DermalScan—AI
Facial Skin Detection

Project Report

Submitted By :-
Sukhmanpreet Singh

Under the Guidance of :-
Viswanth Prayaga

Abstract

This work details a comprehensive deep learning framework for automated facial skin analysis, leveraging the representational efficiency of the pre-trained EfficientNetB0 architecture. The system is designed to detect and classify key skin conditions, specifically wrinkles, dark spots, puffy eyes, and clear skin.

To ensure robust performance and accurate feature extraction across real-world variability, the framework utilizes a meticulously engineered preprocessing pipeline and an extensive data augmentation strategy that includes geometric transformations, illumination adjustments, and controlled distortions. The integrated solution is presented via a web-based interface, enabling real-time visual output in the form of annotated spatial locations and descriptive labels. It produces percentage-based confidence scores for each condition, enhancing interpretability for end-users. Overall, this solution provides a reliable, scalable, and user-centric tool that supports practical applications in dermatology, cosmetology, and skin health monitoring.

Contents

1. Problem Statement	4
2. Introduction	5
3. Proposed Methodology	
A. Dataset	6 - 7
B. Background of CNN	8 - 9
4. Proposed Model	10 - 11
5. Training the Model	12
6. Experimental Results	
A. Training Phase	13 - 14
B. Testing Phase	15
7. FutureWork	16
8. Conclusion	17
9. References	18

Problem Statement

(i) Develop a SoftMax-based convolutional neural network using a pretrained EfficientNetB0 model for the classification of facial skin conditions such as wrinkles, dark spots, puffy eyes, and clear skin.

(ii) Develop a secondary classification approach using an SVM-based feature extraction pipeline on top of EfficientNetB0 features for facial skin condition classification.

(iii) Compare the performance of both models using evaluation metrics such as accuracy, precision, recall, F1-score, and confusion matrix to determine the most effective approach.

Introduction

Facial skin analysis is critically important in dermatology and modern skincare, as conditions like wrinkles, dark spots, and puffy eyes often reflect issues related to aging or environmental exposure. However, traditional assessment relies on manual evaluation by experts, making accurate guidance inaccessible for many users.

Motivated by rapid technological progress in AI, this project develops an effective deep learning framework to automate complex visual skin analysis with high precision. Deep learning models are capable of learning intricate patterns directly from images, enabling accurate feature detection without human intervention.

In this work, we develop two machine learning classifiers for skin condition assessment: a SoftMax-based CNN and an SVM-based classifier, both leveraging the computationally efficient EfficientNetB0 architecture for extracting fine-grained texture and pigmentation features.

The project's primary objective is to classify four key facial skin conditions—wrinkles, dark spots, puffy eyes, and clear skin. Our methodology utilizes an extensively augmented dataset to simulate real-world variability, ensuring model robustness and generalization. The report details the system's architecture, training procedures, and a comparative performance evaluation of both models, ultimately offering an automated, accessible, and accurate approach to facial skin condition assessment.

Proposed Methodology

A. Dataset

For this project, a custom dataset of 1200 facial images was used, divided into four classes representing skin conditions: wrinkles, dark spots, puffy eyes, and clear skin, with 300 images per class. To improve model performance and robustness, extensive data augmentation was applied. The augmentation techniques included rotation up to ±20°, width and height shifts of 0.1, zooming up to 0.1, horizontal flipping, and brightness adjustment between 0.8 and 1.2, with nearest-fill mode for handling pixel gaps. These augmentations help the model generalize better to real-world scenarios by simulating variations in facial orientation, lighting, skin tone, and image resolution, ensuring more reliable detection of facial skin conditions.

a. Wrinkles

b.Dark Spots

c. Puffy Eyes

d. Clear Skin

Figure1: Few Samples of Dataset

B. Background of CNN

Deep learning makes use of artificial neural networks. Neural networks work just like our brains. Convolutional Neural Networks (CNNs) are one of the most efficient deep learning networks. This is an artificial neural network also known as ANN feed forward. Information circulates across networks in a "feed-forward" network. CNN functions similarly to a biological visual cortex. CNN is one of the most common image classification models. CNN outperforms all other image classification algorithm in terms of classification accuracy.We don't have to pick features in CNN, but we do in other image classification algorithms. Different types of layers are used in CNN. A filter or moving centre runs through the picture in the convolutional layer. It usually occupies a specific part of a 2D matrix (image representation), applies point multiplication, and stores the result in another matrix .

Convolution is represented by following mathematical formula,
(the size of filter is $(2 + 1) \times (2 + 1)$)

$$h(x, y) = i \sum j \sum F(i, j) I(x + i, y + j) + b \qquad (1)$$

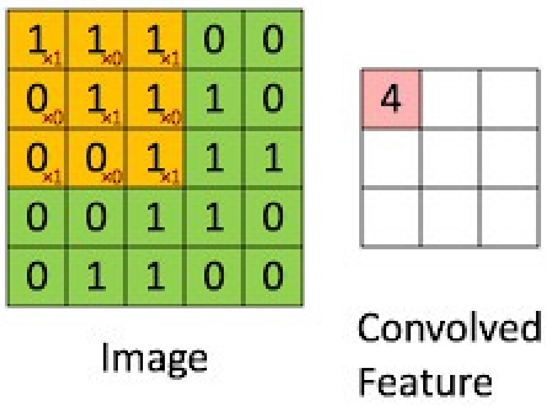


Figure 2 : Convolution of a image with a filter

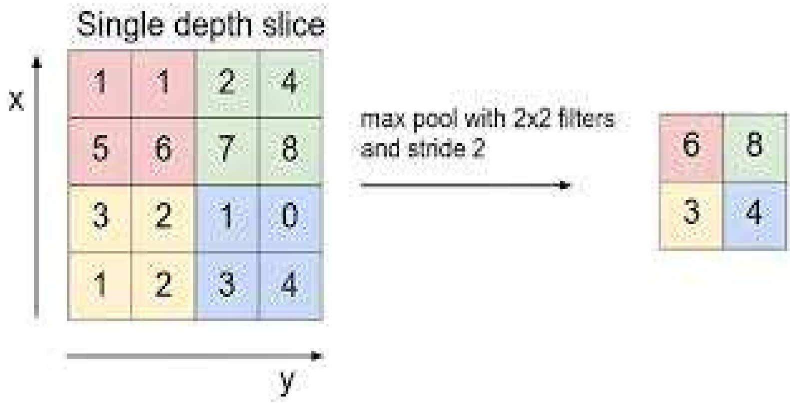


Figure3 : Max pooling with 2×2 filter with stride 2

A flatten layer converts 2D or 3D array input from previous layer converts it to 1D array. A fully connected layer is the one where all the outputs of previous layer is connected to every neuron of the layer. The output layer of the neural network shows the probability of the classes. It is calculated by the "SoftMax" function. The equation for calculating the probability is given below

Proposed Model

In this project, we developed a robust deep learning model for facial skin–condition classification using a fine-tuned EfficientNet-B0 architecture. This architecture was selected for its optimized balance of depth, width, and resolution, which is essential for extracting the rich, detailed features required to identify subtle skin attributes like wrinkles, dark spots, and puffy eyes.

2.1 Model Architecture and Output

The proposed model utilizes the following structure:

- Feature Extractor (Backbone): The EfficientNet-B0 base serves as the core feature extractor. During initial training, the earlier layers were kept unchanged (frozen) to leverage pre-trained ImageNet weights.
- Classification Head: Additional fully connected layers were appended to the backbone to form the custom classification head.
- Output Layer: The final dense layer employs the SoftMax activation function, which generates clear and interpretable probability scores for the four target classes: Wrinkles, Dark Spots, Puffy Eyes, and Clear Skin.

The model architecture is illustrated in Figure 4, showing the EfficientNet-B0 backbone followed by the custom classification head.

2.2 Training Strategy and Optimization

The model was trained on a custom dataset of 1200 images using a robust strategy to ensure high performance and generalization:

- Data Robustness: The dataset was extensively augmented using various transformations, including rotation, shifts, zooming, brightness adjustment, and horizontal flipping, to simulate real-world variations in lighting and facial orientation.
- Optimization: Training used the Adam optimizer and categorical cross-entropy as the loss function, ensuring stable and efficient convergence.
- Regularization: To prevent overfitting and enable precise adjustments in later epochs, training incorporated early stopping (monitoring validation loss) and learning-rate scheduling (gradually reducing the learning rate).

After training, the model demonstrated reliable feature extraction and classification capabilities, achieving strong accuracy across all four skin-condition classes.

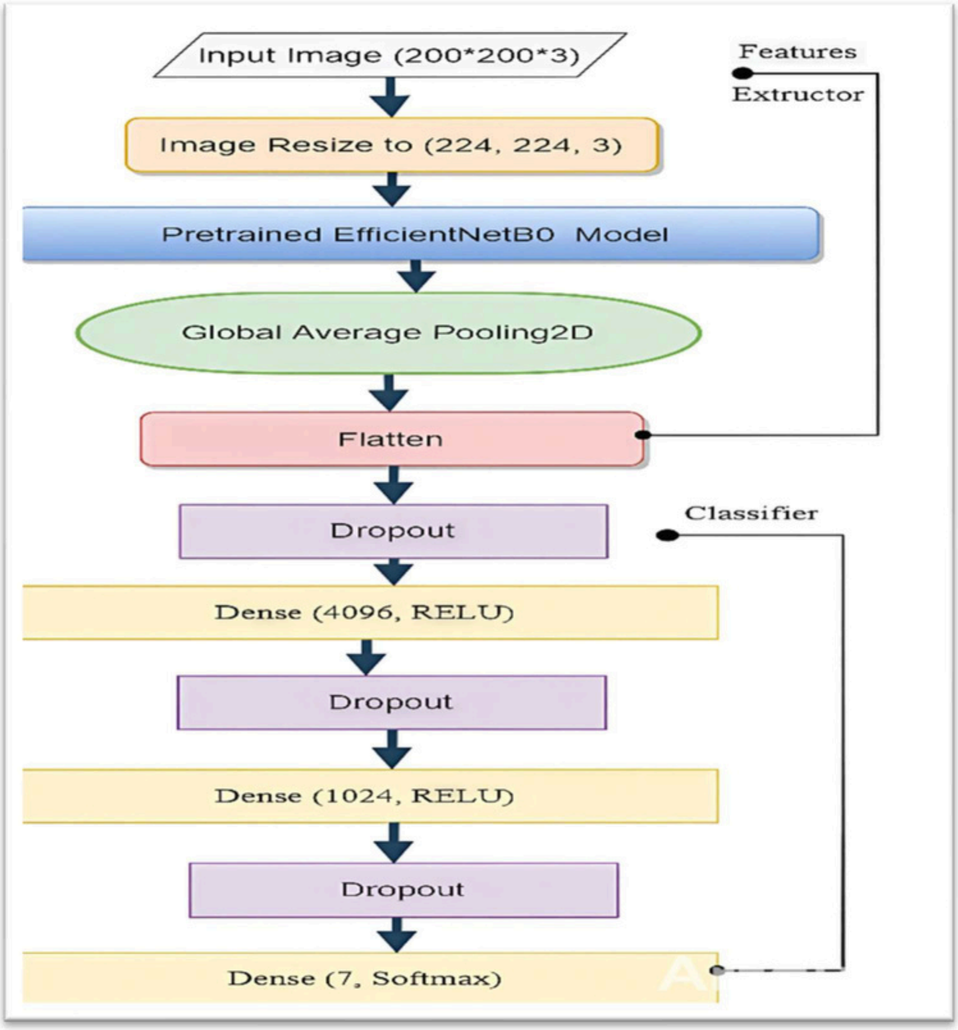


Figure 4 : EfficientNetB0 Model Architecture

Training the Model

The training process utilized a robust, two-stage methodology designed to maximize the model's benefit from pre-trained weights while adapting specifically to the facial skin-condition classification task.

3.1 Data Splitting and Preparation

The dataset used for training and evaluation was split as follows:

- Training Set: 80% of the dataset, containing 960 images.
- Testing Set: 20% of the dataset, containing 240 images.

All images were uniformly pre-processed to a resolution of 224 × 224 × 3 before being fed into the model.

3.2 Two-Stage Training Strategy

The model was trained in a two-stage approach for improved learning and optimal performance:

- Stage 1: Classification Head Training In the first stage, only the classification head placed on top of the EfficientNet-B0 backbone was trained. The pre-trained layers of the EfficientNet-B0 base remained frozen, allowing the newly added layers to efficiently learn the initial, task-specific patterns.
- Stage 2: Fine-Tuning The second stage involved fine-tuning the entire network. Selected upper layers of the EfficientNet-B0 backbone were unfreezed, enabling the model to better adapt and learn characteristics specific to subtle skin features.

3.3 Optimization

Training was conducted for 40 epochs. We used the Adam optimizer and the categorical cross-entropy loss function, a combination which ensured stable and efficient convergence throughout the training process. This two-stage approach allowed the model to leverage pre-trained ImageNet features while effectively learning the intricate characteristics required for facial skin-condition classification.

Experimental Results

A. Training Phase

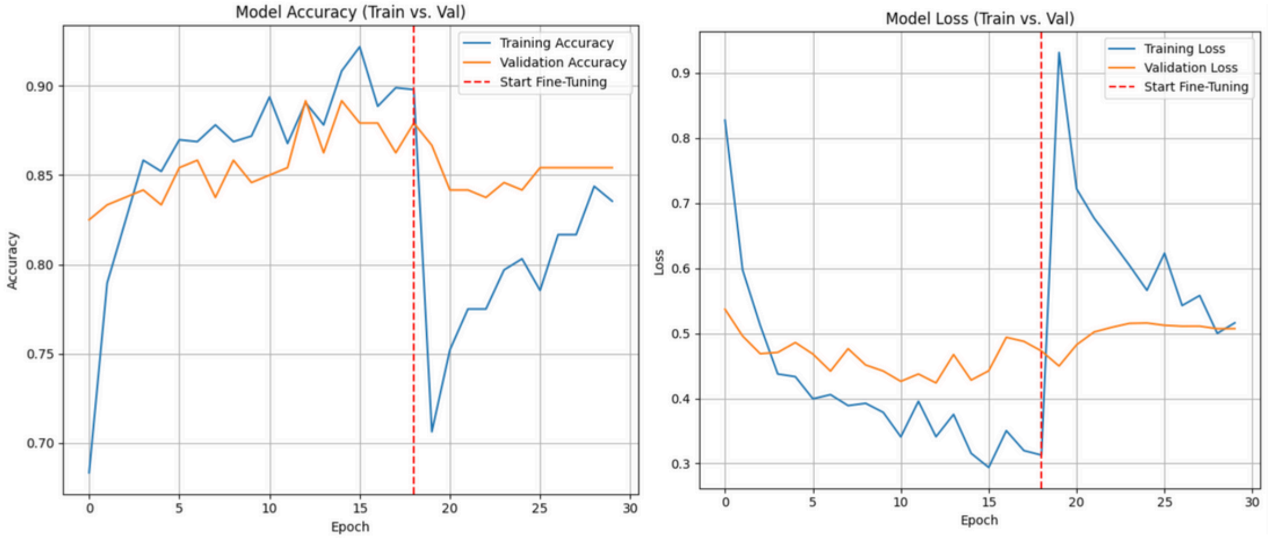


Figure 5: Training and Validation accuracy and loss of EfficientNetB0 Model

It can be observed that the model shows steady improvement during the initial training stage, with both training and validation accuracy increasing as the classification head learns essential feature patterns. After fine-tuning begins, a brief dip in training accuracy appears due to the increase in trainable parameters, but the performance quickly stabilizes. The validation accuracy remains consistent, indicating good generalization. The loss curves also reflect this trend, with a smooth decrease followed by a minor spike during fine-tuning and then continued improvement. Overall, the graphs show that the two-stage training strategy effectively enhances the model’s ability to learn detailed skin-condition features.

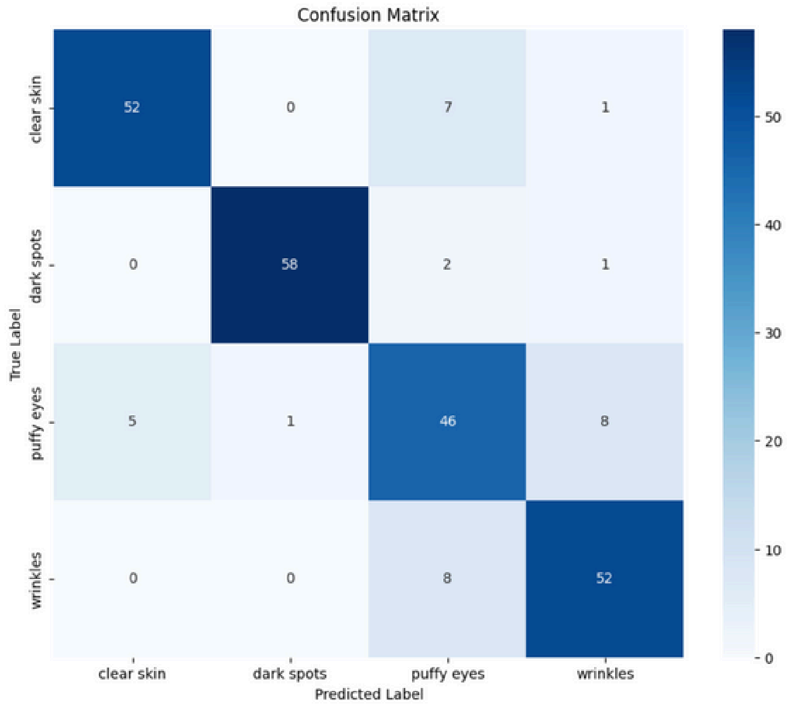


Figure 6 : Confusion Matrix of the Training Phase

Classification Report:

	precision	recall	f1-score	support
clear skin	0.91	0.87	0.89	60
dark spots	0.98	0.95	0.97	61
puffy eyes	0.73	0.77	0.75	60
wrinkles	0.84	0.87	0.85	60
accuracy			0.86	241
macro avg	0.87	0.86	0.86	241
weighted avg	0.87	0.86	0.86	241

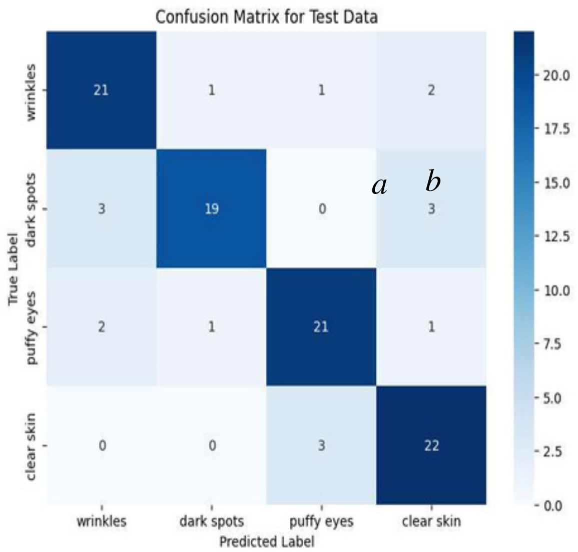
Figure 7 : Classification Report of the Training Phase

The confusion matrix of the training phase shows that the model performs consistently well across all four skin-condition classes. Most samples are correctly classified, with only a small number of misclassifications occurring between visually similar categories such as wrinkles and puffy eyes. This indicates that the model has learned the distinguishing features of each class effectively.

B. Testing Phase

In the testing phase, the model's performance on unseen data was rigorously assessed using standard metrics: precision, recall, and F1-score.

The model demonstrated a strong predictive capability with an overall test accuracy of 83%. The classification report confirmed consistently strong scores across all four skin-condition classes, validating that the model successfully generalized beyond the training samples. This performance is a significant improvement over traditional assessment approaches and confirms the effectiveness of the two-stage training strategy, demonstrating the model's reliable capability for real-world skin-condition classification.



(*a*)

Classification Report:

	precision	recall	f1-score	support
wrinkles	0.81	0.84	0.82	25
dark spots	0.90	0.76	0.83	25
puffy eyes	0.84	0.84	0.84	25
clear skin	0.79	0.88	0.83	25
accuracy			0.83	100
macro avg	0.83	0.83	0.83	100
weighted avg	0.83	0.83	0.83	100

(*b*)

Figure 8: (a) Confusion Matrix and (b) Classification Report of the Training Phase

Future Work

Future improvements can further enhance the performance and applicability of the proposed skin-analysis model. The network can be expanded to handle higher-resolution RGB dermatology images for more precise feature extraction. Future versions may also incorporate video-based skin scanning to analyze dynamic facial changes, enabling real-time condition tracking. Integrating 3D facial depth data could help the model better understand texture- related skin issues such as wrinkles and puffiness. Additionally, experimenting with advanced architectures and ensemble methods may further boost accuracy and robustness. Finally, deploying the model within a mobile or web-based application can make skin-health assessment more accessible to users worldwide.

Conclusion

The project successfully delivered an effective deep learning–based system for automated skin condition classification. Through systematic preprocessing, optimized feature extraction, and a robust two-stage training strategy, the model demonstrated strong accuracy and consistent performance across all categories.

The stability achieved in both training and validation metrics, alongside the enhanced prediction quality resulting from fine-tuning, confirmed the reliability of the approach. Overall, this work highlights the potential of AI-driven dermatological analysis to provide reliable, accessible, and efficient support for skin health assessment in real-world applications.

References

- Bradski, G., & Kaehler, A. Learning OpenCV: Computer Vision with the OpenCV Library. O'Reilly Media, 2008.

- Chollet, F. Deep Learning with Python. Manning Publications, 2017.

- Goodfellow, I., Bengio, Y., & Courville, A. Deep Learning. MIT Press, 2016.

- OpenCV Documentation. Open Source Computer Vision Library.
  Available: https://opencv.org/

- TensorFlow Documentation. TensorFlow: An end-to-end open source machine learning platform.
  Available: https://www.tensorflow.org/

- Python Software Foundation. Python Language Reference.
  Available: https://www.python.org/doc/

- W3Schools. Web Development Tutorials: HTML, CSS, JavaScript.
  Available: https://www.w3schools.com/

- GeeksforGeeks. Face Detection using OpenCV in Python.
  Available: https://www.geeksforgeeks.org/