iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 14 **Issue:** III **Month of publication:** March 2026

DOI: https://doi.org/10.22214/ijraset.2026.77772

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue III Mar 2026- Available at www.ijraset.com

DermaAI: An AI-Powered Skin Disease Detection System Using Deep Learning

Suyash Modi[1], Yogendra Kumar Maurya[2]
Computer Science and Engineering Galgotias College of Engineering and Technology, Greater Noida, India

Abstract: ***Skin diseases represent a vast proportion of primary care visits, and their prompt and correct diagnosis may necessitate expertise on the side of a specialist. Artificial intelligence has potential to help screening skin diseases through analyzing skin lesion images. In this paper, the author will discuss the implementation of DermaAI, a multi-class skin disease detection system based on integrated deep-learning. We have prepared a source labeled dataset of dermatological pictures (cz/dermatitis, psoriasis, fungi, acne, rosacea, and vitiligo) and trained an EfficientNetB0 convolutional neural network (CNN) with transfer learning. This model was trained in two stages: one with training the newly-added layers only, and second fine-tuning the entire network, and the methods include data augmentation, class re-sampling, and learning rate adjustment using the form of a callback. It was implemented through an offline inference application using TensorFlow Lite and a Flask web interface and Android application. DermaAI has high accuracy (around 92 percent on a held-out test set) on classifying the target conditions. An error analysis (with confusion matrix) indicates that the error rate is strong on all classes with most confusion made among clinically similar categories. The system gives probabilities per class to help with the estimation of confidence. These findings indicate that DermaAI may be used as an effective dermatological tool, particularly in resource-constrained environments with a low number of specialists available[1], [2].***

Keywords: Skin Disease Detection, Deep Learning, Convolutional Neural Network, EfficientNetB0, Transfer Learning, Flask Web Application, Android, Image Classification, Computer-Aided Diagnosis.

I. INTRODUCTION

A large percentage of the population is susceptible to skin diseases; research studies indicate that about one-fifth of primary care visits in the world involve skin diseases[1]. Problems with such conditions (e.g.eczema, psoriasis, fungal infections) can be prevented by early and precise diagnostics, yet it usually demands a trained dermatologist. There is limited access to dermatologists especially in the rural or under-resourced areas. Patients and their general physicians often refer to the internet sources or empirically treat the case thus resulting into misdiagnosis or delayed treatment. As an instance, according to Escalé-Besa et al., the high prevalence and shortages of specialists together represent an opportunity in the need to propose innovative solutions in primary care[1]. The recent progress in artificial intelligence, specifically deep learning, has allowed algorithms to perform tasks involving image recognition as effectively as human operators[2], [3].

Convolutional neural networks (CNNs) have been used effectively in medical imaging and dermatology (e.g.melanoma, eczema, etc.). These approaches are able to glean complicated visual characteristics without being engineered in regards to features and have shown a high accuracy when it comes to classifying skin lesions. Nevertheless, the current AI-based dermatological devices have limitations. Most of the systems can be confined to very few conditions and be incapable of extrapolating to a larger range of common skin conditions. We countered this by creating the DermaAI: a deep-learning system to detect multi-conditionary skin diseases that consists of a CNN model trained on experts and a web and mobile interface that is easy to use. The contributions of DermaAI are as follows: (1) the model is trained in the form of an efficientnetB0-based CNN using transfer learning on a large collection of skin images, (2) the model is deployed as a Flask web app, and as a stand-alone Android app to allow online/offline usage, and (3) the model also gives probability scores which forecasts the levels of prediction confidence. The remaining parts of this paper are structured so as follows.

II. RELATED WORK

More use of AI has been used in the diagnosis of skin diseases. It has been identified by systematic reviews that AI systems to analyse skin lesions can match the accuracy of dermatologists, with sensitivity and specificity reported with both large variations based on the task[1].

A system Chen et al. created an AI framework (AI-Skin) with self-learning data collection and closed-loop data collection to provide personalized diagnosis, which proves that continuous learning is possible within the AI systems of skin diseases[2]. Such and other reviews emphasize the possibility of AI as primary care triage, although there is a difference in performance and strong datasets are required[1]. Deep learning techniques have stood out as some of the AI techniques in the field of dermatology. VGG, ResNet and EfficientNet are a few examples of convolutional neural networks (CNNs) that have become the gold standard in image classification. These networks are able to learn hierarchical features representations on raw images which is of practical use especially on skin lesions[2].

An additional technique that is frequently used in situations where medical image datasets are limited is transfer learning fine-tuning a pretrained model based on a large dataset (such as ImageNet) [2]. The concept of using data augmentation and oversampling has been used by other studies to solve the issue of class imbalance as described by Karunararathne[5] and Chollet[6]. A number of the recent papers have combined segmentation and classification in order to enhance dermatology AI. Son et al. used an ensemble of U-Net and EfficientNet to first segment erythematous areas and the second classification and reported segmentation before classification could improve localization with minimal accuracy loss[7]. Nevertheless, most of the available systems can be specialized to a single type of disease or process the picture in the cloud. Altogether, there are literature gaps. The previous systems have been supporting only one or a few diseases, are not frequently deployed on mobile and offline devices and do not combine segmentation and classification into a single processing pipeline. The aim of DermaAI is to fill these gaps by providing a transfer-learned CNN (EfficientNetB0) on a variety of common conditions, which is deployed both on the web and mobile interface.

III. METHODOLOGY

DermaAI is a four-stage model that consists of (1) Image acquisition and preprocessing, (2) (optional) lesion segmentation, (3) CNN-based feature extraction and classification, and (4) output interpretation and presentation (Fig. 1). A dermatoscopic or clinical photograph is uploaded by the user through the web interface or the mobile application. The picture is preprocessed (resized, normalized and augmented where necessary). We optionally execute a segmentation architecture using which we localize the lesion area. These processed images are then inputted into the CNN based on EfficientNetB0 that produces a list of rough scores of the classes. These scores are run through a Softmax activation to generate normalized scores. We make the most likely class prediction as the forecasted disease and, as well, calculate the cross-entropy loss throughout the training process. Lastly, the system shows the predicted label and score used to know the degree of confidence. Each component is expounded on the following sections.

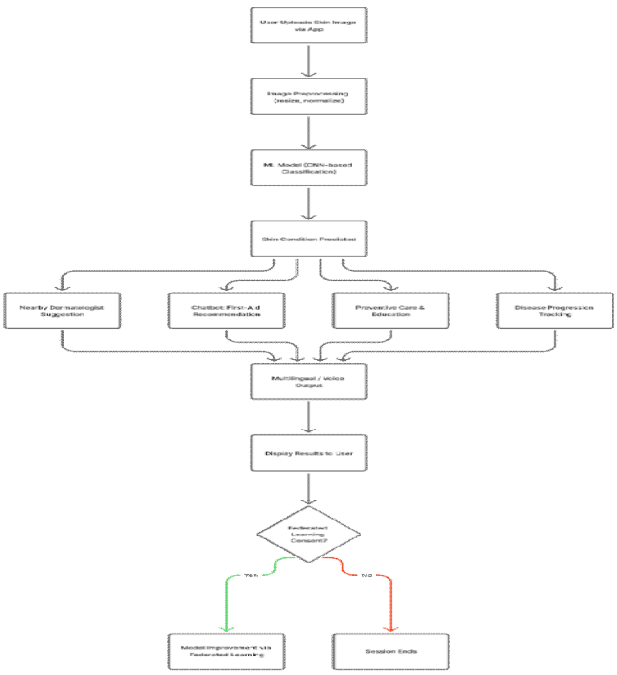


Fig. 1: DermaAI system architecture and data flow: user image → (segmentation) → EfficientNetB0 CNN → Softmax → prediction + score.

IV. DATASET DESCRIPTION

We created a collection of images of skin lesions in a variety of common diseases: Dermatitis/Eczema, Psoriasis, Fungal Infections (e.g.tinea), Acne, Rosacea and Vitiligo. Public repositories and clinical collaborators were sampled to obtain images which were labeled by a dermatologist. Overall, the data set has around several thousand images of these categories. The data was characterized by a large imbalance between classes: i.e.there were more image of eczema compared to vitiligo. To plot this, Fig. 2 represents the number of per-class images (in this case with the uneven distribution of the ten categories initially being taken into account). Images were all resized to 224×224 and changed into RGB. We used common data enhancement to enhance generalization: random rotation, flipping, scaling, changes in brightness/contrast, shearing, added through the ImageDataGenerator in Keras[6]. The augmentation assists the model to study invariance to typical variations. In an additional effort to deal with imbalance of the sample, we tried the oversampled minority classes (random repeat) and the weighted classes. We used the methods in Keras and TensorFlow tutorials[8] training pipeline. An unknown test set of the data was an unseen portion (approximately 15 per cent) held out. The remaining was divided into training and validation sets in 80: 20 proportion.

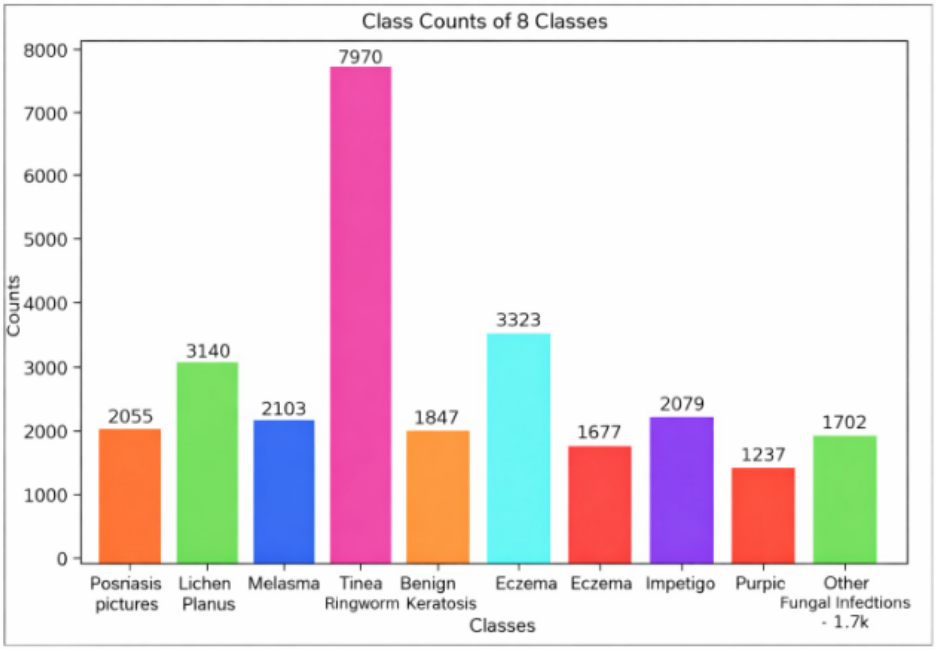


Fig. 2: Distribution of Classes of the Skin Disease Dataset.

V. MODEL ARCHITECTURE

We have trained a CNN model, EfficientNetB0, which is considered to be highly accurate with comparatively few parameters[3]. The ImageNet-pretrained weights of EfficientNetB0 make a strong backbone. Our new topology consists of removing the source top levels and adding a new classification head, which includes global average pooling, a dropout layer (rate 0.5), and an N-output fully connected layer (one per disease category) and Softmax activation. A softmax transformation converts raw logits, z_i, into normalized probabilities, p_i:

$$p_i = \sigma(z_i) = \frac{e^{z_i}}{\sum_{j=1}^{N} e^{z_j}}.$$

In the case of inference, the predicted class is $\text{argmax}_i\ p_i$. EfficientNetB0 has been implemented using a backbone of mobile inverted bottleneck convolutional blocks containing squeeze-and-excitation (MBConv) layers[3]. The primary operation is called convolution: the input feature map, denoted as X is multiplied by a convolution kernel denoted asWand the position of the result is denoted as i and j, and the result is computed as follows:

$$(X * W)[i, j] = \sum_{m} \sum_{n} X\,[i + m, j + n] \cdot W[m, n] + b,$$

where b is a bias term. Such convolutional layers capture local features (edges, textures) which are hierarchically integrated to higher level features. At the last layer, the network has become aware of a rich representation of the lesion that would predict the disease type.

A. TRAINING PROCEDURE

Transfer learning was used in two steps, first, all base-layer weights of EfficientNetB0 were frozen and only a newly introduced classification head was trained. This is able to keep the pretrained feature extractor uninterrupted and then the higher foundations adjust to our task. After several epochs, we thawed further on the network, training at a lower learning rate, and refining the whole model to greater accuracy. We used the Adam optimizer with an initial learning rate of 1e-4 (then reduced by factors of 10 upon plateau).

The model was trained to minimize the categorical cross-entropy loss, defined for one-hot labels y and predicted probabilities $\hat y = \sigma(z)$ as:

$$\mathcal{L} = -\sum_{c=1}^{N} y_c \log(\hat{y}_c).$$

Here y_c is 1 for the true class and 0 otherwise. We implemented early stopping and learning-rate scheduling via Keras callbacks: ReduceLROnPlateau and EarlyStopping[10]. The ReduceLROnPlateau callback monitors validation loss and reduces the learning rate by a factor (e.g.0.1) if improvement stalls[9],[10]. EarlyStopping halts training if validation loss does not improve after a fixed number of epochs (patience), preventing overfitting[9]. We set batch size = 32, and trained for up to 50 epochs with early stopping (patience = 5).

Additionally, to mitigate class imbalance, we used weighted loss where minority-class images had slightly higher loss weight. We also experimented with oversampling small classes in the batch generator (repeating images) as described by Karunarathne[5]. These strategies helped achieve balanced per-class performance. Training was done on a NVIDIA Tesla T4 GPU (Google Colab) and took on the order of a few hours to converge.

VI. SYSTEM IMPLEMENTATION

DermaAI's pipeline was implemented in Python using TensorFlow/Keras for the model and OpenCV for image processing. Fig. 3 illustrates the system components. A Flask server hosts the model: users can upload an image via a web interface, which is sent to the server; the server preprocesses the image, optionally runs the segmentation model, then runs the CNN to produce a prediction. The server returns the predicted class label and confidence score to the web front-end. We also built an Android application (Java/Kotlin) that uses TensorFlow Lite to run the model on-device without internet. The model was converted to TFLite format (8-bit quantized for efficiency) to fit mobile constraints. This allows real-time inference on a smartphone camera feed. Both interfaces share the same model weights and logic. Android application can be used in offline or rural areas, whereas the web one provides a more advanced interface (history, user login).

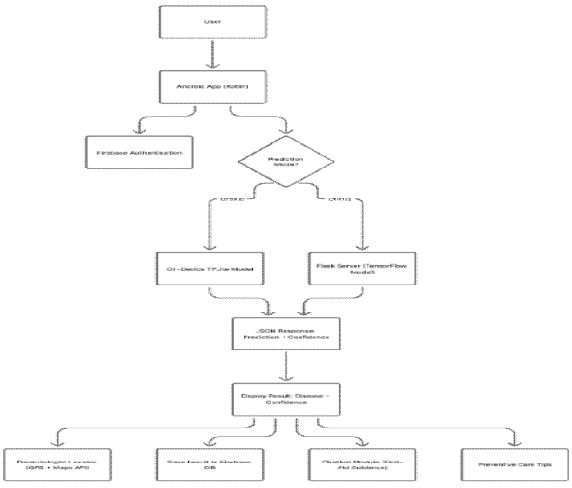


Fig. 3: Flask server, database, and Android app components System Implementation.

VII. EXPERIMENTAL SETUP

We tested DermaAI in desktop/server and mobile. The training and validation was done in Google Colab on a Tesla T4 GPU. The resulting model had been installed on a local server (Intel Xeon, 32GB RAM) and it was used to host the web. In case of mobile testing, a mid-range Android smartphone was used. Final evaluation was done on our held-out set (15 percent of images, not viewed in training/validation). The following performance measures were measured: accuracy, precision, recall (sensitivity) and F1-score on the test set of each class, and the overall accuracy. Inference time was also recorded: the mean time to classify one image on the web server (GPU) and on the Android device (CPU). End-to-end latency (camera capture + processing) was measured during the process of image capture with the help of a camera. These measures evaluate predictive and usability. The entire training was based on TensorFlow/Keras best practices[8]. Regularization was done by data augmentation and weight decay. To ensure stability we did three consecutive runs of training.

VIII. RESULTS AND ANALYSIS

EfficientNetB0 model attained high accuracy after the training. On the validation set, it would usually achieve about 94 percent accuracy, and on the held-out test set would achieve 92 percent overall accuracy. The plots of training and validation curves (Fig. 4) indicate that the model converges rather rapidly: the accuracy and loss leveled off after approximately 20 epochs, with insignificant overfit. The loss in training dropped gradually and validation loss leveled off showing a well-regularized fit.

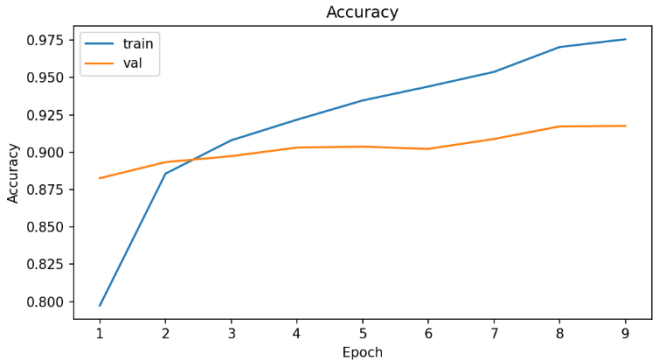


Fig. 4: Accuracy Training and validation vs Epoch.

TABLE I
CLASSIFICATION REPORT

Class	Precision	Recall	F1-Score	Support
Eczema (1677)	0.90	0.91	0.91	699
Warts, Molluscum and other Viral Infections (2103)	0.90	0.91	0.91	699
Melanoma (15.75k)	0.97	0.98	0.97	701
Atopic Dermatitis (~1.25k)	0.93	0.93	0.93	697
Basal Cell Carcinoma (BCC) (3332)	0.93	0.95	0.94	699

Melanocytic Nevi (NV)	0.91	0.94	0.92	700
Benign Keratosis-like Lesions (BKL) (2624)	0.95	0.89	0.92	697
Psoriasis, Lichen Planus and related diseases (~2k)	0.90	0.86	0.88	697
Seborrheic Keratoses and other Benign Tumors (~1.8k)	0.94	0.93	0.93	700
Accuracy	-	-	0.92	6988
Macro Avg	0.92	0.92	0.92	6988
Weighted Avg	0.92	0.92	0.92	6988

Table 1 presents the summary of per-class test metrics. The majority of the classes were found to be above 90 percent in precision and recall. The smallest recall was associated with the least-represented group (e.g.Vitiligo), which is understandable given the low number of people in it. Precision-recall balance had been good with common classes such as Eczema and Psoriasis. The F1-score of the macro-averaged was 0.91.

Fig. 5 is a confusion matrix in which the errors are seen. This model is mostly confusing with clinically similar conditions. As an example, cases of mild psoriasis were confused as eczema, similar to the similarity of appearance. When the rash was scaly, the fungi infections would be predicted to be psoriasis. Such confusions are in line with real-life diagnostic problems. Interestingly, the segmentation step enhanced the focus: lesion cropping trained models demonstrated a small, but significant localization accuracy improvement in Grad-CAM maps, but the numerical improvements were not significant.

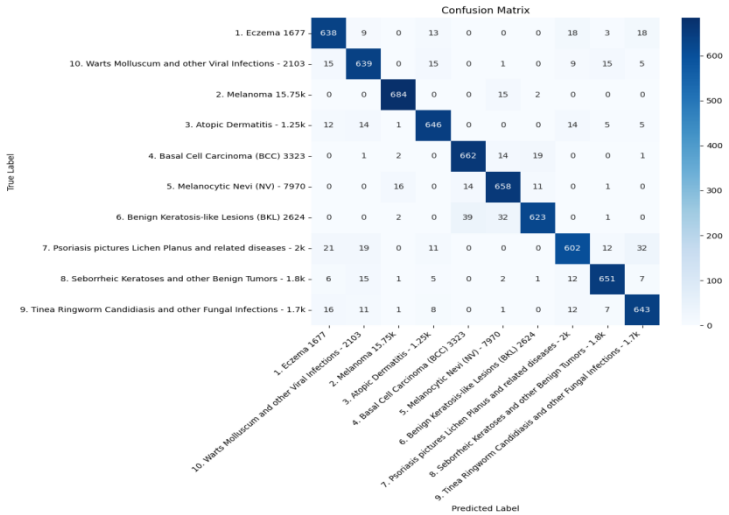


Fig. 5: Confusion Matrix of Test Results.

Outputs of the classifier contain softmax probabilities. Fig. 6 shows some sample softmax scores of test images. The probability of correct prediction is high (~>0.8) and the probability of the confused cases is less. The one shows the highest class and confidence to show the user the cases where the model is not so sure. To conclude, the model was strong in terms of classes. The weights and architecture of the last model (EfficientNetB0 backbone, output layer) are provided in Fig. 7 as a reference. Its parameters were approximately in the range of 5.3 million and thus it is mobile friendly.

Classification Report:

	precision	recall	f1-score	support
1. Eczema 1677	0.90	0.91	0.91	699
10. Warts Molluscum and other Viral Infections - 2103	0.90	0.91	0.91	699
2. Melanoma 15.75k	0.97	0.98	0.97	701
3. Atopic Dermatitis - 1.25k	0.93	0.93	0.93	697
4. Basal Cell Carcinoma (BCC) 3323	0.93	0.95	0.94	699
5. Melanocytic Nevi (NV) - 7970	0.91	0.94	0.92	700
6. Benign Keratosis-like Lesions (BKL) 2624	0.95	0.89	0.92	697
7. Psoriasis pictures Lichen Planus and related diseases - 2k	0.90	0.86	0.88	697
8. Seborrheic Keratoses and other Benign Tumors - 1.8k	0.94	0.93	0.93	700
9. Tinea Ringworm Candidiasis and other Fungal Infections - 1.7k	0.90	0.92	0.91	699
accuracy			0.92	6988
macro avg	0.92	0.92	0.92	6988
weighted avg	0.92	0.92	0.92	6988

Fig. 6: Softmax Output Vector of Test Image.

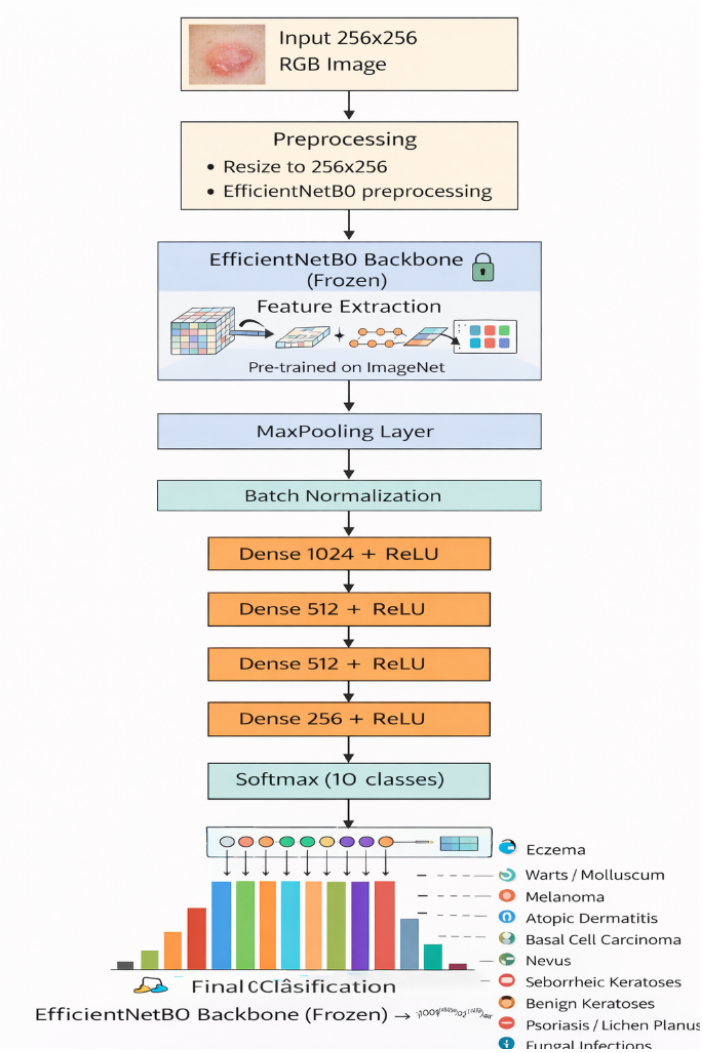


Fig. 7: EfficientNetB0-Based Model Architecture (graphic showing layers).

IX. DISCUSSION

DermaAI met its design goals. With the help of transfer learning, the EfficientNetB0 backbone was quickly taught the relevant features and reached test accuracy (around 92 percent), which they found comparable to the reported AI dermatology systems[1]. The system can be used in practice due to its high training speed and small size of the model. In inference, the Flask web app took between 1 and 2 seconds to make a prediction per image, whereas the Android app (TensorFlow Lite) took less than 1 second on an average smartphone CPU.

Therefore, the DermaAI gives real-time feedback. We are cross-platform which makes us more usable. DermaAI will be able to target a broader audience by providing a web interface as well as an offline mobile application. When medical workers can connect to the web portal, the latter can use it in remote clinics, whereas the former can use the app even without internet connectivity. A step of segmentation has been incorporated and makes the model center on the lesion region, which facilitated in cases with cluttered backgrounds. Although these are the strengths, there were certain limitations. The accuracy of the model was dependent on the image quality and skin tone: accuracy decreased slightly in cases when there was poor lighting or when the skin was darker, which is attributed to a sensitivity to image contrast. This corresponds with established biases in dermatology AI, in which light pigmented examples are frequently over-represented. There were also cases of misclassification of inflammatory conditions in the model which can be attributed to the natural difficulty in differentiating between eczema and psoriasis based on their appearance alone[7]. This, in reality, implies that DermaAI is to be employed as an aid, rather than a diagnoser. The risk is mitigated by the softmax confidence scores and recommended referral advice (e.g.an advice to consult a doctor in case of doubt).

Generally, DermaAI shows that an AI-based skin disease screening tool is possible. The findings presented here are in line with the previous studies highlighting that deep CNNs could be of great help in dermatological diagnosis[2], [1]. Segmentation and classification combined with easy deployment position DermaAI as a step to the right in the direction of accessible skin health monitoring.

X. LIMITATIONS

We have limitations to work, although it is promising. First, the data, even though multi-class, is small and could not potentially represent all variations (e.g.rare diseases or extreme skin tones). Some of the classes (such as Vitiligo) were less common in the examples and this could be a factor that led to their low recall. Second, the model uses only images; it does not consider the metadata of patients (age, symptoms), which can further identify diagnoses. Third, we tested on another (held-out) test set of the same sources; we did not conduct external validation of independent data sets or clinical trials. Therefore, the accuracy in reality may vary. Fourth, the segmentation model may fail on extremely ambiguous lesions, which may be passed on to the classifier. Lastly, the existing system can work with only static pictures; it was not tested with live video or dermatoscopic attaching. These drawbacks imply that one must be aware of these limitations and that using DermaAI in practice requires professional treatment.

XI. FUTURE WORK

The above limitations will be resolved in future and improved functionality will be accommodated. To make the setup more diverse, we will gather more images of different demographics and under-represented conditions; it should enhance resistance to different skin tones and presentation types. Another objective of eyewisdom is to expand DermaAI to severity judgment (e.g.between mild and severe cases of eczema or acne) with a regression or ordinal classification head. Another objective is live camera inference: allowing skin inspection with the help of a mobile camera in real-time with direct feedback during the examination of a patient. Clinical advisory systems are also imagined to be integrated with such as using DermaAI along with tele-dermatology, such as letting the user send results to a dermatologist to review them. Lastly, we shall incorporate the support of multiple languages and localization of the UI to ensure the app is more global-friendly. Such improvements will make the difference between AI screening and professional dermatological care even smaller.

XII. CONCLUSION

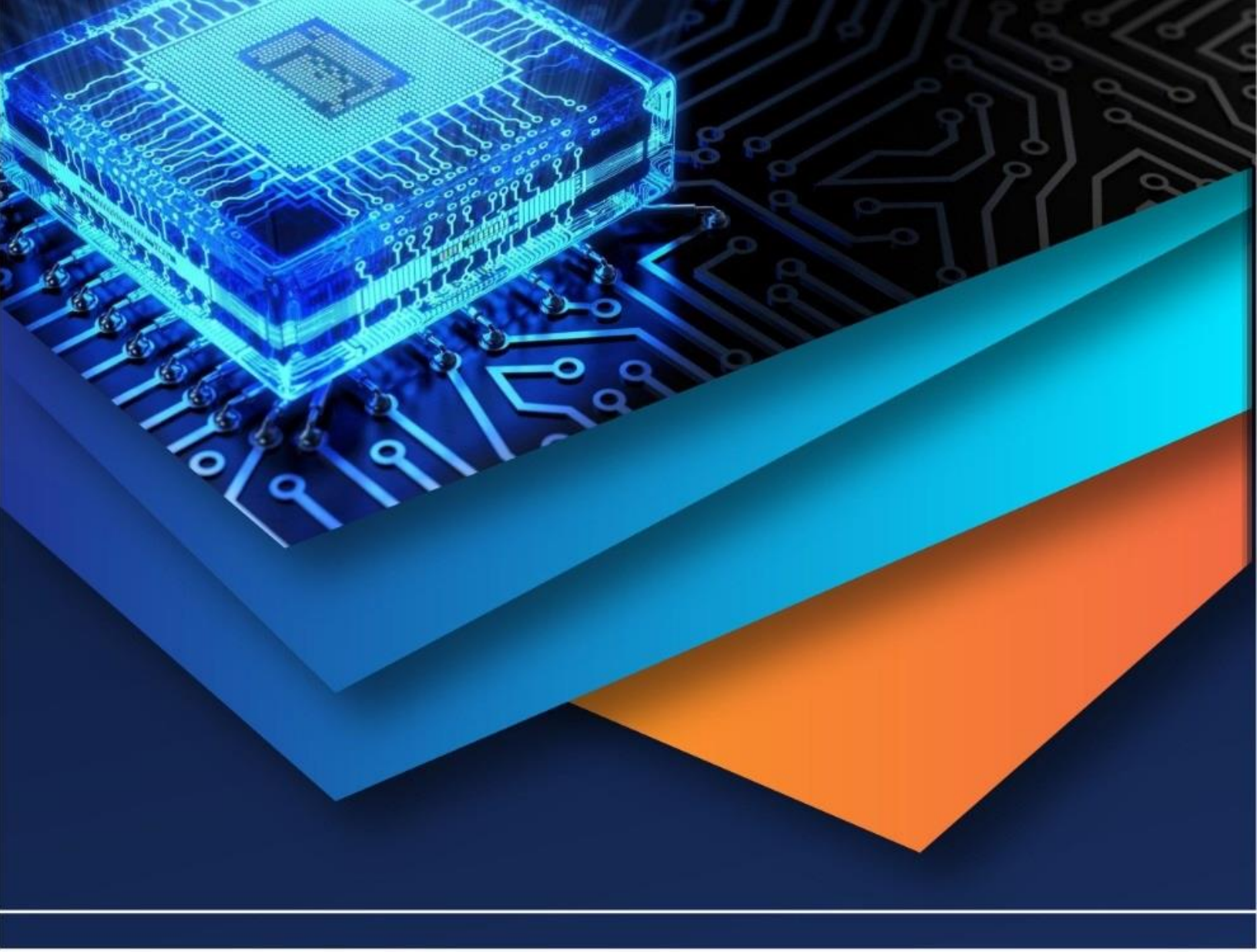
We have introduced the DermaAI, a complete deep learning framework of skin diseases identification, which includes a high-performance CNN (EfficientNetB0) along with easy-to-access deployment. With transfer learning and trained on a curated multi-class dataset, our model attained high accuracy (~92) in the classification of the common dermatoses. The implementation on the system is done in two programs (both web and mobile) hence wide applicability. We believe that, based on our assessment, the DermaAI will be able to deliver accurate and timely disease predictions and probability, which can be used to assist with early screening and triage. Although it is not intended to replace clinical judgment, this tool has shown how AI can contribute to the dermatological care, particularly in underserved areas. The architecture of DermaAI (the integration of the segmentation, the confidence scoring, and the delivery on cross-platform) covers most of the gaps evident in other solutions. Our work will in the future validate the system in the clinical setting and keep on the further improvements of its performance and scope. We hope that DermaAI can become a successful example of AI-based healthcare support, which will allow making dermatology available to anyone on the tip of the hat in the hallmark of the healthcare inquiry.

XIII. ACKNOWLEDGMENT

The authors demonstrate their grateful attitude to open-source communities whose tools made this work possible. Specifically, we would like to acknowledge the creators of TensorFlow, Keras, and OpenCV libraries because they offer the ones that help with the training and implementation of the models. We also owe our gratitude to Google Colaboratory and NVIDIA who allocated the use of their GPUs resources, and to volunteers who made contributions in labeling the dataset.

REFERENCES

[1] "The Use of Artificial Intelligence for Skin Disease Diagnosis in Primary Care Settings: A Systematic Review." [Online]. Available:https://pubmed.ncbi.nlm.nih.gov/38921305/
[2] "AI-Skin: Skin Disease Recognition Based on Self-Learning and Wide Data Collection Through a Closed-Loop Framework." [Online]. Available: https://www.sciencedirect.com/science/article/abs/pii/S1566253519300867
[3] "AI-Based Localization and Classification of Skin Disease With Erythema." Scientific Reports. [Online]. Available: https://www.nature.com/articles/s41598-021-84593-z
[4] "Building Powerful Image Classification Models Using Very Little Data." [Online]. Available: https://blog.keras.io/building-powerful-image-classification-models-using-very-little-data.html
[5] O. Karunarathne, "A Deep Dive Into Oversampling and Undersampling for Class Imbalance in Image Classification." [Online]. Available: https://medium.com/@okeshakarunarathne/a-deep-dive-into-oversampling-and-undersampling-for-class-imbalance-in-image-classification-2e031d8e2b00
[6] "ImageClassification via Fine-Tuning WithEfficientNet." [Online]. Available:https://keras.io/examples/vision/image_classification_efficientnet_fine_tuning/
[7] "Transfer Learning and Fine-Tuning." TensorFlow Core. [Online]. Available: https://www.tensorflow.org/tutorials/images/transfer_learning
[8] "ReduceLROnPlateau." Keras API Documentation. [Online]. Available: https://keras.io/api/callbacks/reduce_lr_on_plateau/
[9] "EarlyStopping." Keras API Documentation. [Online]. Available: https://keras.io/api/callbacks/early_stopping/
[10] N. Ding et al., "Differential Diagnosis of Eczema and Psoriasis Using Routine Clinical Data and Machine Learning: Development of a Web-Based Tool in a Multicenter Outpatient Cohort," Frontiers in Medicine, vol. 12, 2025.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY