Automated Skin Lesion Classification: A Comprehensive Analysis of Transfer Learning and Machine Learning Techniques

Muhammad Zain Amin[1], Md. Imran Hossain[2], Taiabur Rahman[3]

[1,2,3] Erasmus Mundus Joint Master Degree in Medical Imaging and Applications
[1,2,3] University of Casino and Southern Lazio, Italy

Abstract

Skin cancer counts for one-third of all types of cancers and causes many deaths every year. Early detection of this disease could increase the patient's chance of survival but manually detecting cancerous skin lesions is expensive and time-consuming. This raises the need for using the technology to detect skin lesions at an early stage. Hence, an automated classification system for skin cancer diagnosis has proved to be a very helpful tool for dermatologist. The aim of this project is to develop a computer aided diagnosis system in order to detect skin lesions using deep learning and machine learning techniques. In this project, the ISIC2017 challenge dataset is used to classify the lesions categories such as benign, melanoma, and seborrheic keratosis. A two-step hierarchal classification pipeline is developed: the first stage is "benign vs. others" and the second stage is "melanoma vs. seborrheic keratosis". The hybrid model, a combination of Xception and Random Forest, achieves a maximum Balanced Multiclass Accuracy (BMA) score of 79%, outperforming the VGG16, InceptionResNetv2, and DenseNet201 architectures. Also, we will examine the research gaps in the skin cancer domain and address the critical challenges that require further investigation.

Keywords

Skin Lesion, Transfer Learning, Machine Learning, Small Dataset, Image Classification

1. Introduction

Skin lesion refers to an abnormal growth or appearance on the skin that deviates from the surrounding healthy skin. There are two main types of skin lesions: primary and secondary. Primary skin lesions are abnormal skin conditions that can develop gradually or be present from birth. On the other hand, secondary skin lesions are a result of modifications or exacerbations of primary lesions. For instance, when a mole is scraped until it bleeds, a crust forms, resulting in a secondary skin lesion. Dermatologists offer different treatments for affected skin based on the type of lesion, including home care, medications, or surgical interventions. It is important to note that despite their seemingly innocuous appearance, certain skin lesions can pose significant risks to patients as they may indicate the presence of malignancy, necessitating surgical removal. Melanoma, in particular, is the most dangerous form of skin cancer. While it can be deadly once it has spread, early detection greatly increases the chances of successful treatment. Therefore, precise diagnosis of skin lesions is crucial to ensure timely and appropriate care for patients. The American Cancer Society estimates for melanoma in the United States for 2018 are: About 91,270 new melanomas will be diagnosed (about 55,150 in men and 36,120 in women). About 9,320 people are expected to die of melanoma (about 5,990 men and 3,330 women). The rates of melanoma have been rising for the last 30 years. Melanoma is more than 20 times more common in whites than in African Americans. Overall, the lifetime risk of getting melanoma is about 2.6% (1 in 38) for whites, 0.1% (1 in 1,000) for blacks, and 0.58% (1 in 172) for Hispanics [1, 2].

Computer-Aided Diagnosis (CAD) systems have emerged as powerful tools in the medical field, offering the potential to automate analysis and provide contextual relevance, thereby improving clinical reliability and aiding physicians in making objective decisions. These systems hold the promise of reducing errors related to human fatigue, enhancing communication between healthcare professionals, lowering mortality rates, and potentially reducing overall medical costs. In the specific domain of dermatology, CAD systems have shown great potential in assisting with the identification and classification of skin lesions, particularly pigmented lesions that may be indicative of melanoma or other forms of skin cancer [3].

To achieve accurate classification of pigmented skin lesions, machine learning and deep learning methods have been extensively explored. Convolutional Neural Networks (CNNs), a type of deep learning architecture, have demonstrated exceptional performance in image recognition tasks [4]. In our proposed work, a range of well-established CNN frameworks, including VGG16, ResNet50, InceptionV3, Xception, DenseNet201, and EfficientNet, are employed to analyze dermoscopic images of pigmented skin lesions. These CNN models are trained on the ISIC 2017 datasets containing benign and malignant (melanoma and seborrheic keratosis) lesions, allowing them to learn intricate patterns and features that distinguish between the two classes. In addition to deep learning techniques, traditional machine learning classifiers are also utilized in the classification of pigmented skin lesions. Random Forest, Support Vector Machine, K Nearest Neighbor, XGBoost, and Gradient Boosting Machines are among the commonly used classifiers. These models utilize various algorithms and techniques to create decision boundaries and make predictions based on the extracted features from the skin lesion images.

The ultimate goal of employing CAD systems and machine learning algorithms in the diagnosis of pigmented skin lesions is to detect malignant lesions as early as possible. Early detection plays a crucial role in the successful treatment and prognosis of skin cancer, especially melanoma, which can be life-threatening if left undetected or untreated. By accurately classifying skin lesions, CAD systems can aid dermatologists in making informed decisions regarding the need for further investigation or intervention, such as biopsies or surgical removal. The integration of CAD systems with the expertise of dermatologists holds tremendous potential in improving patient outcomes and healthcare delivery. By combining the capabilities of machine learning algorithms and the clinical expertise of dermatologists, CAD systems can provide a valuable second opinion, support differential diagnosis, and assist in the decision-making process. Additionally, the use of CAD systems in tele dermatology and remote consultations allows for enhanced access to specialized dermatological care, particularly for individuals in remote areas or underserved communities [5].

In conclusion, the application of CAD systems and machine learning algorithms in the classification of pigmented skin lesions represents a significant advancement in dermatology. These technologies have the potential to improve diagnostic accuracy, facilitate early detection of malignant lesions, enhance communication and collaboration between healthcare professionals, and ultimately contribute to better patient outcomes in the fight against skin cancer. Continued research and development in this field will further refine and optimize CAD systems, ultimately benefiting patients and healthcare providers alike.

2. Methodology

This section explains the ISIC 2017 database, the distribution of the dataset, data handling and balancing methods, CAD classification architecture based on hybrid approach and transfer learning approach, and the evaluation metrics for classification.

2.1. Dataset Description:

The original International Skin Image Collaboration (ISIC) 2017 dataset was used for the project. The ISIC 2017 skin lesion dataset consists of 2000 training, 160 validation, and 600 test images representing various types of skin lesions, including benign, melanoma, and seborrheic keratosis lesions. The ISIC 2017 dataset has been widely used for training and evaluating machine learning and deep learning models in skin classification tasks. The ISIC 2017 has contributed to advancements in automated skin diagnosis, benchmarking algorithms, and fostering collaboration among researchers for improving the efficiency of early detection and treatment of skin cancer [6].

Table 1. ISIC 2017 Distribution

Lesion Types	Number of Images		
	Train	Validation	Test
Benign	1372	78	393
Melanoma	374	30	117
Seborrheic keratosis	254	42	90

2.2. Data Preprocessing:

The ISIC 2017 dataset images have a high resolution and different sizes (from 540x722 to 4000x6000 pixels). Here you can see a few image samples from the dataset.

Figure 1. ISIC 2017 Image Samples

This requires preprocessing the images before feeding them to the network. For each image the following preprocessing pipeline is applied:

- Resizing the images to 128x128.
- Normalization of dataset images. We divided each pixel value of images by 255. By dividing by 255, the pixel values are scaled to the range of 0 to 1. This is a common technique for normalizing pixel values in images, as it brings them to a standardized range.

2.2. Data Balancing:

After analyzing the ISIC 2017 dataset, we can clearly see that the training set is highly imbalanced. The number of benign cases is relatively higher as compared to the number of melanoma and seborrheic keratosis cases. To deal with the class imbalance, we downsampled the benign cases in the training set and upsampled the melanoma and seborrheic keratosis cases. The numpy library was used to perform data upsampling and downsampling on the training set.

Since we are dealing with the two-step hierarchal classification, data handling has been implemented for Step 1 and Step 2 of the classification hierarchy.

(i) The first step of the classification hierarchy considered the binary classification between the Benign Class and the Other class. The Other class has been created by combining the cases of both melanoma and seborrheic keratosis.

(ii) The second step of the classification hierarchy considered the binary classification between the Melanoma class and the Seborrheic keratosis class.

To perform the first step of the hierarchal classification, we have downsampled the Benign cases to 1000 samples from 1372 samples. Others class consisting of melanoma and seborrheic keratosis cases have been upsampled to 1000 samples from 628 samples.

To perform the second step of the hierarchal classification, we have removed the benign cases from the training set in order to do the melanoma vs. seborrheic keratosis classification.

2.3. Proposed Computer Aided Diagnosis (CAD) System:

For our project, Keras Applications is used as a framework. Our CAD system is developed based on a two-step binary hierarchal classification. Several models based on Hybrid Architecture and Transfer Learning Techniques have been evaluated.

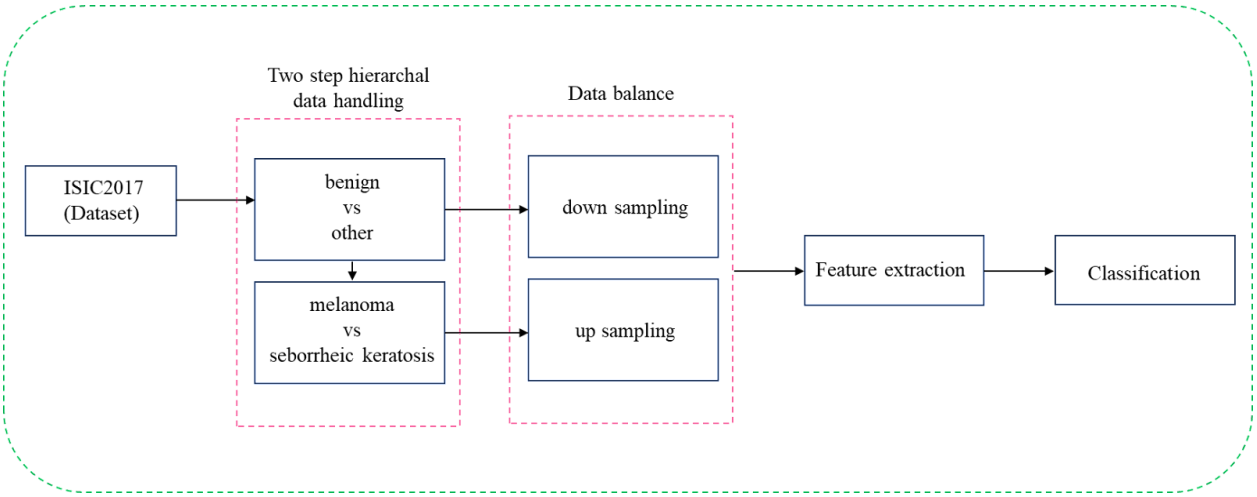

Figure 2. Proposed CAD system for Skin Lesion Classification

2.3.1. Transfer Learning Approach based on Deep Learning

Convolutional neural networks are at the core of most state-of-the-art computer vision solutions for a broad range of tasks. In this project, we implemented a Transfer Learning based approach using deep learning models for classifying skin lesions. Based on pre-trained Keras models of Tensorflow platform, we use the transfer learning to retrain the last few layers of the renowned CNN architectures for our classification problem. Due to the excellent performance of CNN models in the image classification competitions, improvements in the CNN architectures are very active. A series of CNN-based networks continue to appear, making CNN an irreplaceable mainstream method in the field of computer vision.

In transfer learning architecture, the pre-trained CNN models act as a feature extractor. The initial layers of the models, which learn low-level and generic features, are retained, while the later layers, which are more task-specific, are fine-tuned to classify the skin lesions.

2.3.1.1. Convolutional Neural Networks

A convolutional neural network is a network architecture for deep learning [7,8]. CNNs are deep artificial neural networks [9] that are primarily used to classify images cluster them by similarity and perform object recognition [9] within scenes. A CNN is comprised of one or more convolutional layers followed by one or more fully connected layers as in a standard multilayer neural network. It learns directly from images. CNN can be trained to do image analysis tasks including classification, object detection, segmentation, and image processing. CNNs are made of several types of layers, like Convolutional Layer, Non-Linearity Layer, Rectification Layer, Rectified Linear Units (ReLU), Pooling Layer, Fully Connected Layer and Dropout Layer.

The Visual Geometry Group of Oxford proposes the VGG network. The network uses a deeper network structure with depths of 11, 13, 16, and 19 layers. Meanwhile, VGG networks use a smaller convolution kernel (3×3 pixels) instead of the larger convolution kernel, which reduces the parameters and increases the expressive power of the networks [10].

ResNet solves the "degradation" problem of deep neural networks by introducing residual structure. ResNet networks use multiple parameter layers to learn the representation of residuals between input and output, rather than using parameter layers to directly try to learn the mapping between input and output as VGGs networks do. Residual networks are characterized by ease of optimization and the ability to improve accuracy by adding considerable depth [11].

The inception-V3 network is mainly improved in two aspects. Firstly, branch structure is used to optimize the Inception Module; secondly, the larger two-dimensional convolution kernel is unpacked into two one-dimensional convolution kernels. This asymmetric structure can deal with more and richer spatial information and reduce the computation [12].

The Inception-ResNet network is inspired by ResNet, which introduces the residual structure of ResNet in the Inception module. Adding the residual structure does not significantly improve the model effect. But the residual structure helps to speed up the convergence and improve the calculation efficiency. The calculation amount of Inception-ResNet-v1 is the same as that of Inception-V3, but the convergence speed is faster [13].

Xception is an improvement of Inception-V3. The network proposes a novel Depthwise Separable Convolution align them in column, the core idea of which lies in space transformation and channel transformation. Compared with Inception, Xception has fewer parameters and is faster [14].

The DenseNet network is inspired by the ResNet network. DenseNet uses a dense connection mechanism to connect all layers. This connection method allows the feature map learned by each layer to be directly transmitted to all subsequent layers as input, so that the features and the transmission of the gradient is more effective, and the network is easier to train. The network has the following advantages: it reduces the disappearance of gradients, strengthens the transfer of features, makes more effective use of features, and reduces the number of parameters to a certain extent [15].

EfficientNetV2L is an advanced deep learning architecture from the EfficientNet family. It focuses on depth-wise separable convolutions and uses compound scaling to balance model size and performance. It introduces improvements like SELU and scaled Swish activations. By scaling width, depth, and resolution, it achieves state-of-the-art results in computer vision tasks while considering resource constraints. EfficientNetV2L is a compact and efficient solution for various deep learning applications [16].

ConvNeXt Xlarge is a deep learning architecture that enhances convolutional neural networks by introducing grouped convolutions. It divides input channels into groups and performs convolutions separately on each group. This approach increases model capacity without significantly increasing computational cost. ConvNeXt Xlarge achieves state-of-the-art results on various computer vision tasks by effectively capturing both local and global features. It is a scalable and efficient architecture that offers improved performance in deep learning applications [17].

2.3.1.2. Experimental Environment for the Training

The comparative experiments are performed on the local computer. The computer hardware configuration is shown in Table 2. The computer software configuration is as follows: Windows 11 Professional operating system, Python 3.6, and Jupyter Notebook.

Table 3. Computer hardware configuration

Hardware	Product Name
CPU	Intel(R) Core (TM) i7-1185G7
GPU	Intel Iris Xe
RAM	16 GB

The experiment mainly uses deep learning models and some relatively novel deep learning models. The hyperparameters uniformly set by these models are shown in Table 4.

Table 4. Deep Learning Model Parameters

Parameters	Value
Batch Size	16
Epochs	10
Learning Rate	0.001, 0.005 [varies]
Optimizers	Adam, SGD

(i) The goal of the optimizer is to update the parameters of a model iteratively in order to minimize the loss function and improve the model's performance.

(ii) This learning rate parameter specifies the learning rate for the optimizer. The learning rate determines the step size taken during optimization. It controls how much the weights of the model are updated based on the calculated gradients.

(iii) Batch size refers to the number of training examples or data points that are processed together in a single forward and backward pass during the training of a neural network. It is one of the hyperparameters that need to be defined before training a model. During each iteration of the training process, the batch size determines the number of samples that are propagated through the network and used to compute the gradient for updating the model's parameters.

(iv) In deep learning, an epoch refers to a complete iteration over the entire training dataset during the model training process. In other words, it represents the number of times the model has seen and processed the entire training dataset.

In addition, we have used the binary cross entropy loss function for our classification purpose. It measures the dissimilarity between the predicted probability distribution and the true binary labels. The formula can be expressed as:

$$Binary\ Cross\ Entropy = -\ (y * log\ \sigma\ (p) + (1 - y) * log(1 - \sigma\ (p))$$

where p is the prediction of the model, y is the ground and σ is the sigmoid function.

2.3.2. Hybrid Approach based on Deep Learning and Machine Learning

In hybrid architecture, different Deep Learning models have been used for feature extraction stage, while the machine learning models serve as the classifier in the subsequent stage. The following ML classifiers were trained: Random Forest (RF), Naïve Bayes (NB), Support Vector Machine (SVM), Logistic Regression (LR), Decision Tree (DT), K-Nearest Neighbors (KNN), Gradient Boosting Machine (GBM), XG-Boost (XGB).

(i) Random Forest

The Random Forest Classifier is a popular ensemble learning algorithm that combines multiple decision trees to make predictions.

Its best parameters are found through the grid search method among the different options.

- n_estimators parameter specifies the number of decision trees in the random forest ensemble. In our case, we are using 100 trees.
- random_state parameter sets the random seed for reproducibility, ensuring that the same random numbers are generated each time the code runs.

(ii) Naïve Bayes Classifier

One of many ways to solve the skin lesion classification problem is by using Naïve Bayes. Naïve Bayes uses Bayes Theorem, which for our classification problem, gives us:

$$P(label \mid features) = \frac{P\ (label)\ x\ P(features\ |label)}{P\ (features)}$$

(iii) Support Vector Machine Classifier

This algorithm works on a simple strategy of separating hyperplanes. Given training data, the algorithm categorizes the test data into an optimal hyperplane. The data points are plotted in a n-dimension vector space (n depends upon the features of the data points). SVM algorithm is used for binary classification and regression tasks but in our case, we have a 2-step hierarchical binary classification. We adopt the pairwise classification technique where each pair of classes will have one SVM classifier trained to separate the classes. The overall accuracy of this classifier will be the accuracies of every SVM classification included. Then on performing classification we find a BMA that defines the overall performance of the model on all the classes very well.

(iv) Logistic Regression Classifier

This algorithm was named after the core function used in it that is the logistic function. The logistic function is also known as the sigmoid function. It is an S-shaped curve that takes real values as input and converts it into a range between 0 and 1. The sigmoid function is defined as follows:

$$S(x) = \frac{1}{1 + e^{-x}} = \frac{e^{x}}{e^{x} + 1}$$

(v) Decision Tree Classifier

The Decision tree classification approach is a technique that makes classification predictions by carrying out a series of true or false decisions. The decision tree approach is the foundation for the implementation of Random Forest. A Decision Tree is a flowchart-like tree structure, in which each internal node represents a test on an attribute and each branch represents an outcome of the test, and each leaf node represents a class.

(vi) K Nearest Neighbors Classifier

KNN Classifier is an instance-based learner used for both classification and regression tasks. This algorithm does not use the training data to make any generalizations. It is based on feature similarity. A test sample is classified based on a majority vote of its neighbors; the class assigned to the test sample is the most common class among k nearest neighbors. When used for regression the output value is the average of the outputs of its k nearest neighbors. This classifier is a lazy learner because nothing is done with the training data until the model tries to classify the test data. We have taken the k value to be 3 which gave us the most accurate result. The k value must not be too large in that it includes the noise points or points that belong to the neighboring class.

(vii) Gradient Boosting Machine Classifier

The Gradient Boosting Machine (GBM) algorithm works by iteratively building an ensemble of weak learners, usually decision trees, to correct errors made by the previous models. It optimizes the model by training each weak learner to predict the negative gradient of the loss function with respect to the residuals. The algorithm updates the model's prediction by combining the predictions of all weak learners. GBM is capable of capturing complex patterns and delivering accurate predictions, but it requires tuning of hyperparameters to prevent overfitting.

(viii) XGBoost Classifier

XGBoost, short for Extreme Gradient Boosting, is a highly efficient implementation of the Gradient Boosting Machine (GBM) algorithm. It works by iteratively adding weak learners (decision trees) to form a strong ensemble model. XGBoost improves upon traditional GBM by incorporating regularization techniques, gradient-based optimization, and efficient parallel processing. It handles missing values, performs tree pruning for better generalization, and provides feature importance analysis. With its excellent performance and scalability, XGBoost has become a popular choice for various machine learning tasks, achieving high accuracy in competitions and real-world applications.

2.4. Evaluation Metrices

To scientifically evaluate the classification performance of our proposed model, choosing appropriate indicators is a crucial factor. We have used the following metrices given below: -

(i) Balanced Multiclass Accuracy (BMA)

$$BMA = \frac{1}{M}\sum_{m=1}^{M}\frac{tp_m}{n_m}$$

Here M is number of classes (3 in our case), tp_{m} is number of true positives of class m, and n_m is number of samples of class m.

(ii) Accuracy

The accuracy is the proportion of the total number of predictions that were correct. Accuracy is calculated as the number of all correct predictions divided by the total number of the dataset. It is determined using the equation:

$$Accuracy = \frac{TP+TN}{TP+FP+FN+TN}$$

(iii) Sensitivity

The true positive rate (TPR) or Sensitivity is the proportion of positive cases that were correctly identified. TPR or Recall is calculated as the number of correct positive predictions divided by the total number of positives, as calculated using the equation:

$$Sensitivity = \frac{TP}{TP+FN}$$

(iv) Specificity

Specificity, also known as the True Negative Rate (TNR), is a metric used in binary classification to measure the proportion of actual negative cases that are correctly identified as negative. It quantifies the model's ability to correctly classify negative samples. Specificity is calculated as the number of true negative predictions divided by the total number of negatives, expressed using the equation:

$$Specificity = \frac{TN}{TN + FP}$$

3. Results and Discussion

The evaluation of the research is based on the BMA, accuracy, sensitivity, and specificity. Various machine learning and deep learning techniques have been implemented in order to find the best-fit algorithms for the system. Each model was tested on the 600 images of the ISIC2017 test set. The results are summarized in the table.

Table 4. Performance details of different deep learning models based on Transfer Learning approach.

Model	**Optimizer**	**Learning Rate**	**Benign vs Other**			**Melanoma vs Seborrheic Keratosis**			**BMA**
			Accuracy	*Sensitivity*	*Specificity*	*Accuracy*	*Sensitivity*	*Specificity*	
VGG16	Adam	0.005	0.72	0.90	0.38	0.70	0.71	0.68	**0.76**
	SGD	0.005	0.72	0.83	0.52	0.70	0.66	0.76	0.75
ResNet50	Adam	0.005	0.64	0.76	0.40	0.62	0.51	0.76	0.64
	SGD	0.001	0.48	0.23	0.97	0.58	0.97	0.07	0.42
Inception V3	Adam	0.001	0.66	0.64	0.71	0.71	0.80	0.60	0.68
	SGD	0.001	0.70	0.72	0.66	0.69	0.79	0.58	0.69
InceptionResNetv2	Adam	0.005	**0.77**	0.83	0.64	0.67	0.53	0.86	0.74
	SGD	0.005	**0.77**	0.79	0.75	0.74	0.75	0.73	**0.76**
Xception	Adam	0.001	0.73	0.80	0.58	0.72	0.70	0.74	0.75
	SGD	0.001	0.72	0.73	0.69	**0.75**	0.83	0.66	0.74
DenseNet201	Adam	0.005	0.69	0.75	0.58	**0.75**	0.71	0.82	**0.76**
	SGD	0.001	0.66	0.62	0.73	0.70	0.76	0.62	0.67
EfficientNetV2L	Adam	0.005	0.64	0.94	0.07	0.56	1.00	0.00	0.65
	SGD	0.005	0.36	0.03	0.99	0.57	1.00	0.00	0.34
ConvNeXtXLarge	Adam	0.005	0.62	0.73	0.45	0.62	0.68	0.70	0.66
	SGD	0.005	0.65	0.71	0.53	0.65	0.71	0.62	0.63

Performance Analysis of Transfer Learning Models:

In the case of BMA, the highest score 0.76 was obtained by the VGG16 with Adam, InceptionResNetV2 with SGD, and DenseNet201 with Adam models. However, Xception with both Adam and SGD, InceptionResNetV2 with Adam, and VGG16 with SGD scored 0.75, 0.74, and 0.75 respectively, which is very close to the maximum performance. On the other hand, ResNet50 with SGD and EfficientNetV2L with SGD provided the minimum score of 0.42 and 0.34 respectively. The performance of Inception V3 and ConvNeXtXLarge were average with both Adam and SGD optimizer and obtained the BMA scores of 0.68, 0.69, 0.66 and 0.63 respectively.

In the case of benign and other classes, InceptionResNetv2 with both Adam and SGD optimizer provided the highest accuracy score of 0.77 whereas the accuracy scores obtained by the ResNet50 with SGD and EfficientNetV2L with SGD were 0.48 and 0.36, which is the minimum.

In the case of melanoma and seborrheic keratosis, Xception with SGD and DenseNet201with Adam provided the highest accuracy score of 0.75 whereas the accuracy scores obtained by the EfficientNetV2L with SGD and Adam were 0.56 and 0.57, which is the minimum.

It is clear from the above table that VGG16, InceptionResNetv2, and DenseNet201 outperformed other models.

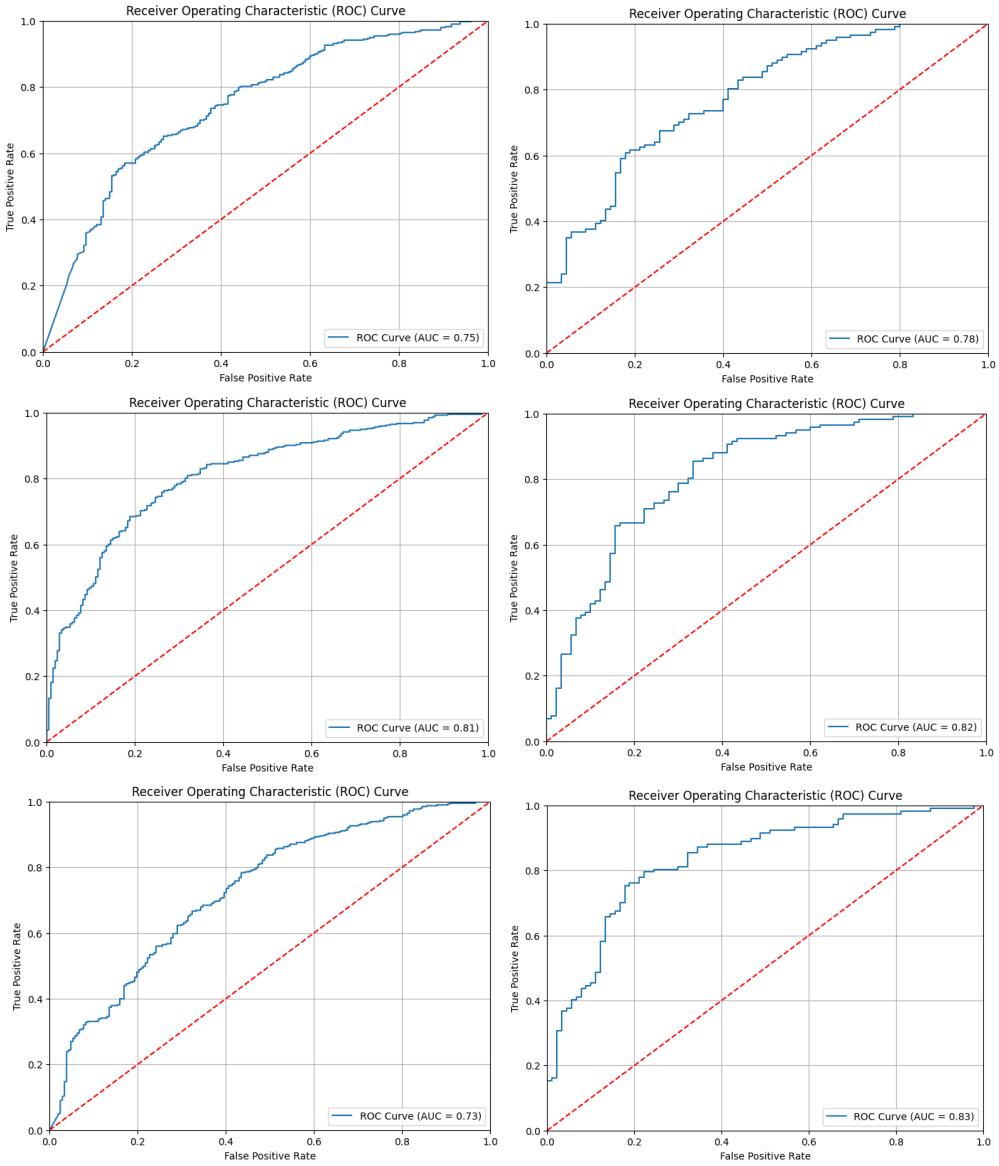

Figure 3. Area under the ROC Curves for benign vs others class and melanoma vs seborrheic keratosis class of the best Transfer Learning models (VGG16, InceptionResNetv2, DenseNet201).

Table 5. Performance details of different hybrid models based on deep learning and machine learning classifiers.

Feature Extractor	**Classification Model**	**Benign vs Other**			**Melanoma vs Seborrheic Keratosis**			**BMA**
		Accuracy	*Sensitivity*	*Specificity*	*Accuracy*	*Sensitivity*	*Specificity*	
VGG16	Random Forest	0.70	0.79	0.53	0.73	0.64	0.84	**0.76**
	Naïve Bayes	0.44	0.35	0.62	0.58	0.41	0.80	0.52
	Support Vector Machine	0.67	0.73	0.61	0.65	0.61	0.71	0.68
	Logistic Regression	0.70	0.72	0.64	0.68	0.60	0.79	0.70
	Decision Tree	0.63	0.69	0.51	0.55	0.57	0.52	0.59
	K Nearest Neighbors	0.62	0.49	0.86	0.68	0.62	0.77	0.62
	Gradient Boosting Machines	0.73	0.76	0.71	0.74	0.64	0.88	**0.76**
	XGBoost	0.71	0.78	0.56	0.55	0.57	0.52	0.63
ResNet50	Random Forest	0.68	0.83	0.39	0.64	0.68	0.59	**0.70**
	Naïve Bayes	0.39	0.10	0.94	0.50	0.14	0.98	0.40
	Support Vector Machine	0.64	0.59	0.72	0.67	0.56	0.80	0.65
	Logistic Regression	0.62	0.54	0.76	0.64	0.62	0.68	0.61

	Decision Tree	0.62	0.69	0.50	0.63	0.64	0.61	0.65
	K Nearest Neighbors	0.60	0.54	0.71	0.63	0.71	0.52	0.59
	Gradient Boosting Machines	0.65	0.66	0.63	0.61	0.60	0.62	0.63
	XGBoost	0.70	0.76	0.58	0.63	0.64	0.61	0.67
Inception V3	Random Forest	0.70	0.85	0.42	0.75	0.78	0.71	**0.78**
	Naïve Bayes	0,60	0.51	0.79	0.64	0.64	0.64	0.60
	Support Vector Machine	0.67	0.69	0.62	0.67	0.61	0.76	0.68
	Logistic Regression	0.68	0.70	0.65	0.70	0.66	0.74	0.70
	Decision Tree	0.61	0.71	0.43	0.60	0.57	0.63	0.64
	K Nearest Neighbors	0.59	0.50	0.75	0.65	0.82	0.43	0.59
	Gradient Boosting Machines	0.72	0.75	0.66	0.72	0.74	0.70	0.73
	XGBoost	0.72	0.79	0.58	0.60	0.57	0.63	0.66
InceptionResNetv2	Random Forest	0.72	0.82	0.53	0.73	0.74	0.71	**0.76**
	Naïve Bayes	0.57	0.44	0.81	0.71	0.66	0.79	0.63
	Support Vector Machine	0.73	0.76	0.68	0.67	0.66	0.68	0.70
	Logistic Regression	0.75	0.77	0.71	0.71	0.70	0.71	0.73
	Decision Tree	0.63	0.70	0.51	0.71	0.68	0.76	0.71
	K Nearest Neighbors	0.58	0.48	0.77	0.69	0.79	0.55	0.61
	Gradient Boosting Machines	0.71	0.73	0.68	0.75	0.71	0.80	0.75
	XGBoost	0.73	0.78	0.63	0.71	0.68	0.76	0.74
Xception	Random Forest	0.70	0.82	0.47	0.76	0.74	0.80	**0.79**
	Naïve Bayes	0.56	0.48	0.72	0.71	0.74	0.69	0.63
	Support Vector Machine	0.68	0.73	0.59	0.73	0.72	0.74	0.74
	Logistic Regression	0.70	0.75	0.62	0.72	0.70	0.74	0.73
	Decision Tree	0.60	0.65	0.50	0.70	0.66	0.76	0.69
	K Nearest Neighbors	0.56	0.46	0.75	0.70	0.82	0.54	0.60
	Gradient Boosting Machines	0.70	0.72	0.67	0.72	0.68	0.77	0.72
	XGBoost	0.70	0.77	0.57	0.70	0.66	0.76	0.73
DenseNet201	Random Forest	**0.76**	0.79	0.69	0.72	0.66	0.80	**0.75**
	Naïve Bayes	0.62	0.50	0.86	0.64	0.46	0.87	0.61
	Support Vector Machine	0.69	0.69	0.67	0.74	0.69	0.80	0.73
	Logistic Regression	0.69	0.69	0.68	0.74	0.70	0.80	0.73
	Decision Tree	0.62	0.65	0.55	0.67	0.64	0.71	0.67
	K Nearest Neighbors	0.68	0.63	0.76	0.75	0.87	0.59	0.70
	Gradient Boosting Machines	0.70	0.66	0.76	**0.78**	0.72	0.87	**0.75**
	XGBoost	0.73	0.76	0.69	0.67	0.64	0.71	0.70
EfficientNetV2L	Random Forest	0.61	0.78	0.30	0.61	0.67	0.53	**0.66**
	Naïve Bayes	0.56	0.48	0.72	0.52	0.54	0.50	0.53
	Support Vector Machine	0.55	0.55	0.56	0.58	0.63	0.52	0.57
	Logistic Regression	0.53	0.51	0.56	0.60	0.63	0.56	0.57
	Decision Tree	0.55	0.64	0.38	0.59	0.65	0.51	0.60
	K Nearest Neighbors	0.47	0.39	0.61	0.52	0.71	0.28	0.46
	Gradient Boosting Machines	0.55	0.62	0.42	0.60	0.68	0.50	0.60
	XGBoost	0.61	0.79	0.27	0.59	0.65	0.71	0.65
ConvNeXtXLarge	Random Forest	0.69	0.83	0.43	0.65	0.69	0.58	**0.70**
	Naïve Bayes	0.46	0.30	0.76	0.49	0.28	0.77	0.45
	Support Vector Machine	0.59	0.55	0.56	0.62	0.56	0.70	0.60
	Logistic Regression	0.60	0.56	0.67	0.63	0.66	0.59	0.60
	Decision Tree	0.57	0.64	0.44	0.57	0.60	0.53	0.59

	K Nearest Neighbors	0.58	0.50	0.73	0.58	0.66	0.48	0.55
	Gradient Boosting Machines	0.68	0.75	0.55	0.66	0.63	0.69	0.69
	XGBoost	0.70	0.79	0.54	0.57	0.60	0.53	0.64

Performance Analysis of Hybrid Models:

In the case of BMA, the highest score 0.79 was obtained by the Xception – Support Vector Machine and Xception – Gradient Boosting Machines hybrid models. However, InceptionV3 – Random Forest hybrid model scored 0.78, which is very close to the maximum performance. In addition, VGG16 – Random Forest, InceptionResNetv2 – Random Forest, and DenseNet201–Random Forest provided scores of 0.76, 0.75, and 0.75.

In the case of benign and other classes, the highest accuracy score of 0.76 was obtained by the DenseNet201–Random Forest hybrid model whereas in the case of melanoma and seborrheic keratosis, the best accuracy score of 0.78 was obtained by the DenseNet201– Gradient Boosting Machines hybrid model.

It is clear from the above table that the Xception – Support Vector Machine hybrid model outperformed other models. In addition, the table 4 and 5 show that the hybrid model performed more better than the transfer learning model. The height BMA score obtained by the hybrid model was 0.79 and the maximum BMA score obtained by the transfer learning model is 0.76.

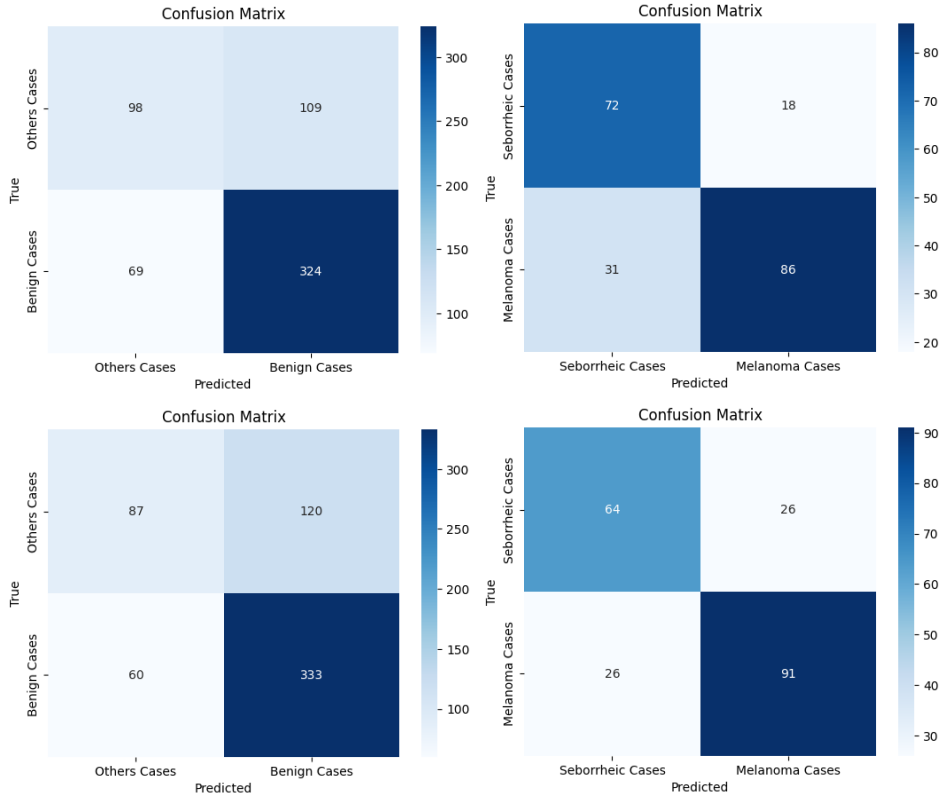

Figure 4. Confusion matrix for benign vs others class and melanoma vs seborrheic keratosis class of the best hybrid models (Xception-Random Forest, InceptionV3-Random Forest).

6. Conclusion

In this project, the tasks of skin lesion classification were addressed using two different approaches: Transfer Learning method followed by the Hybrid Method. The classification task was challenging for different reasons such as the different artifacts present in dermoscopy images, high resolution and the heterogeneity of the lesions. However, the results show a potential of improvement in this task especially using hybrid technique. Possible future work includes more optimization and preprocessing in the feature engineering step of machine learning, further fine tuning of the model parameters in deep learning. It is clear that some models performed outstanding with benign and others classes and some models performed better with melanoma and seborrheic keratosis classes. Therefore, a new model with the combination of the best performed model with benign and others classes and melanoma and seborrheic keratosis classed can be proposed in future.

7. Acknowledgement

We would like to express our gratitude to our supervisor Professor Alessandra Scotto di Freca for her invaluable support, guidance, and contributions to this project. Her assistance has been instrumental in its completion.

References

[1] Shetty, B., Fernandes, R., Rodrigues, A. P., Chengoden, R., Bhattacharya, S., & Lakshmanna, K. (2022). Skin lesion classification of dermoscopic images using machine learning and convolutional neural network. Scientific Reports, 12(1), 1-11. https://doi.org/10.1038/s41598-022-22644-9.

[2] D, S. R., & A, S. (2019). Deep Learning Based Skin Lesion Segmentation and Classification of Melanoma Using Support Vector Machine (SVM). Asian Pacific Journal of Cancer Prevention : APJCP, 20(5), 1555-1561. https://doi.org/10.31557/APJCP.2019.20.5.1555.

[3] Doi, K. (2007). Computer-Aided Diagnosis in Medical Imaging: Historical Review, Current Status and Future Potential. Computerized medical imaging and graphics : The official journal of the Computerized Medical Imaging Society, 31(4-5), 198. https://doi.org/10.1016/j.compmedimag.2007.02.002.

[4] Kassem, M. A., Hosny, K. M., Damaševičius, R., & Eltoukhy, M. M. (2021). Machine Learning and Deep Learning Methods for Skin Lesion Classification and Diagnosis: A Systematic Review. Diagnostics, 11(8). https://doi.org/10.3390/diagnostics11081390.

[5] Chang, Y., Huang, A., Yang, Y., Lee, H., Chen, C., Wu, Y., & Chen, S. (2013). Computer-Aided Diagnosis of Skin Lesions Using Conventional Digital Photography: A Reliability and Feasibility Study. PLoS ONE, 8(11). https://doi.org/10.1371/journal.pone.0076212.

[6] Codella, N. C., Gutman, D., Celebi, M. E., Helba, B., Marchetti, M. A., Dusza, S. W., Kalloo, A., Liopyris, K., Mishra, N., Kittler, H., & Halpern, A. (2017). Skin Lesion Analysis Toward Melanoma Detection: A Challenge at the 2017 International Symposium on Biomedical Imaging (ISBI), Hosted by the International Skin Imaging Collaboration (ISIC). https://doi.org/10.48550/arXiv.1710.05006.

[7] Zeiler, M. D., Taylor, G. W., & Fergus, R. (2011, November). Adaptive deconvolutional networks for mid and high level feature learning. In Computer Vision (ICCV), 2011 IEEE International Conference on (pp. 2018-2025). IEEE.

[8] Christian Szegedy, Wei Liu, et aI, Going Deeper with Convolutions, arXiv: 1409.4842,2014.

[9] Zeiler, M. D., & Fergus, R. (2013). Visualizing and Understanding Convolutional Neural Networks, arXiv: 1311.2901

[10] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556, 2014.

[11] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 770– 778, 2016.

[12] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 2818–2826, 2016.

[13] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alexander Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. In Proceedings of the AAAI Conference on Artificial Intelligence, volume 31, 2017.

[14] François Chollet. Xception: Deep learning with depthwise separable convolutions. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 1251–1258, 2017.

[15] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 4700–4708, 2017.

[16] Tan, M., & Le, Q. V. (2021). EfficientNetV2: Smaller Models and Faster Training. ArXiv. /abs/2104.00298

[17] Liu, Z., Mao, H., Wu, C., Feichtenhofer, C., Darrell, T., & Xie, S. (2022). A ConvNet for the 2020s. ArXiv. /abs/2201.0354