiently large training datasets. With an emphasis on medical (clinical) image difficulties, this study offers strategies to help deep-learning models handle these concerns, specifically when it comes to skin cancer diagnosis. This research delves into the topic of melanoma classification using transfer learning and compares several state-of-the-art Convolutional Neural Network (CNN) designs. Many convolutional neural network (CNN) models have been pre-trained on the extensive ImageNet dataset; these models are among those that have been considered for this analysis. Dermoscopy pictures of the skin taken from the publicly available dataset are used to refine and test these models. To increase the dataset size, images are enhanced. The results show that various CNN architectures have varied strengths and weaknesses when it comes to this categorization job. The VGG16 model outperformed all others with a test accuracy of 85.76 percent and a train accuracy of 90.1 percent. This research sheds light on how to use dermoscopy pictures for effective deep-learning skin cancer screening and detection. There are more chances to adjust the model's hyperparameters and increase the variety and amount of the training data.

KEYWORDS
Deep Learning, Medical image analysis, Transfer Learning, Skin Cancer Classification, Melanoma, Malignant Melanoma, Dermatology

1. Introduction

The Cancer is among the top killers in the world. Cancer is projected to surpass all other causes of mortality by 2030, accounting for 13.1 million deaths, according to data compiled by the World Health Organization[1]. Skin cancer ranks first among all cancers in the United States. Estimates show that 20% of the US population will be affected by skin cancer at some point in their lives. Not all cases of skin cancer are lethal[2]. However, saving lives is greatly aided by early detection. It is necessary to study human skin and various skin cancers to comprehend skin cancer early detection and diagnosis. Skin cancer may be either basal cell carcinoma (BCC), squamous cell carcinoma (SCC), or melanoma (MEL). Despite its rarity, melanoma is the deadliest form of cancer due to its high metastasis rate[3].

1.1 Convolutional Neural Networks (CNN)

One subset of Artificial Neural Networks developed specifically for use with grid-like data is the Convolutional Neural Network or CNN[4]. Natural Language Processing (NLP), autonomous video categorization, speech recognition, and self-driving cars are just a few of the numerous real-world challenges that have found effective applications of CNN architectures. When it comes to learning visual characteristics from images, this deep-learning approach is currently the most effective. CNNs are networks that use the convolution operator in one of their layers rather than standard matrix multiplication, as the name implies. A key distinction between this network and more conventional Multi-Layer Perceptron(MLP) is its emphasis on heavy-weight sharing as a means of lowering computational complexity.

Research into the visual cortex, the area of the brain that processes visual information, served as an early model for the convolutional neural network. Here, several brain regions work together to analyse visual data in a hierarchical fashion; each region is responsible for a unique task. The visual cortex is home to neurons with a narrow local receptive field, meaning they can only process visual information from a very specific area of the visual spectrum. Neurons can share the whole vision field in their receptive fields.

1.1.1 CNN layers

The three main layers of a typical CNN are the convolutional, pooling, and fully-connected ones:

Convolutional layer

The network's convolutional layer is its most crucial component. A mathematical operation called the convolution operator takes two functions, g, and h, and returns a third function, which may be seen as the way h changes g. This is the basis of the system:

$$(g \otimes h) = \int_{-\infty}^{\infty} g(\tau)(t-\tau)d\tau \qquad (1)$$

where the integral is defined for two continuous functions, g and h. Looking at the discrete case:

$$(g \otimes h)[n] = \sum_{m=-\infty}^{\infty} g[m]h[n-m] \qquad (2)$$

The input, or function g in the jargon of convolutional networks, is a multidimensional array of data, such an image. The h-function is a filter or kernel, which are array of learnable parameters with several dimensions. Finally, feature maps are another kind of multidimensional array of data that is produced by convolution. Tensors are a common term for these arrays with several dimensions. Suppose we have a 2D input matrix I and a 2D kernel K. To apply Equation (2) to a limited number of array items, we assume that all undefined points in these matrices are zero.

$$F_{map} = (I \otimes K)[i,j] = \sum_{n1} \sum_{n2} I[n_1, n_2]K[i-n_1, j-n_2] \qquad (3)$$

The following is the formula for calculating the spatial size of a convolutional layer's feature maps:

$$s_{fm} = \frac{s_{img} - s_k + 2zp}{str} + 1 \qquad (4)$$

Pooling layer

One function that summarises a portion of the feature maps using pre-defined statistics is the pooling operator, which is also called sub-sampling. Two hyperparameters—the pooling stride and the size of the spatial neighbourhood (spool)—must be defined initially (strpool). The next step is to use strpool points of spacing to summarise each area [spool, spool]. This function may take two possible values: the maximum (MaxPool) and the average (AvgPoll).

Reducing the number of parameters and computation in the network, as well as the spatial size of feature maps, is an obvious advantage of the pooling layer. Furthermore, the network is better able to handle slight input translations and distortions because to the pooling operation, which summaries activations across a whole neighbourhood. One thing to keep in mind is that a convolutional network doesn't always need a pooling layer to function correctly.

1.2 Deep Learning and Transfer Learning

Deep learning, which is a subfield of ML backed by a plethora of algorithms, is also called Deep Structured Learning. There is a cascade of numerous layers in deep learning that is comparable to NNs, as most recent deep learning models are built on neural networks[5]. Deep learning differs from traditional machine learning methods in that it can directly extract valuable features from a variety of inputs, including pictures, text, and audio, in both supervised and unsupervised ways. Feature extraction is really seen as an integral aspect of learning when using this technique. There is a decreased demand for ML solutions that need manual tuning due to these deep learning properties.

Most modern deep learning applications, particularly in computer vision, depend on transfer learning. One machine learning (ML) approach is transfer learning, which involves reusing a learned model for a different but similar job. Training a CNN from the beginning is a time-consuming process, and most issues in the medical computer vision area, such as skin cancer detection, use small data sets (for instance, there are only hundreds of images, while CNNs need considerably larger datasets). For this reason, it is usual to practice to employ a network that has been pre-trained on a large dataset (such as ImageNet's 1.2 million images) as an initialization for the job at hand [6].

Here are two of the most typical applications of transfer learning:

Fixed Feature Extractor: The pre-trained model may be used to extract features. It does this by repurposing the remaining network nodes as a fixed feature extractor for the dataset of interest after deleting the output layer or the last fully connected node.

Fine-tuning: To fine-tune anything is to make little changes so it performs better. To illustrate, given a single dataset, it is possible to arbitrarily divide it into a training dataset and a testing (validation) dataset according to any ratio desired. After that, we can use the training dataset to train the model file, and then we can use the testing dataset to train the same model.

Several reasons that take use of pre-trained neural network designs drive the adoption of transfer learning in skin cancer categorization. Some important justifications for using transfer learning here are as follows:

- Limited Data Availability:

The difficulties in gathering annotated medical pictures mean that medical datasets, especially skin cancer datasets, are often on the smaller side. To improve the model's performance on the smaller target dataset, transfer learning enables using information from big, diversified datasets in other fields (like ImageNet).

- Feature Extraction and Generalization:

Many visual identification applications might benefit from the extensive hierarchical features learnt by pre-trained models, particularly those trained on big and varied datasets[7]. To help the model generalise to unseen skin cancer pictures, transfer learning allows for the retrieval of significant information from the early layers of the neural network.

- Reduction in Training Time and Resources:

It may be computationally costly and time-consuming to train deep neural networks from scratch using medical picture datasets. By using the information already contained in the weights of pre-trained models, researchers may drastically cut down on the time and computing resources needed for training via transfer learning.

- Addressing Overfitting:

When working with a limited target dataset, transfer learning may help reduce the likelihood of overfitting. Models that draw from a wide variety of source domains are better able to generalise to novel instances because they are less prone to remember the destination dataset's noise.

- Enhanced Model Convergence:

In many cases, the training process may be expedited by transfer learning. The model can swiftly adjust its parameters to the job at hand because to the pre-trained weights, which serve as a solid foundation for optimization.

- Improved Performance in Complex Tasks:

Understanding complicated patterns and subtle traits is crucial for skin cancer categorization, which is a challenging undertaking in and of itself.

Due to their superior performance in picture recognition tasks, transfer learning architectures like deep convolutional neural networks (e.g., VGG, ResNet, Inception) are ideal for the complex dermatological image processing.

- Accessibility of Pre-trained Models:

Skin cancer classification jobs are made easier with the availability of well-established pre-trained models like ResNet50, InceptionV3, VGG16, and others. This architectural library is easily accessible to researchers, who can then tailor it to their needs in medical imaging.

This research compares and evaluates several state-of-the-art convolutional neural networks (CNNs) that use transfer learning to classify melanoma from dermoscopy pictures. The performance is evaluated using important diagnostic metrics on the massive dataset. The results should help guide the development of deep learning algorithms for use in the assessment and detection of skin cancer. Such AI systems have the potential to improve patient outcomes by facilitating earlier and more precise identification.

1.3 Outline of the paper

Following is the structure of the remaining part of the paper: Several other researchers in the area of skin cancer categorization have investigated the core ideas of Deep Learning, which are presented in Section 2. Section 3 sheds information on the proposed methodology. The findings and their in-depth analysis for the identification of skin cancer using the dataset are described in Section 4. Section 5 is the last section, and it is where the conclusions are drawn as well as the directions that need to be addressed to go on.

2. Literature Review

The use of Deep Learning models for the interpretation of medical images has just lately become practical. Hence, several models have been suggested to address the issue of skin cancer detection[8], [9]. Romeo et al.[10] introduced a deep convolutional neural network (CNN) and many methods for learning with sparse input. After training more than 120,000 images using a pre-trained GoogleNet CNN architecture, Shoeib et al.[11] were able to attain a diagnosis level comparable to that of a dermatologist. Additionally, Kalouche et al.[12] and Bi et al.[13] compared the efficacy of pretrained models to that of dermatologists. Both studies found that the models were just as good as, if not better than, the dermatologists. Additional deep learning-based skin cancer detection initiatives include feature aggregation across models[14], and ensembles of models[15], [16], [17], [18].

Models that integrate skin lesion images with patient demographics have been suggested by Harangi et al.[19], Sun et al.[20]., and Rutkowski [21]. Despite the encouraging results, these studies all use feature concatenation to merge the two sets of data, which can overlook the connection between picture metadata and the visual features retrieved from the images. A multiplication-based fusion method that integrates picture characteristics with information using 1D convolution was recently suggested by Zaman et al.[22]. Results for skin cancer detection were promising, according to the investigators, but they only covered a small subset of patients. To sum up, we need to look at better aggregation approaches since there is space for improvement.

3. Material and Method

3.1 Dataset

This study's dataset comes from the International Skin Imaging Collaboration (ISIC)[23], which has a balanced collection of pictures of benign and malignant skin moles. Each kind of mole is represented by 1800 images, with dimensions of 224 x 244. Figure 1 displays several example images.

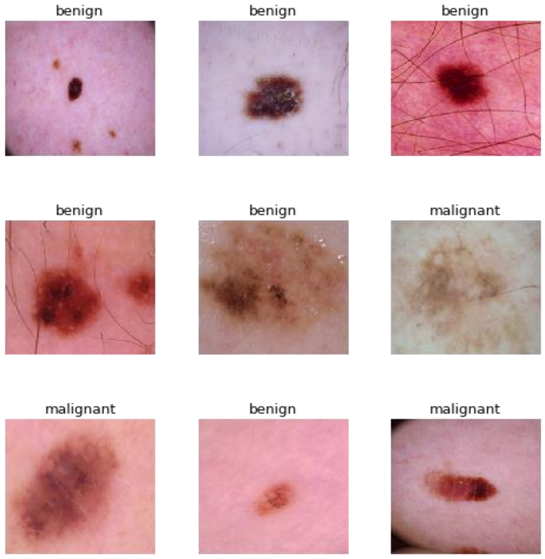

Fig. 1 Sample images from the dataset

3.2 Methodology

3.2.2 Data Preprocessing:

- Set all of the picture dimensions and resolutions to the same.
- Put normalization and augmentation methods to work to make datasets more unpredictable.
- Indicate the kinds of skin cancers seen in the images.

3.2.2 Transfer Learning Model Selection:

- When classifying skin cancers, use one of these pre-trained models: ResNet50, InceptionV3, VGG16, VGG19, MobileNetV2, MobileNet, DenseNet121, InceptionResNetV2, or NASNetMobile.
- The pre-trained weights may be obtained from reliable sources such as ImageNet.

ResNet 50

A deep convolutional neural network architecture, ResNet50 stands for Residual Network with 50 layers[24]. Here is the mathematical expression for a generic ResNet residual block:

The residual block is trained using an input x and a residual mapping denoted by F(x). This is how the residual block calculates its output, y:

$$y=F(x)+x$$

The residual block can be further broken down into the following steps:

i. Projection Shortcut:

If the dimensions of x and $F(x)$ are not the same, a linear projection W_s is applied to x to match the dimensions. The projection is defined as: $\text{proj}(x)=W_s \cdot x$

ii. Convolutional Layers:

$F(x)$ typically consists of multiple convolutional layers, batch normalization, and activation functions. The output is obtained by passing x through these layers.

$$F(x)=\text{ReLU}(\text{BN}(\text{Conv}(x,W_i))+W_b)$$

where:

- Conv is the convolution operation.
- BN is the batch normalization.
- W_i are the weights of the convolutional layer.
- W_b are the biases.

iii. Final Output:

The final output y is obtained by adding the residual mapping $F(x)$ to the input x.

$$y=F(x)+\text{proj}(x)$$

A ResNet50 network, for example, might have several convolutional layers organised into stacked residual blocks. Using residual connections, the architecture as a whole is designed to make training extremely deep networks easier while reducing the impact of the vanishing gradient issue.

InceptionV3

A deep convolutional neural network architecture, InceptionV3 is renowned for its inception modules. These modules use many filters of varying sizes in parallel to gather data at different scales[24]. Mathematically, a generic Inception module in InceptionV3 looks like this:

Assume that x is the input to the inception module and that F(x) is the mapping that has to be learnt. The inception module's output y is calculated by merging the outcomes of many simultaneous operations:

$$y=\text{Concat}([F_1(x),F_2(x),F_3(x),\ldots,F_n(x)])$$

The input x is processed differently by each of the individual branches of the inception module, and each $F_i(x)$ represents that operation.

The operations within each branch typically include:

- **1x1 Convolution:** $F_i(x)=\text{ReLU}(\text{BN}(\text{Conv}_{1\times1}(x,W^i_{1\times1}))+W^i_b)$
- **3x3 Convolution**$F_i(x)=\text{ReLU}(\text{BN}(\text{Conv}_{3\times3}(x, W^i_{3\times3}))+W^i_b)$
- **5x5 Convolution:** $F_i(x)=\text{ReLU}(\text{BN}(\text{Conv}_{5\times5}(x, W^i_{5\times5}))+W^i_b)$
- **Pooling Operation:** $F_i(x)=\text{Pool}_{\text{type}}(x)$

After all the branches have been combined, the final output y is achieved. For even more efficient computing, the inception module may include a 1x1 convolution prior to the concatenation phase, in addition to the parallel operations. This will decrease the number of channels.

VGG16 and VGG19

The simplicity and recurrence of 3x3 convolutional layers distinguish the VGG16 and VGG19 designs[25]. A generic VGG block with two 3x3 convolutions may be expressed mathematically as follows:

The input to the VGG block may be represented as x, and the mapping to be learnt can be denoted as F(x). Here is how the VGG block calculates its output y:

$$y=ReLU(BN(Conv_{3\times3}(x,W_1))+W_{b1})$$
$$y=ReLU(BN(Conv_{3\times3(}y,W_2))+W_{b2})$$

In this formulation:

- Conv3×3 denotes the 3x3 convolution operation.
- The first convolutional layer's weight is W_1 and the second layer's weight is W_2.
- W_{b1} and W_{b2} are the biases.
- The rectified linear unit activation function is abbreviated as ReLU.

This kind of block is stacked in both VGG16 and VGG19. In terms of network depth, the only variation between VGG16 and VGG19 is the quantity of these blocks. By piling these blocks and finishing with completely linked layers, the VGG16 and VGG19 architectures are constructed. While VGG19 and VGG16 have more complex architectures that include features like fully linked and pooling layers, the above description gets to the heart of what a VGG block is all about: the convolutional layers.

MobileNet:

The depth wise separable convolutions used by MobileNet are composed of a depthwise convolution and a 1x1 pointwise convolution. In MobileNet, a depthwise separable convolution may be expressed mathematically as follows:

$y=ReLU(BN(DepthwiseConv(x,W_d))+W_{bd})$

$y=ReLU(BN(PointwiseConv(y,W_p))+W_{bp})$

In this formulation:

- DepthwiseConv represents the depthwise convolution operation.
- PointwiseConv represents the 1x1 pointwise convolution operation.
- W_d and W_p are the weights for the depthwise and pointwise convolutions, respectively.
- W_{bd} and W_{bp} are the biases.

MobileNetV2:

MobileNetV2 enhances the depthwise separable convolution with an inverted residual structure[26]. The mathematical formulation for a single block in MobileNetV2 is as follows:

$y=LinearBottleneck(x,W_s,W_d,W_p,W_{bd,}W_{bp},Wb_s)$

In this formulation:

- LinearBottleneck represents the inverted residual block.
- W_s, W_d, W_p, W_{bd}, W_{bp}, and W_{bs} are the weights associated with the block.
- The specific details of the inverted residual block involve a combination of depthwise separable convolutions, skip connections, and linear bottleneck structures to achieve a good balance between efficiency and performance.

DenseNet121

A neural network design known as DenseNet (Densely Connected Convolutional Networks) links all of the layers in a feed-forward manner. The version with 121 layers is particularly called DenseNet121. The dense block, made up of layers that are densely coupled, is the main part of DenseNet. Theoretically, DenseNet121's general dense block looks like this:

An example of an input to a dense block is x, and the mapping to be learnt inside the dense block is denoted as F(x). The dense block's output, y, is calculated by adding the outcomes of all the preceding layers:

$$y=Concat([F_1(x),F_2(x),\ldots,F_n(x)])$$

Each $F_i(x)$ corresponds to the output of the i-th layer within the dense block.

The operation within each layer typically includes:

$$F_i(x)=ReLU(BN(Conv_{3\times3}([x,F_1(x),F_2(x),\ldots,F_{i-1}(x)],W_i))+W_{bi})$$

In this formulation:

- W_i are the weights of the convolutional layer within the i-th layer.
- W_{bi} are the biases.

Throughout the network, the dense block is replicated, and information may be easily sent from lower to higher levels using skip links. Layers of transitions, global average pooling, and several dense blocks make up the whole DenseNet121 design.

NASNetMobile:

NASNetMobile (Neural Architecture Search Network Mobile) is a neural network architecture designed through neural architecture search[27].

The mathematical formulation for a generic cell in NASNetMobile can be expressed as a combination of normal and reduction cells:

$$y=\text{Cell}(x,W)$$

Here:

- Cell represents a cell in NASNetMobile.
- x is the input to the cell.
- W represents the weights associated with the cell.

Multiple procedures are involved in the cell's construction, including as convolutional layers, pooling, skip connections, and nonlinear activation functions. The search procedure for neural architecture determines the particular processes and the parameters that govern them.

3.2.3 Model Fine-tuning:

Substitute a new layer with the same number of skin cancer classes for each pre-trained model's final completely linked layer. Stop overfitting from happening by freezing the first layers of the chosen architectures. This will keep the learnt features.

3.2.4 Training Process:

- Make three separate sets: one for training, one for validation, and one for testing.
- Use the training set to train your transfer learning models, and use the validation set to test them.
- To avoid overfitting and preserve the top-performing models, use early halting.

The complete flow of methodology is shown in fig. 2.

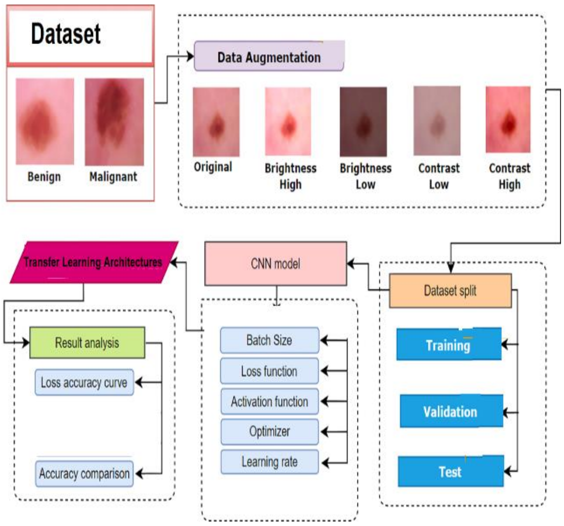

Fig. 2 Flow of proposed methodology

4. Results and Discussion

4.1 Experimental Setup

In this study, we implemented the algorithms using several open-source frameworks. Python is the foundational language for all of these frameworks. We use NumPy and Pandas to manage generic data. We use OpenCV and Pillow for image processing. We ran the scikit-learn package for measurements and methods that are common in machine learning. Matplotlib and Seaborn are used for all visualisations. And lastly, Pytorch is used for all the deep learning model implementations.

4.2 Results

Table 1 training and testing accuracy/loss for different deep learning architectures.

Table 1 training and testing accuracy/loss comparison

Model	Train Loss	Train Acc.	Test Loss	Test Acc.
CNN	0.5861	0.6310	0.5769	0.6394
ResNet50	0.1807	0.9367	0.3968	0.8242
InceptionV3	0.1321	0.9594	0.4154	0.8364
VGG16	0.2349	0.9010	0.3271	0.8576
VGG19	0.2768	0.8760	0.3610	0.8394
MobileNetV2	0.3574	0.8354	0.4281	0.7848
MobileNet	0.3457	0.8369	0.4122	0.8121
DenseNet121	0.1945	0.9314	0.3279	0.8545
InceptionResNetV2	0.0924	0.9735	0.4736	0.8379
NASNetMobile	0.1702	0.9416	0.3709	0.8333

Fig. 3-4 shows the comparison of training and testing accuracy/loss of state-of-art CNN and used deep learning architectures (ResNet50, InceptionV3, VGG16, VGG19, MobileNetV2, MobileNet, DenseNet121, InceptionResNetV2, and NASNetMobile).

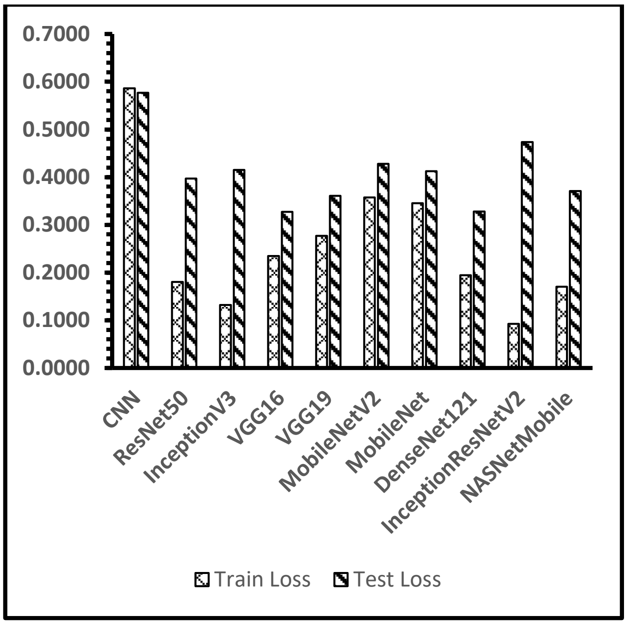

Fig. 3 Training/testing loss comparison

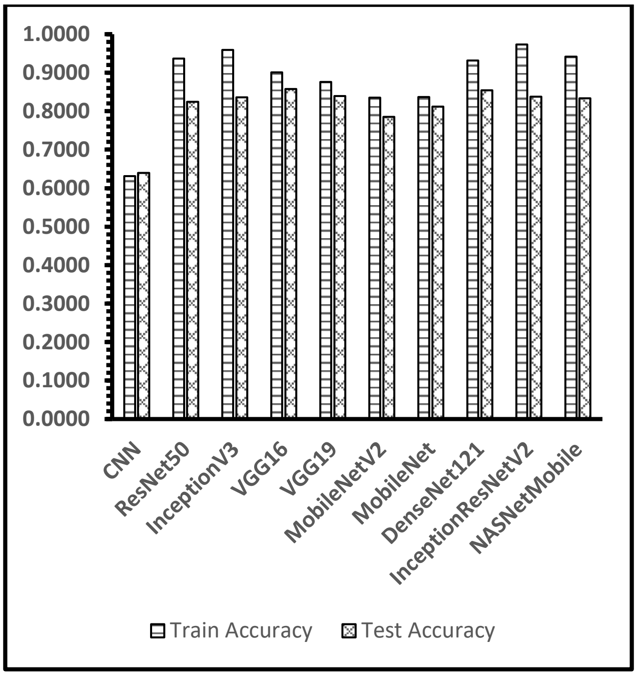

Fig. 4. Training/testing accuracy comparison

Accuracy and loss curves for used deep architectures are shown in fig 5-14.

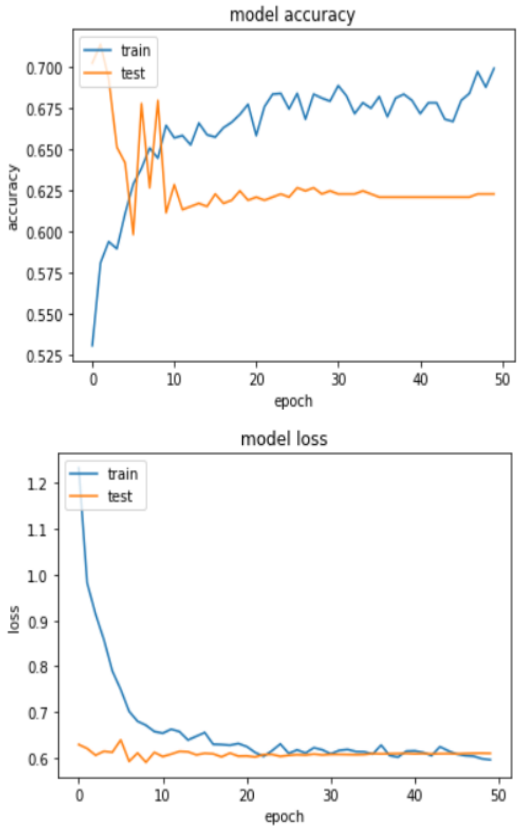

Fig. 5 Accuracy and loss curve for CNN

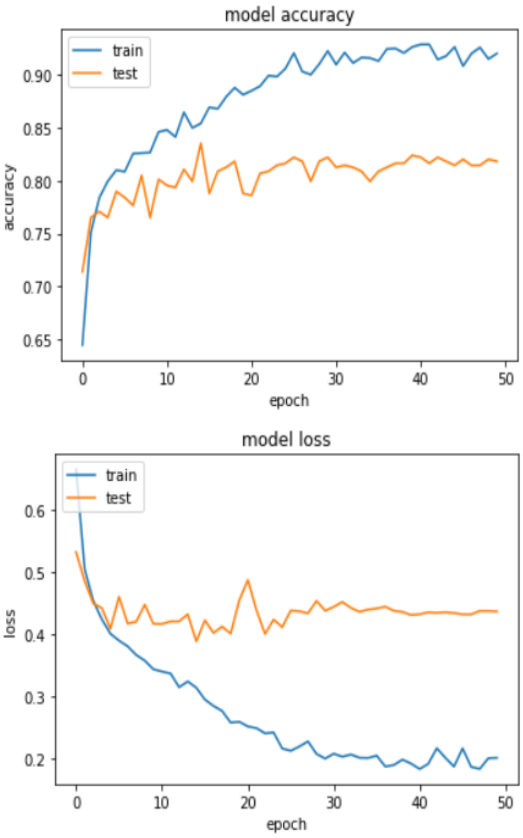

Fig. 6 Accuracy and loss curve for ResNet50

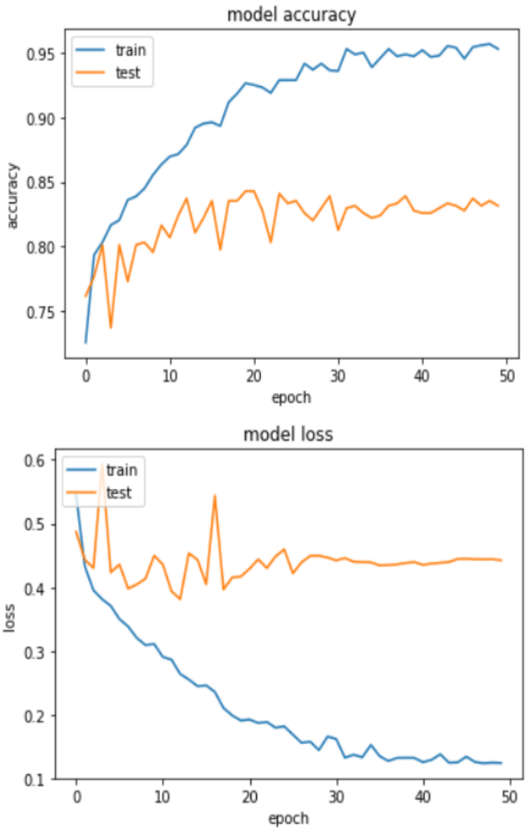

Fig. 7 Accuracy and loss curve for InceptionV3

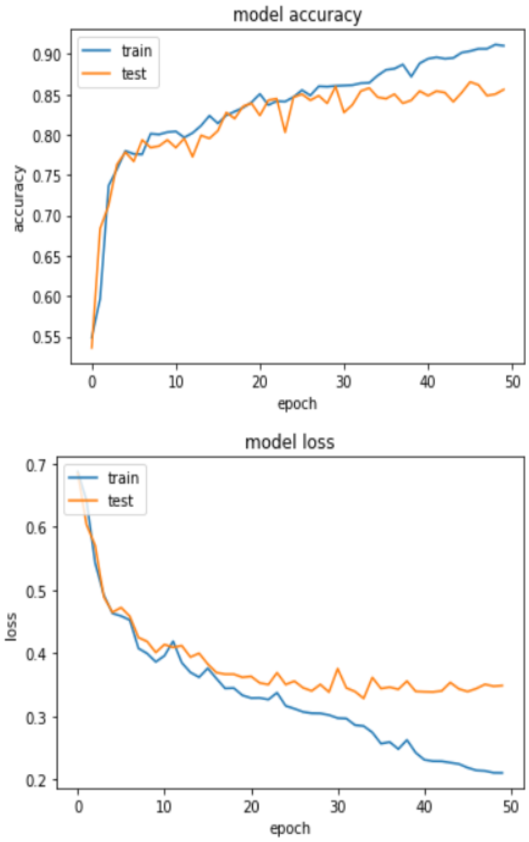

Fig. 8 Accuracy and loss curve for VGG16

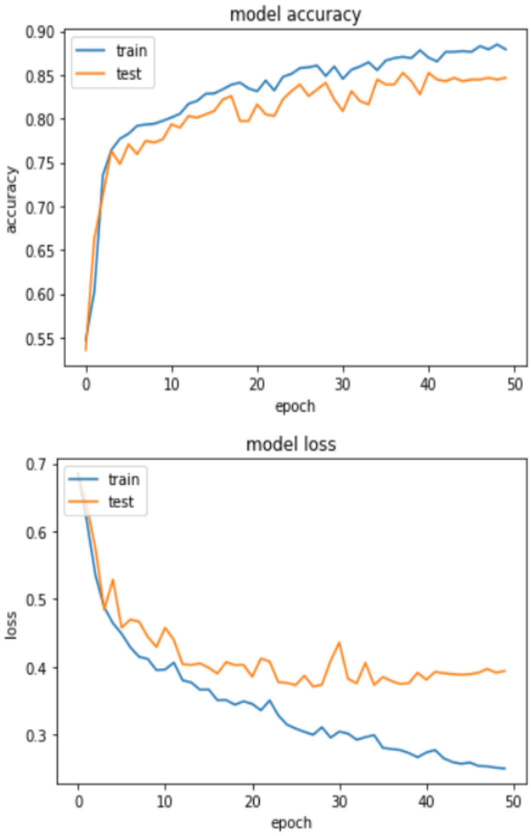

Fig. 9 Accuracy and loss curve for VGG19

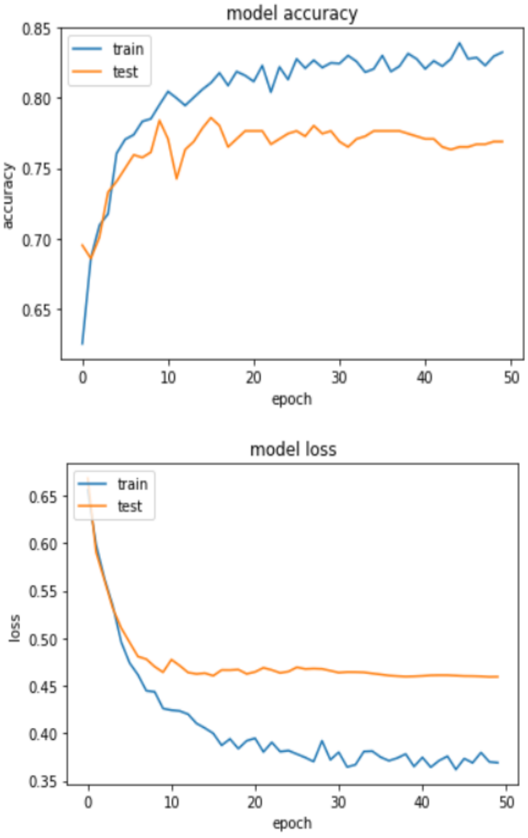

Fig. 10 Accuracy and loss curve for MobileNetV2

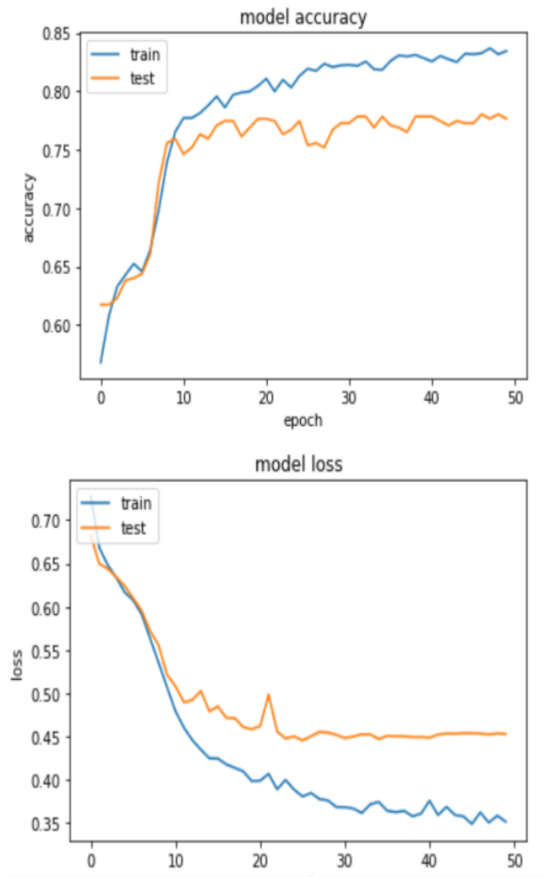

Fig. 11 Accuracy and loss curve for MobileNet

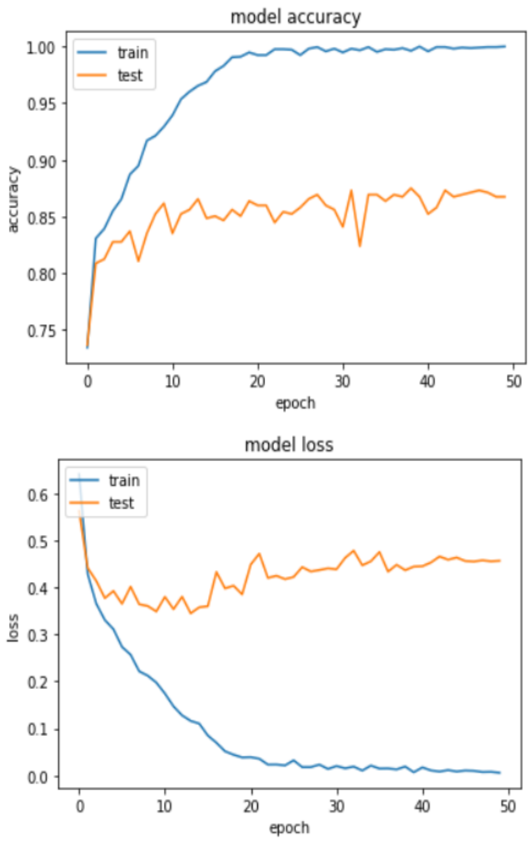

Fig. 13 Accuracy and loss curve for InceptionResNetV2

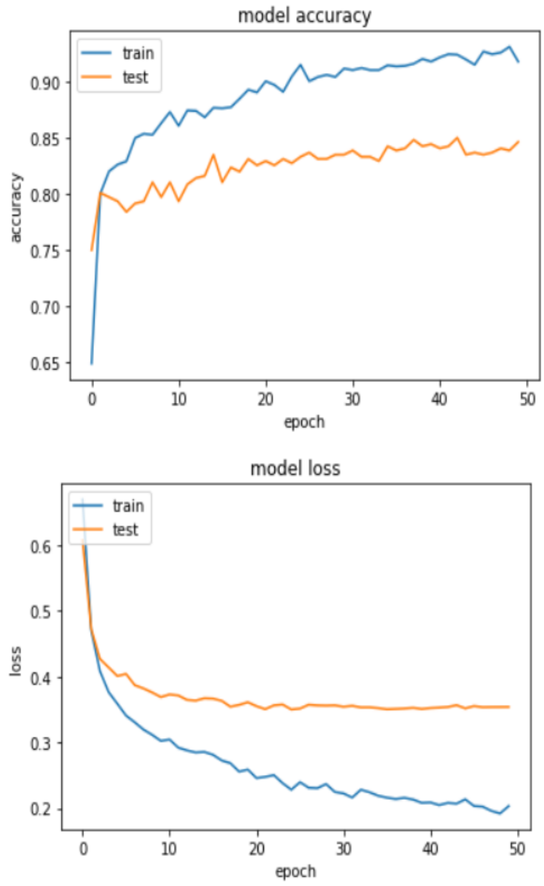

Fig. 12 Accuracy and loss curve for DenseNet121

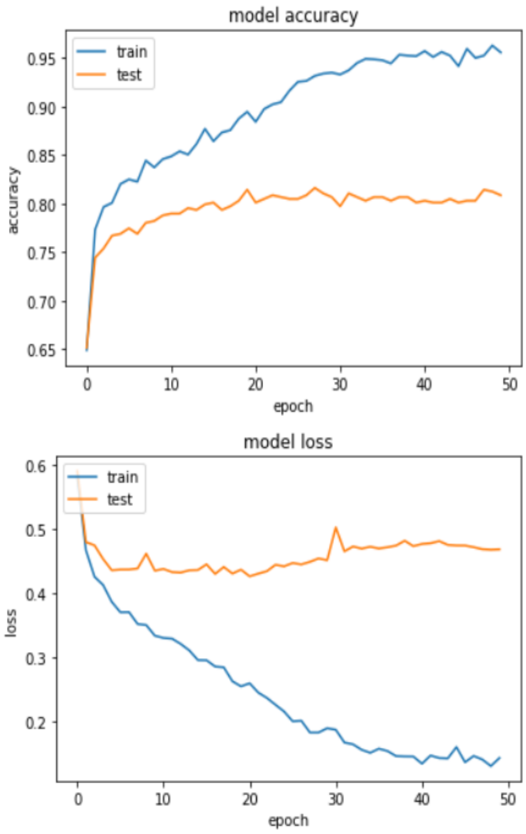

Fig. 14 Accuracy and loss curve for NASNetMobile

4.3 Discussion

Based on the results from fig. 3-4, we can make several key observations:

i. With reduced loss and improved accuracy, the deep learning models decisively defeat the baseline CNN model on the test and train sets. This demonstrates how pre-trained network transfer learning methods are better.

ii. With the quickest convergence and overfitting on the training data, InceptionV3 obtains the best train accuracy and lowest train loss. On the other hand, its advantage is lessened when tested.

iii. With an accuracy rate of 85.76% and a test loss of just 0.3271%, VGG16 outperforms the competition. This indicates that, in comparison to the other models, it has superior generalizability.

iv. Although they perform well on the training set, models such as InceptionResNetV2 and NASNetMobile fail miserably when tested. Their decreased test accuracy and relatively large test loss demonstrate the degree to which overfitting has occurred.

v. When comparing test results, the majority of models, except VGG16, show very slight variations. The effectiveness of transfer learning in this context is shown by the excellent accuracy attained by all deep networks.

vi. The difference between the model's test and train scores indicates that there is room for improvement in optimizing the model's hyperparameters and the use of extra regularisation approaches to mitigate overfitting. Strengthening assessments is another benefit of increasing the size of the collection.

In summary, VGG16 emerges as the most balanced model but the competitive performance of all CNNs demonstrates promising avenues for developing reliable automated skin cancer screening systems.

5. Conclusion

Numerous individuals are impacted daily by the rising incidence of skin cancer. If caught early, this cancer may respond well to treatment. The likelihood of survival and the rate of death may be improved with early detection and treatment. On the other hand, inexperienced doctors and subjective clinical procedures make malignant melanoma diagnoses susceptible to human error. Consequently, better methods are required that can help both experienced and inexperienced doctors. Automated melanoma classification from dermoscopy pictures was the subject of this study's thorough assessment of transfer learning using several state-of-the-art CNN architectures. Several models were evaluated using the publicly available dataset. These included ResNet50, InceptionV3, VGG16, MobileNetV2, and others.

Fundamental results show that deep learning outperforms a baseline CNN, demonstrating the efficacy of transfer learning from large-scale pre-trained networks. The VGG16 model stands out for its optimal accuracy-to-time ratio. The competitive performance seen in the majority of models suggests that there is potential in using these state-of-the-art CNNs for skin cancer screening applications.

The discrepancy between the two sets of results suggests overfitting, which may be mitigated by using strategies like dropout and improved regularisation. Another way to improve generalization is to add more training instances to the dataset. It might also be beneficial to investigate ensemble methods, which combine the predictions of several models, for evaluation purposes. Finally, to find classifier designs that are specifically made for this job, it is recommended to look at model optimization approaches like neural architecture search.

References

[1] C. Barata, M. E. Celebi, and J. S. Marques, "A Survey of Feature Extraction in Dermoscopy Image Analysis of Skin Cancer," *IEEE J Biomed Health Inform*, vol. 23, no. 3, pp. 1096–1109, May 2019, doi: 10.1109/JBHI.2018.2845939.

[2] F. Ulutaş, E. Çomut, and V. Çobankara, "Development of Two Types of Skin Cancer in a Patient with Systemic Sclerosis: a Case Report and Overview of the Literature," *Case Rep Oncol Med*, vol. 2021, pp. 1–5, Feb.

2021, doi: 10.1155/2021/6628671.

[3] R. G. Tiwari, S. Kumar, G. V. Londhe, A. K. Agarwal, and R. Bhardwaj, "Accurate and Automated Deep Learning Solution for Skin Cancer Detection," *International Journal of Intelligent Systems and Applications in Engineering*, vol. 11, no. 5s, pp. 490–500, Apr. 2023, Accessed: Feb. 28, 2024. [Online]. Available: https://ijisae.org/index.php/IJISAE/article/view/2810

[4] M. A. Al-masni, M. A. Al-antari, M. T. Choi, S. M. Han, and T. S. Kim, "Skin lesion segmentation in dermoscopy images via deep full resolution convolutional networks," *Comput Methods Programs Biomed*, vol. 162, pp. 221–231, Aug. 2018, doi: 10.1016/J.CMPB.2018.05.027.

[5] N. Ujjwal, A. Singh, A. K. Jain, and R. G. Tiwari, "Exploiting Machine Learning for Lumpy Skin Disease Occurrence Detection," *2022 10th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO)*, pp. 1–6, Oct. 2022, doi: 10.1109/ICRITO56286.2022.9964656.

[6] R. G. Tiwari, A. Misra, and N. Ujjwal, "Image Embedding and Classification using Pre-Trained Deep Learning Architectures," pp. 125–130, Jan. 2023, doi: 10.1109/ICSC56524.2022.10009560.

[7] R. G. Tiwari, D. S. Yadav, and A. Misra, "Performance Evaluation of Optimizers in the Classification of Marble Surface Quality Using CNN," pp. 181–191, 2023, doi: 10.1007/978-981-19-3148-2_15/COVER.

[8] N. Ujjwal, A. Singh, A. K. Jain, and R. G. Tiwari, "Exploiting Machine Learning for Lumpy Skin Disease Occurrence Detection," *2022 10th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions), ICRITO 2022*, 2022, doi: 10.1109/ICRITO56286.2022.9964656.

[9] T. J. Brinker *et al.*, "Skin Cancer Classification Using Convolutional Neural Networks: Systematic Review," *J Med Internet Res 2018;20(10):e11936 https://www.jmir.org/2018/10/e11936*, vol. 20, no. 10, p. e11936, Oct. 2018, doi: 10.2196/11936.

[10] A. Romero Lopez, X. Giro-I-Nieto, J. Burdick, and O. Marques, "Skin lesion classification from dermoscopic images using deep learning techniques," *Proceedings of the 13th IASTED International Conference on Biomedical Engineering, BioMed 2017*, pp. 49–54, Apr. 2017, doi: 10.2316/P.2017.852-053.

[11] D. A. Shoieb, S. M. Youssef, and W. M. Aly, "Computer-Aided Model for Skin Diagnosis Using Deep Learning," *Journal of Image and Graphics*, pp. 122–129, 2016, doi: 10.18178/JOIG.4.2.122-129.

[12] S. Kalouche, A. Ng, and J. Duchi, "Vision-based classification of skin cancer using deep learning," *2015, conducted on Stanfords Machine Learning course (CS 229) taught*, 2016.

[13] L. Bi, J. Kim, E. Ahn, D. Feng, and M. Fulham, "Semi-automatic skin lesion segmentation via fully convolutional networks," *Proceedings - International Symposium on Biomedical Imaging*, pp. 561–564, Jun. 2017, doi: 10.1109/ISBI.2017.7950583.

[14] A. Mahbod, G. Schaefer, I. Ellinger, R. Ecker, A. Pitiot, and C. Wang, "Fusing fine-tuned deep features for skin lesion classification," *Computerized Medical Imaging and Graphics*, vol. 71, pp. 19–29, Jan. 2019, doi: 10.1016/J.COMPMEDIMAG.2018.10.007.

[15] T. DeVries and D. Ramachandram, "Skin Lesion Classification Using Deep Multi-scale Convolutional Neural Networks," Mar. 2017, Accessed: May 04, 2022. [Online]. Available: http://arxiv.org/abs/1703.01402

[16] A. Esteva *et al.*, "Dermatologist-level classification of skin cancer with deep neural networks," *Nature 2017 542:7639*, vol. 542, no. 7639, pp. 115–118, Jan. 2017, doi: 10.1038/nature21056.

[17] K. Zaman, J. I. Bangash, S. S. Maghdid, S. Hassan, H. Afridi, and M. Zohaib, "Analysis and Classification of Skin Cancer Images using Convolutional Neural Network Approach," *4th International Symposium on*

Multidisciplinary Studies and Innovative Technologies, ISMSIT 2020 - Proceedings, Oct. 2020, doi: 10.1109/ISMSIT50672.2020.9255356.

[18] K. Sriwong, S. Bunrit, K. Kerdprasop, and N. Kerdprasop, "Dermatological classification using deep learning of skin image and patient background knowledge," *Int J Mach Learn Comput*, vol. 9, no. 6, pp. 862–867, 2019, doi: 10.18178/IJMLC.2019.9.6.884.

[19] B. Harangi, "Skin lesion classification with ensembles of deep convolutional neural networks," *J Biomed Inform*, vol. 86, pp. 25–32, Oct. 2018, doi: 10.1016/J.JBI.2018.08.006.

[20] X. Sun, J. Yang, M. Sun, and K. Wang, "A benchmark for automatic visual classification of clinical skin disease images," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 9910 LNCS, pp. 206–222, 2016, doi: 10.1007/978-3-319-46466-4_13/FIGURES/6.

[21] P. Rutkowski *et al.*, "Expert recommendation on diagnostic-therapeutic management in skin carcinomas," *Oncology in Clinical Practice*, vol. 18, no. 2, pp. 69–91, 2022, doi: 10.5603/OCP.2021.0032.

[22] K. Zaman, J. I. Bangash, S. S. Maghdid, S. Hassan, H. Afridi, and M. Zohaib, "Analysis and Classification of Skin Cancer Images using Convolutional Neural Network Approach," *4th International Symposium on Multidisciplinary Studies and Innovative Technologies, ISMSIT 2020 - Proceedings*, Oct. 2020, doi: 10.1109/ISMSIT50672.2020.9255356.

[23] "ISIC | International Skin Imaging Collaboration." Accessed: Feb. 27, 2024. [Online]. Available: https://www.isic-archive.com/

[24] Muhathir, M. F. Dwi Ryandra, R. B. Y. Syah, N. Khairina, and R. Muliono, "Convolutional Neural Network (CNN) of Resnet-50 with Inceptionv3 Architecture in Classification on X-Ray Image," *Lecture Notes in Networks and Systems*, vol. 724 LNNS, pp. 208–221, 2023, doi: 10.1007/978-3-031-35314-7_20/COVER.

[25] S. Mascarenhas and M. Agarwal, "A comparison between VGG16, VGG19 and ResNet50 architecture frameworks for Image Classification," *Proceedings of IEEE International Conference on Disruptive Technologies for Multi-Disciplinary Research and Applications, CENTCON 2021*, pp. 96–99, 2021, doi: 10.1109/CENTCON52345.2021.9687944.

[26] R. Liu, "Paddy Disease Classification Based on the Lightweight MobileNet-V2," *Highlights in Science, Engineering and Technology*, vol. 57, pp. 126–133, Jul. 2023, doi: 10.54097/HSET.V57I.9990.

[27] D. R. Arulanandar, V. Padmanabhan, P. Nammalwar, and S. P. Govindan, "Static signature verification using NASNet deep learning architectures," *AIP Conf Proc*, vol. 2829, no. 1, Jul. 2023, doi: 10.1063/5.0156725/2902162.