DEEP CNN MODEL FOR MULTI-CLASS CLASSIFICATION OF SKIN CANCER FROM DERMOSCOPIC IMAGES

P. Vemulamma[1*], Kande Tharun[2], Kusuma Sunanda[2], Lavudiya Shirisha[2]

[1]Assistant Professor, [2]UG Student, [1,2]Department of Computer Sciences and Engineering,

[1,2]Vaagdevi College of Engineering (UGC - Autonomous), Bollikunta, Warangal, Telangana.

[*]Corresponding author: P. Vemulamma (vemulamma@vaagdevi.edu.in)

ABSTRACT

Skin cancer, one of the most common and potentially fatal cancers, necessitates early and accurate diagnosis. Traditional diagnostic methods, although effective, are often slow, costly, and prone to human error. This project leverages advanced machine learning techniques to address these issues. Specifically, it uses the InceptionResNetV2 model for feature extraction and the Extra Trees Classifier (ETC) for multi-class classification of skin cancer, aiming to create an automated and efficient diagnostic tool for dermatologists. Traditional diagnosis of skin cancer involves a dermatologist's visual examination, followed by a biopsy and histopathological analysis. These methods, while reliable, are time-consuming, expensive, and heavily dependent on skilled dermatologists, which can be a major limitation in remote areas. This Problem statement is extracting discriminative features from skin lesion images, addressing data imbalance where some cancer types are underrepresented, and ensuring high classification accuracy. Effective feature extraction captures essential lesion characteristics, while data balancing techniques prevent model bias. The motivation is to Automated classification systems can significantly aid dermatologists by providing preliminary diagnostic results, especially in areas with limited medical expertise. Early and accurate diagnosis can lead to better treatment outcomes and save lives. The system integrates InceptionResNetV2 for feature extraction and Extra Trees Classifier for classification. Users upload categorized skin lesion images, which are preprocessed for the deep learning model. InceptionResNetV2, pre-trained on ImageNet, extracts robust features from these images. To address data imbalance, SMOTE generates synthetic samples for underrepresented classes, ensuring a balanced training set. The Extra Trees Classifier is trained on the training set's feature vectors and compared with a K-Nearest Neighbors (KNN) classifier. The trained model predicts the class of new skin lesion images, displaying the label directly on the image.

Keywords: Skin Cancer Diagnosis, InceptionResNetV2, Extra Trees Classifier (ETC), Machine Learning, Image Classification

1. INTRODUCTION

The journey of skin cancer diagnosis has evolved significantly over the past few decades. Historically, the primary method of diagnosing skin cancer was through visual examination by dermatologists. [1] This traditional approach relied heavily on the expertise and experience of medical professionals, who would assess the physical appearance of skin lesions to identify potentially malignant growths. [2] This initial step was often followed by a biopsy, where a sample of the suspicious skin tissue would be removed and examined under a microscope by a pathologist. [3] The pathologist's role was crucial in confirming the presence of cancer cells and determining the type and stage of the cancer. [4] In the early days, the diagnosis of skin cancer was a lengthy and labor-intensive process. [5] The reliance on manual examination and subjective judgment led to variations in diagnostic accuracy. Furthermore, the requirement for specialized medical professionals and laboratory facilities made it difficult to provide timely and accurate diagnoses, especially in remote and underserved areas. [6] These challenges highlighted the need for more efficient and standardized diagnostic methods.

The advent of digital imaging and advancements in medical technology brought significant changes to the field of dermatology. [7] The introduction of dermatoscopes, which are handheld devices equipped with magnifying lenses and light sources, allowed dermatologists to examine skin lesions with greater detail. Dermatoscopy improved the ability to distinguish between benign and malignant lesions, thereby increasing the accuracy of initial diagnoses. However, despite these technological advancements, the diagnostic process remained largely manual and dependent on the skill and experience of the dermatologist. [8] The rapid development of computer technology and the emergence of machine learning and artificial intelligence (AI) opened new possibilities for skin cancer diagnosis. Researchers began exploring the potential of using digital image analysis and machine learning algorithms to automate the diagnostic process. The goal was to develop systems that could analyze skin lesion images and accurately classify them as benign or malignant. Early attempts in this field involved basic image processing techniques and traditional machine learning methods, such as support vector machines and decision trees.

With the advent of deep learning, a subset of machine learning that involves neural networks with many layers, the accuracy and capabilities of automated diagnostic systems improved dramatically. [9] Convolutional neural networks (CNNs), in particular, proved to be highly effective in analyzing medical images. Researchers developed and trained CNNs on large datasets of skin lesion images to recognize patterns and features indicative of different types of skin cancer. These advancements laid the foundation for the development of more sophisticated diagnostic tools. InceptionResNetV2, a state-of-the-art deep learning model, represents one of the significant breakthroughs in this field. [10] It combines the strengths of the Inception architecture, which allows the model to capture features at multiple scales, with the residual connections of ResNet, which help mitigate the vanishing gradient problem in deep networks. InceptionResNetV2, pre-trained on the ImageNet dataset, has demonstrated remarkable accuracy in various image classification tasks, making it an ideal choice for feature extraction in skin cancer diagnosis. The integration of InceptionResNetV2 with the Extra Trees Classifier (ETC), a robust ensemble learning method, further enhances the system's ability to classify skin cancer accurately. [12] The Extra Trees Classifier, known for its high performance and ability to handle large datasets, provides a powerful tool for the final classification of extracted features. Together, these technologies form the backbone of a modern, automated skin cancer diagnostic system that aims to overcome the limitations of traditional methods and provide timely, accurate, and accessible diagnoses.

2. LITERATURE SURVEY

Chaturvedi et al. [16] presented an automated multi-class skin classification (MCSC) approach using five pre-trained Convolutional Neural Networks (CNNs) and four ensemble methods on the HAM10000 dataset. The individual CNNs achieved an accuracy of 93%, while the ensemble methods achieved 92.8%. Among these, ResNeXt101 was identified as the most suitable model due to its optimized structure and superior accuracy. Ali et al. [17] proposed an enhanced skin cancer classification model using Deep Convolutional Neural Networks (DCNN) with transfer learning. This model incorporated filtering for noise removal, image normalization, feature extraction, and data augmentation, achieving a high accuracy of 93.1% on the HAM10000 dataset. Sedighi et al. [18] utilized a CNN model on the ISIC dataset to detect skin cancer. To address the lack of training data, they employed Generative Adversarial Networks (GANs) to create synthetic images. The model's accuracy improved from 53% to 71% with the inclusion of synthetic images.

Garg et al. [19] developed a decision support model for skin cancer detection and classification using the HAM10000 dataset. The model enhanced image resolution, utilized image augmentation, and employed transfer learning, resulting in F1 and precision scores of 0.77 and 0.88, respectively. Kumar

et al. [20] introduced a differential evolution-based Artificial Neural Network (ANN) model for skin cancer detection on the PH2 and HAM10000 datasets. This model used various filters for image enhancement and extracted color and texture features, achieving an accuracy of 97.4% and sensitivity of 90%. Mijwil [21] evaluated three deep learning models—VGG19, ResNet, and InceptionNet—on the ISIC dataset for classifying skin cancer as malignant or benign. The InceptionV3 model was found to be the best, with accuracies of 86.9% and precision of 87.4%. Zhang et al. [22] proposed a skin cancer diagnosis model using an optimized CNN and the Levy flight-based whale optimization algorithm to adjust CNN weights. Tested on the Dermquest dataset, this model effectively segmented lesions and identified those with artifacts.

Amin et al. [23] focused on localizing and classifying skin cancer using deep learning features. They resized images, segmented them using bi-orthogonal wavelet and Otsu methods, and combined VGG16 and AlexNet models. Features were selected using Principal Component Analysis (PCA) to classify benign and malignant lesions. Afza et al. [24] introduced a hybrid feature selection and classification approach using an Extreme Learning Machine (ELM) and whale optimization on the ISIC and HAM10000 datasets. Enhanced image features were selected and classified, showing effective results. Raju et al. [26] developed a clustering-based model for classifying various skin diseases like paederus, melanoma, benign lesions, herpes, and psoriasis. The model used a fuzzy-set model for lesion segmentation, extracted color, shape, and texture features, and employed a Support Vector Machine (SVM) with black widow optimization, achieving 92% accuracy on the ISIC-2018 dataset.

Karthik et al. [27] designed an EfficientNetV2 model with Efficient Channel Attention (ECA) for diagnosing skin cancer. It classified lesions such as melanoma, psoriasis, acne, and actinic keratosis, analyzing approximately 16 million variables and achieving an overall accuracy of 84.7%. Melbin et al. [28] developed an automatic skin disease classification model using diverse features and the Improved Grey Wolf Optimizer (IGWO). The model performed lesion segmentation using circular kernel and morphological operations, with features extracted by the ABCD method.

3. PROPOSED SYSTEM

The Project skin cancer classification system integrates several advanced machine learning and image processing techniques to classify skin cancer from images. Here's an overview of the system, detailing its components, workflow, and functionalities.

- **Dataset Upload:** The upload Dataset function allows the user to select a directory containing the dataset. The function reads the folder names within the selected directory, assuming each folder represents a different class of skin cancer. This directory structure is essential for organizing the data into classes, which the system later uses for training and classification.
- **Feature Extraction using InceptionResNetV2:**

  The InceptionResNetV2_feature_extraction function leverages the InceptionResNetV2 model, a pre-trained deep learning model known for its high performance in image classification tasks. This function processes the images by resizing them to 224x224 pixels, converting them to arrays, and extracting features using the InceptionResNetV2 model. These features, which capture essential image characteristics, are saved for further processing. If the features are already extracted and saved, the function loads them from the disk to save time.
- **Data Balancing with SMOTE:** The Data_balancing function addresses class imbalance issues using the Synthetic Minority Over-sampling Technique (SMOTE). SMOTE generates synthetic samples for the minority classes, thus balancing the dataset. This step is crucial for ensuring

that the machine learning models do not become biased towards the majority class. The function also visualizes the class distribution before and after applying SMOTE.

- **Train-Test Splitting:** The Train_test_spliting function splits the balanced dataset into training and testing sets. This division is necessary to evaluate the model's performance on unseen data. The function ensures that the split is random but reproducible by using a fixed random seed.
- **Model Training and Evaluation:** Two machine learning models are trained using the extracted features and the balanced dataset:
    - **K-Nearest Neighbors (KNN):** Implemented in the Model2_Base function, the KNN algorithm is a simple, yet effective classifier that assigns the class of the majority among the k-nearest neighbors of a data point.
    - **Extra Trees Classifier (ETC):** Implemented in the Model1_Base function, the ETC is an ensemble learning method that aggregates the results of multiple de-correlated decision trees to improve predictive performance.

  Both functions check if a pre-trained model already exists. If not, they train a new model and save it to disk. After training, the models' performances are evaluated using metrics such as accuracy, precision, recall, and F1-score. Confusion matrices and classification reports are generated and displayed for a comprehensive evaluation.
- **Prediction:** The predict function allows users to upload a test image and classify it using the trained model. The function preprocesses the image, extracts its features using the InceptionResNetV2 model, and uses the selected classifier (KNN or ETC) to predict the class of the image. The prediction result is displayed on the image using OpenCV, indicating the classified disease.

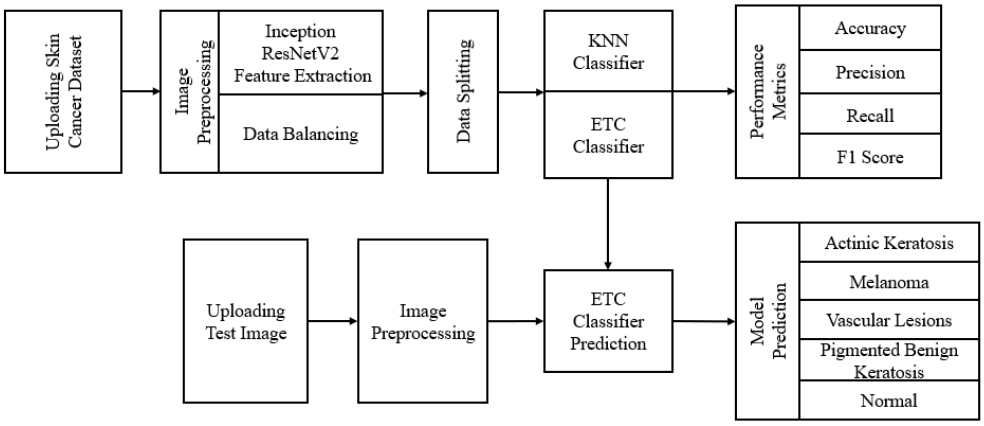

Fig. 1: Block diagram of the Proposed System.

3.1 Image preprocessing

Image preprocessing is a critical step in computer vision and image analysis tasks. It involves a series of operations to prepare raw images for further processing by algorithms or neural networks. Here's an explanation of each step-in image preprocessing:

Step 1. Image Read: The first step in image preprocessing is reading the raw image from a source, typically a file on disk. Images can be in various formats, such as JPEG, PNG, BMP, or others. Image reading is performed using libraries or functions specific to the chosen programming environment or framework. The result of this step is a digital representation of the image that can be manipulated programmatically.

Step 2. Image Resize: Image resize is a common preprocessing step, especially when working with machine learning models or deep neural networks. It involves changing the dimensions (width and height) of the image. Resizing can be necessary for several reasons:

- Ensuring uniform input size: Many machine learning models, especially convolutional neural networks (CNNs), require input images to have the same dimensions. Resizing allows you to standardize input sizes.
- Reducing computational complexity: Smaller images require fewer computations, which can be beneficial for faster training and inference.
- Managing memory constraints: In some cases, images need to be resized to fit within available memory constraints.

When resizing, it's essential to maintain the aspect ratio to prevent image distortion. Typically, libraries like OpenCV or Pillow provide convenient functions for resizing images.

Step 3. Image to Array: In this step, the image is converted into a numerical representation in the form of a multidimensional array or tensor. Each pixel in the image corresponds to a value in the array. The array is usually structured with dimensions representing height, width, and color channels (if applicable).

For grayscale images, the array is 2D, with each element representing the intensity of a pixel. For color images, it's a 3D or 4D array, with dimensions for height, width, color channels (e.g., Red, Green, Blue), and potentially batch size (if processing multiple images simultaneously).

The conversion from an image to an array allows for numerical manipulation and analysis, making it compatible with various data processing libraries and deep learning frameworks like NumPy or TensorFlow.

Step 4. Image to Float32: Most machine learning and computer vision algorithms expect input data to be in a specific data type, often 32-bit floating-point numbers (float32). Converting the image array to float32 ensures that the pixel values can represent a wide range of intensities between 0.0 (black) and 1.0 (white) or sometimes between -1.0 and 1.0, depending on the specific normalization used.

This step is essential for maintaining consistency in data types and enabling compatibility with various machine learning frameworks and libraries. It's typically performed by dividing the pixel values by the maximum intensity value (e.g., 255 for an 8-bit image) to scale them to the [0.0, 1.0] range.

Step 5. Image to Binary: Image binarization is a process of converting a grayscale image into a binary image, where each pixel is represented by either 0 (black) or 1 (white) based on a specified threshold. Binarization is commonly used for tasks like image segmentation, where you want to separate objects from the background.

The process involves setting a threshold value, and then for each pixel in the grayscale image, if the pixel value is greater than or equal to the threshold, it is set to 1; otherwise, it is set to 0.

Binarization simplifies the image and reduces it to essential information, which can be particularly useful in applications like character recognition or object tracking, where you need to isolate regions of interest.

3.2 Feature Extraction using InceptionResNetV2

InceptionResNetV2 is a deep learning model that combines the Inception architecture with residual connections. It was introduced by Christian Szegedy and his team at Google in their 2016 paper "Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning". This model is particularly designed for image recognition tasks and aims to enhance the performance of convolutional

neural networks (CNNs) by leveraging the strengths of both the Inception modules and Residual connections.

Working and Processing

The InceptionResNetV2 processes an input image by passing it through the initial convolution and pooling layers, which reduce the size and extract low-level features. It then sequentially processes the image through Inception-ResNet modules interspersed with reduction blocks, capturing complex patterns and hierarchical features at multiple scales. The residual connections ensure that the learning process remains efficient and effective, even as the network gets deeper.

Logic Behind InceptionResNetV2

The core logic behind InceptionResNetV2 is to combine the multi-scale feature extraction capabilities of the Inception modules with the efficient training enabled by residual connections. By doing so, the network can:

- Learn rich and diverse feature representations due to the parallel convolutional paths in Inception modules.
- Maintain efficient gradient flow and mitigate the vanishing gradient problem through residual connections, enabling the training of very deep networks.

This combination allows InceptionResNetV2 to achieve high performance on image recognition tasks, making it a powerful model for various applications in computer vision. The use of diverse filter sizes helps in capturing different aspects of the input image, while residual connections facilitate the training of deeper and more accurate networks

3.3 Extra Trees Classifier

The Extra Trees Classifier, short for Extremely Randomized Trees Classifier, is a powerful machine learning algorithm that belongs to the ensemble learning family, specifically the tree-based methods. It is an extension of the Random Forest algorithm and shares some similarities with it, but it introduces additional randomness in the tree-building process. In this detailed explanation, we'll delve into the inner workings of the Extra Trees Classifier, exploring its key components, algorithmic steps, and advantages.

Before diving into the specifics of the Extra Trees Classifier, it's essential to understand the concept of ensemble learning and decision trees.

Ensemble Learning: Ensemble learning involves combining multiple base learners (weak learners) to build a more robust and accurate predictive model. The fundamental idea is that by aggregating the predictions of individual learners, the ensemble model can often outperform any of its individual components.

Decision Trees: Decision trees are a popular type of machine learning algorithm used for both classification and regression tasks. They partition the feature space into regions and make predictions by traversing the tree from the root node to the leaf nodes, where each leaf node corresponds to a class label or a numerical value.

Random Forest and the Need for Extra Trees:

Random Forest is a widely used ensemble learning method based on decision trees. It builds multiple decision trees using bootstrapped samples of the training data and selects random subsets of features at each split point. While Random Forest is effective in reducing overfitting and improving prediction accuracy, it still involves some level of deterministic decision-making during the tree-growing process.

The Extra Trees Classifier addresses this limitation by introducing additional randomness, making it even more robust and less prone to overfitting.

Key Components of Extra Trees Classifier:

1. **Decision Trees:** Like Random Forest, the Extra Trees Classifier is based on the concept of decision trees. Each decision tree in the ensemble is constructed using a subset of the training data and a random subset of features at each split.
2. **Random Splitting:** In Extra Trees, the splitting of nodes is done randomly, without searching for the best possible split. Unlike Random Forest, which evaluates a subset of randomly chosen features for each split and selects the best one, Extra Trees randomly selects split points without considering feature importance.
3. **Aggregation:** The predictions of individual trees in the ensemble are aggregated to make the final prediction. For classification tasks, the class with the most votes (mode) among the predictions of all trees is chosen as the predicted class.

Algorithmic Steps:

Let's walk through the algorithmic steps involved in training an Extra Trees Classifier:

- **Bootstrap Sampling:** Like Random Forest, the training data is randomly sampled with replacement to create multiple bootstrap samples. Each bootstrap sample is used to train a separate decision tree in the ensemble.
- **Random Feature Subset:** At each split point in each decision tree, a random subset of features is selected. This subset is typically smaller than the total number of features in the dataset. The goal is to introduce diversity among the trees in the ensemble.
- **Random Splitting:** Instead of evaluating all features to find the best split, Extra Trees randomly selects split points without considering feature importance or optimizing impurity measures such as Gini impurity or information gain.
- **Tree Growing:** Each decision tree is grown to its maximum depth or until a stopping criterion is met (e.g., minimum number of samples per leaf node). The randomness in feature selection and splitting helps prevent overfitting and encourages diversity among the trees.
- **Aggregation:** During prediction, the class labels predicted by individual trees are aggregated using a majority voting scheme. For regression tasks, the predicted values from all trees are averaged to obtain the final prediction.

3.4 K-Nearest Neighbours

K-Nearest Neighbors (KNN) is a simple yet powerful supervised machine learning algorithm used for classification and regression tasks. It's based on the idea that data points with similar features tend to belong to the same class or have similar values in the case of regression. KNN is a distance-based classification algorithm. It assigns a new data point to the majority class of its k-nearest neighbors. The choice of 'k' (the number of neighbors) is a crucial hyperparameter that impacts the model's performance. KNN is an instance-based learning method, meaning it doesn't build a model during training. Instead, it memorizes the entire training dataset and uses it for predictions.

Working Principle:

Step 1: Distance Metric:

- KNN uses a distance metric (typically Euclidean distance, but others like Manhattan, Minkowski, etc., are also possible) to measure the similarity between data points. The algorithm finds the 'k' nearest neighbors with the smallest distances to the new data point.

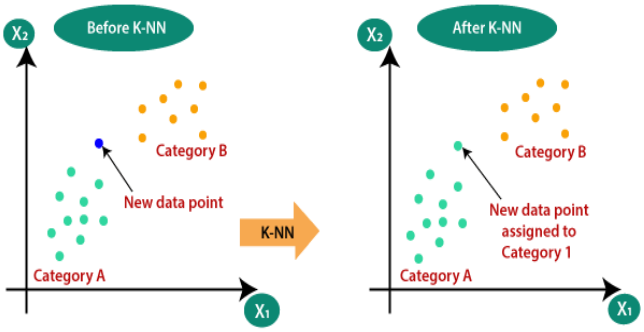

Figure 2. KNN initialization

- Voting Mechanism: For classification, KNN uses a majority voting mechanism among its neighbors. The class that occurs most frequently among the neighbors is assigned to the new data point. For regression, it takes the mean (or median) value of the 'k' nearest neighbors as the prediction.

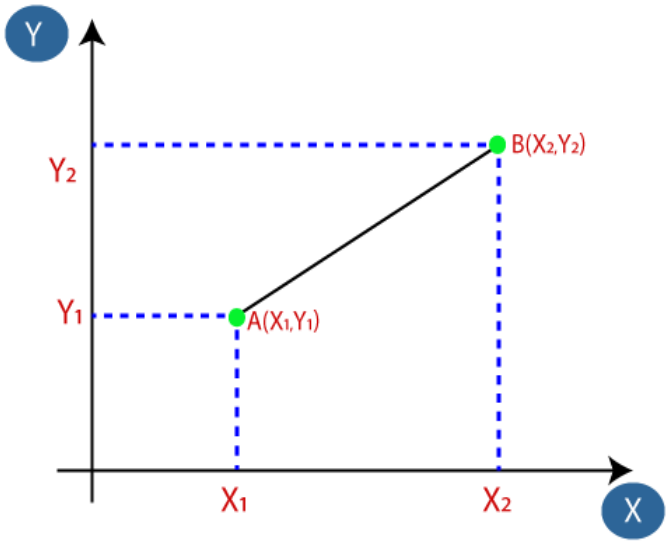

Figure 4.2. Distance measurement in KNN.

Step 2. Hyperparameter 'k':

- Choosing the Right 'k': The choice of 'k' is crucial. A small 'k' makes the model sensitive to noise and outliers but may capture local patterns well. A large 'k' smooths out local variations but can make the model less accurate.
- Methods for Choosing 'k': Cross-validation, grid search, and domain knowledge are common approaches to determine the optimal 'k' value.
- Simplicity: KNN is easy to understand and implement, making it a suitable choice for beginners.
- No Training Phase: It doesn't require a training phase since it memorizes the data, making it suitable for online learning and non-stationary data.

- Non-Parametric: KNN is non-parametric, meaning it makes no assumptions about the underlying data distribution.
- Works for Multiclass Problems: KNN naturally handles multi-class classification problems.

Step 3. Variants:

- Weighted KNN: Assigns different weights to neighbors based on their distance. Closer neighbors have a greater influence on the prediction.
- KNN with Feature Scaling: Feature scaling is essential when using KNN, as it's distance-based. Standardization (scaling features to have mean=0 and standard deviation=1) is often applied.
- KD-Tree and Ball-Tree: These data structures can be used to speed up KNN search for large datasets.

4. RESULTS AND DISCUSSION

4.1 Implementation Description

The Project offers a comprehensive solution for classifying skin cancer using machine learning, leveraging advanced techniques for image processing and model training. The system includes a graphical user interface (GUI) for ease of use, allowing users to interact with the system without needing to write any code. Here is a detailed explanation of the implementation:

- Graphical User Interface (GUI) Setup: The system uses Tkinter, a Python library for creating GUIs, to build the interface. The GUI includes several buttons and text areas that facilitate various functionalities such as dataset upload, feature extraction, data balancing, model training, and prediction. The Tk class initializes the main application window with a specific title and size. Widgets like Button, Label, and Text are used to create interactive elements and display messages to the user.
- Dataset Upload: The uploadDataset function is responsible for loading the dataset from a user-selected directory. The function uses filedialog.askdirectory to prompt the user to select a folder. It then reads the directory structure to identify different classes (subdirectories) of images, which are assumed to be organized by disease type. This structure is critical for supervised learning, where each class label is associated with a set of images.
- Feature Extraction using InceptionResNetV2: The InceptionResNetV2_feature_extraction function utilizes the InceptionResNetV2 model from Keras, a deep learning library. InceptionResNetV2 is a pre-trained model known for its high performance in image classification tasks. The function processes each image by resizing it to 224x224 pixels, converting it to an array, and extracting features through the InceptionResNetV2 model.
- If the features have already been extracted and saved (X.npy and Y.npy), the function loads them directly to save time. Otherwise, it processes all images in the dataset, extracts their features, and saves the results. This approach ensures that feature extraction is done only once, improving efficiency.
- Data Balancing with SMOTE: The Data_balancing function addresses the issue of class imbalance using the Synthetic Minority Over-sampling Technique (SMOTE). SMOTE generates synthetic samples for the minority classes to balance the dataset, which is crucial for training unbiased models. The function reshapes the feature array, applies SMOTE, and saves the balanced dataset.

- The function also visualizes the class distribution before and after SMOTE using seaborn and matplotlib to create bar plots. This visualization helps in understanding the impact of data balancing on the class distribution.
- Train-Test Splitting: The Train_test_spliting function splits the dataset into training and testing sets using train_test_split from Scikit-learn. This step is essential for evaluating the model's performance on unseen data. To ensure reproducibility, the function saves and loads shuffled indices, maintaining a consistent split across different runs.
- Model Training and Evaluation: The system includes two machine learning models: K-Nearest Neighbors (KNN) and Extra Trees Classifier (ETC).
    - K-Nearest Neighbors (KNN): Implemented in the Model2_Base function, KNN is a simple yet effective classifier that assigns the class of the majority among the k-nearest neighbors of a data point. The function checks if a pre-trained model exists; if not, it trains a new KNN model and saves it using joblib. The model's performance is evaluated using metrics such as accuracy, precision, recall, and F1-score. A confusion matrix and classification report are also generated for detailed evaluation.
    - Extra Trees Classifier (ETC): Implemented in the Model1_Base function, ETC is an ensemble learning method that aggregates the results of multiple de-correlated decision trees to improve predictive performance. Similar to the KNN function, it checks for an existing model before training a new one. The evaluation metrics and visualizations are the same as those for the KNN model.
- Prediction: The predict function allows users to upload a test image and classify it using the trained model. It preprocesses the image, extracts its features using InceptionResNetV2, and predicts the class using the selected classifier (KNN or ETC). The prediction result is displayed on the image using OpenCV, indicating the classified disease.

4.2 Dataset Description

The dataset for the skin cancer classification consists of images categorized into multiple classes, each representing a different type of skin cancer or skin condition. Typically, the dataset includes the following classes:

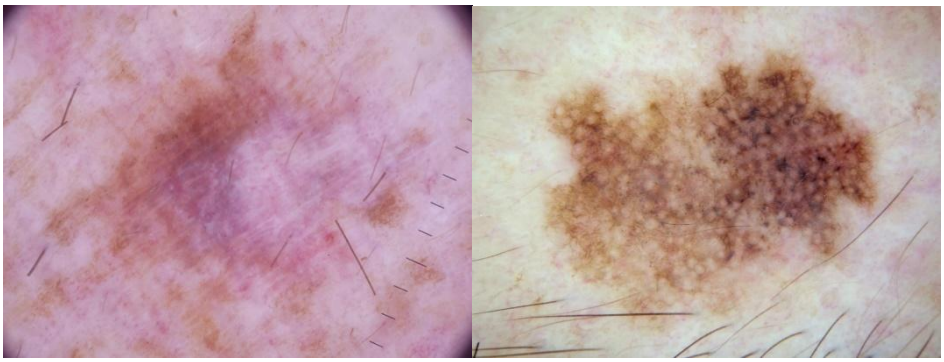

Figure 3: Skin Disease Classes.

- **Melanoma:** This is the most dangerous type of skin cancer, originating from melanocytes, the cells responsible for producing melanin. Melanoma is characterized by irregular shapes and multiple colors within a single lesion. Early detection is crucial as it can spread to other parts of the body.
- **Actinic Keratosis (AK):** AK is a pre-cancerous condition characterized by rough, scaly patches on the skin, primarily caused by long-term sun exposure. While not cancerous, AK can develop into SCC if left untreated.
- **Vascular Lesions:** These include a variety of blood vessel-related lesions such as cherry angiomas, hemangiomas, and pyogenic granulomas. They are generally red or purple and can appear anywhere on the body.
- **Benign Keratosis (BKL):** These are non-cancerous skin growths, including seborrheic keratosis, solar lentigo, and lichen planus-like keratosis. They appear as brown, black, or tan growths on the skin and are typically harmless
- **Normal:** skin cancer dataset represents healthy skin with no signs of cancerous or pre-cancerous lesions. Images in this class display typical skin textures and tones, free from irregularities, discolorations, or unusual growths. This class serves as a baseline for the model to differentiate between normal and abnormal skin conditions. Including normal skin images is crucial for the model to learn and accurately identify deviations indicative of skin diseases.

4.3 Results Description

Figure 4 shows two bar charts illustrating the class distribution of a dataset before and after applying the SMOTE (Synthetic Minority Over-sampling Technique) data balancing method. The left chart, titled "Class Distribution Before Data Balancing," shows an imbalanced dataset where class 1 has the highest number of instances (827), while class 0 has the lowest (130). Classes 2, 3, and 4 have 552, 558, and 142 instances, respectively. The right chart, titled "Class Distribution After SMOTE," demonstrates a balanced dataset where all classes (0, 1, 2, 3, and 4) now have an equal number of instances, specifically 827. This indicates that SMOTE successfully synthesized new minority class samples to match the count of the majority class, thereby addressing the initial class imbalance.

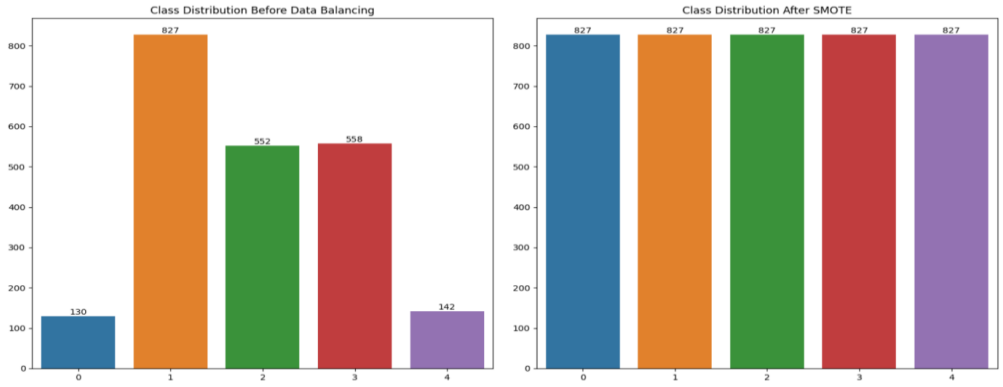

Fig. 4: Shows the count plot of each class by applying SMOTE.

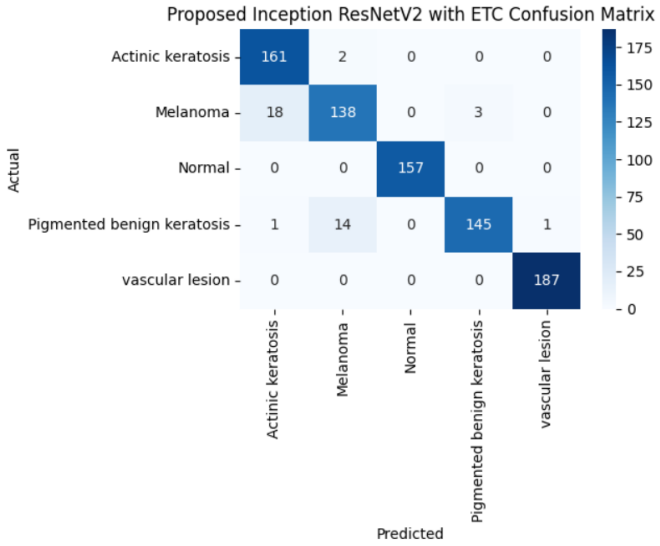

Fig. 5: Confusion Matrix of ResNetV2 with ETC Model.

The proposed model integrates the powerful feature extraction capabilities of the Inception ResNetV2 with the robust classification strength of the Extra Trees Classifier. This combined approach has significantly enhanced the classification performance, achieving an impressive accuracy of 95.28%. This indicates that the model can correctly classify approximately 95 out of every 100 skin lesion images, a substantial improvement over the KNN model. The precision and recall are both high at 0.95, meaning the model is highly accurate in its predictions and equally effective at identifying true positive cases. The f1-score of 0.95 underscores the excellent balance between precision and recall, making this model highly reliable.

In the detailed classification report, the "Actinic keratosis" class shows a high precision of 0.89 and recall of 0.99, resulting in an f1-score of 0.94. This indicates nearly flawless identification of this condition. For "Melanoma," the model has a precision of 0.90 and a recall of 0.87, with an f1-score of 0.88, representing a significant improvement over the KNN model. This performance suggests the model can more accurately identify melanoma cases, crucial for early intervention and treatment. The "Normal" class again shows perfect precision and recall, with an f1-score of 1.00, indicating the model's high confidence and accuracy in identifying healthy skin. The "Pigmented benign keratosis" class has a precision of 0.98 and recall of 0.90, with an f1-score of 0.94, showcasing the model's strong ability to distinguish this condition. The "Vascular lesion" class has near-perfect precision and recall at 0.99 and 1.00 respectively, with an f1-score of 1.00, indicating the model’s high effectiveness.

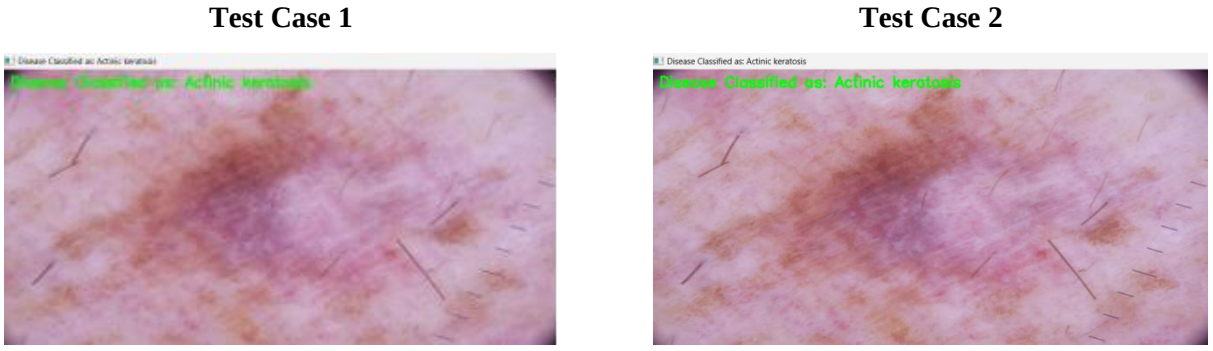

Fig. 6: Model Prediction on uploaded test images.

The figure 6 displays the confusion matrix for the proposed ResNetV2 model with Extra Trees Classifier (ETC). The matrix visualizes the model's performance in classifying different skin lesion types, allowing for a comparative analysis with the existing KNN model. The figure 10 presents performance metrics, including accuracy, precision, recall, and F1-score, for the ResNetV2 model with ETC. These metrics assess the proposed model's classification performance and provide insights into its effectiveness compared to the existing KNN model. In figure 11 the system demonstrates the process of predicting skin lesion types on uploaded test images using the trained classification model. Users upload test images through the GUI, and the model generates predictions for each image, indicating the predicted skin condition for further evaluation and diagnosis.

5. CONCLUSION

This research aims to revolutionize the diagnosis of skin cancer by leveraging advanced machine learning techniques. By integrating the InceptionResNetV2 model for feature extraction and the Extra Trees Classifier (ETC) for classification, the system provides an automated, efficient, and reliable tool for dermatologists. The system addresses several critical challenges in traditional skin cancer diagnosis, including the dependency on skilled medical professionals, the subjective nature of visual examinations, and the time-consuming process of histopathological analysis.

The automated classification system begins by uploading categorized skin lesion images, which are preprocessed for the deep learning model. The InceptionResNetV2 model, pre-trained on ImageNet, extracts robust features from these images. To address data imbalance, Synthetic Minority Over-sampling Technique (SMOTE) generates synthetic samples for underrepresented classes, ensuring a balanced training set. The Extra Trees Classifier, trained on these balanced feature vectors, provides a high-accuracy classification of skin cancer types.

The proposed system not only demonstrates high accuracy and efficiency but also offers significant benefits for various applications. It aids dermatologists by providing preliminary diagnostic results, supports telemedicine by enabling remote diagnosis, and contributes to regular screening and public health initiatives. Furthermore, as an educational and research tool, it can facilitate studies on the prevalence and characteristics of different types of skin cancer, leading to better treatment outcomes and potentially saving lives.

REFERENCES

[1] Vimercati, L., De Maria, L., Caputi, A., Cannone, E.S.S., Mansi, F., Cavone, D., Romita, P.,Argenziano, G., Di Stefani, A., Parodi, A. and Peris, K., 2020. Non-melanoma skin cancer in outdoor workers: a study on actinic keratosis in Italian navy personnel.International journal of environmental research and public health,17(7), p.2321.

[2] Leiter, U., Keim, U. and Garbe, C., 2020. Epidemiology of skin cancer: update 2019.Sunlight, Vitamin D and Skin Cancer, pp.123-139.

[3] Sander, M., Sander, M., Burbidge, T. and Beecker, J., 2020. The efficacy and safety of sunscreen use for the prevention of skin cancer.CMAJ,192(50), pp.E1802-E1808.

[4] Xu, Z., Sheykhahmad, F.R., Ghadimi, N. and Razmjooy, N., 2020. Computer-aided diagnosis of skin cancer based on soft computing techniques.Open Medicine,15(1), pp.860-871.

[5] Adla, D., Reddy, G., Nayak, P. and Karuna, G., 2021. Deep learning-based computer aided diagnosis model for skin cancer detection and classification.Distributed and Parallel Databases, pp.1-20.

[6] Carrera, E.V. and Ron-Domínguez, D., 2018, August. A computer aided diagnosis system for skin cancer detection. InInternational Conference on Technology Trends(pp. 553-563). Springer, Cham.

[7] Setiawan, A.W., 2020, February. Effect of Color Enhancement on Early Detection of Skin Cancer using Convolutional Neural Network. In2020 IEEE International Conference on Informatics, IoT, and Enabling Technologies (ICIoT)(pp. 100-103). IEEE.

[8] [Selvarasa, M. and Aponso, A., 2020, March. A Critical Analysis of Computer Aided Approaches for Skin Cancer Screening. In2020 International Conference on Image Processing and Robotics (ICIP)(pp. 1-4). IEEE.

[9] Kadampur, M.A. and Al Riyaee, S., 2020. Skin cancer detection: Applying a deep learning based model driven architecture in the cloud for classifying dermal cell images.Informatics in Medicine Unlocked,18, p.100282.

[10] Wei, L., Ding, K. and Hu, H., 2020. Automatic skin cancer detection in dermoscopy images based on ensemble lightweight deep learning network.IEEE Access,8, pp.99633-99647.

[11] Le, D.N., Le, H.X., Ngo, L.T. and Ngo, H.T., 2020. Transfer learning with class-weighted and focal loss function for automatic skin cancer classification.arXiv preprint arXiv:2009.05977.

[12] Kumar, S.N. and Ismail, B.M., 2020. Systematic investigation on Multi-Class skin cancer categorization using machine learning approach. Materials Today: Proceedings.

[13] Adla, D., Reddy, G., Nayak, P. and Karuna, G., 2021. Deep learning-based computer aided diagnosis model for skin cancer detection and classification. Distributed and Parallel Databases, pp.1-20.

[14] Ismail, M.A., Hameed, N. and Clos, J., 2021. Deep learning-based algorithm for skin cancer classification. In Proceedings of International Conference on Trends in Computational and Cognitive Engineering (pp. 709-719). Springer, Singapore.

[15] Duggani, K. and Nath, M.K., 2021. A technical review report on deep learning approach for skin cancer detection and segmentation. Data Analytics and Management, pp.87-99.

[16] Chaturvedi, S.S., Tembhurne, J.V. and Diwan, T., 2020. A multi-class skin Cancer classification using deep convolutional neural networks. Multimedia Tools and Applications, 79(39), pp.28477-28498.

[17] Ali, M.S., Miah, M.S., Haque, J., Rahman, M.M. and Islam, M.K., 2021. An enhanced technique of skin cancer classification using deep convolutional neural network with transfer learning models. Machine Learning with Applications, 5, p.100036.

[18] Sedigh, P., Sadeghian, R. and Masouleh, M.T., 2019, November. Generating synthetic medical images by using GAN to improve CNN performance in skin cancer classification. In 2019 7th International Conference on Robotics and Mechatronics (ICRoM) (pp. 497-502). IEEE.

[19] Garg, R., Maheshwari, S. and Shukla, A., 2021. Decision support system for detection and classification of skin cancer using CNN. In Innovations in Computational Intelligence and Computer Vision (pp. 578-586). Springer, Singapore.

[20] Kumar, M., Alshehri, M., AlGhamdi, R., Sharma, P. and Deep, V., 2020. A de-ann inspired skin cancer detection approach using fuzzy c-means clustering. Mobile Networks and Applications, 25(4), pp.1319-1329.

[21] Mijwil, M.M., 2021. Skin cancer disease images classification using deep learning solutions. Multimedia Tools and Applications, 80(17), pp.26255-26271.

[22] Zhang, N., Cai, Y.X., Wang, Y.Y., Tian, Y.T., Wang, X.L. and Badami, B., 2020. Skin cancer diagnosis based on optimized convolutional neural network. Artificial intelligence in medicine, 102, p.101756.

[23] Amin, J., Sharif, A., Gul, N., Anjum, M.A., Nisar, M.W., Azam, F. and Bukhari, S.A.C., 2020. Integrated design of deep features fusion for localization and classification of skin cancer. Pattern Recognition Letters, 131, pp.63-70.

[24] Afza, F., Sharif, M., Khan, M.A., Tariq, U., Yong, H.S. and Cha, J., 2022. Multiclass skin lesion classification using hybrid deep features selection and extreme learning machine. Sensors, 22(3), p.799.

[25] Cassidy, B., Kendrick, C., Brodzicki, A., Jaworek-Korjakowska, J. and Yap, M.H., 2022. Analysis of the ISIC image datasets: usage, benchmarks and recommendations. Medical Image Analysis, 75, p.102305.

[26] Raju, D. Naveen, Hariharan Shanmugasundaram, and R. Sasikumar. "Fuzzy segmentation and black widow–based optimal SVM for skin disease classification." Medical & biological engineering & computing 59, no. 10 (2021): 2019-2035.

[27] Karthik, Ra, Tejas Sunil Vaichole, Sanika Kiran Kulkarni, Ojaswa Yadav, and Faiz Khan. "Eff2Net: An efficient channel attention-based convolutional neural network for skin disease classification." Biomedical Signal Processing and Control 73 (2022): 103406.

[28] Melbin, K., and Y. Jacob Vetha Raj. "Automated detection and classification of skin diseases using diverse features and improved gray wolf-based multiple-layer perceptron neural network." International Journal of Imaging Systems and Technology 31, no. 3 (2021): 1317-1333.