ADVANCE SOCIAL SCIENCE ARCHIVE JOURNAL
Available Online: https://assajournal.com
Vol. 04 No. 02. Oct-Dec 2025.Page#.1785-1797
Print ISSN: 3006-2497 Online ISSN: 3006-2500
Platform & Workflow by: Open Journal Systems
https://doi.org/10.5281/zenodo.17656620

A Privacy-Preserving System for Skin Disease Diagnosis Using a Lightweight CNN and TensorFlow.js for Client-Side Inference

Usman Ali
Department of Computer Science, The University of Faisalabad, Faisalabad, Punjab Pakistan
2023-ms-cs-008@tuf.edu.pk
Abdul Rauf
Department of Computer Science, The University of Faisalabad, Faisalabad, Punjab Pakistan
abdulrauf2000.pk@gmail.com
Uzair Saeed
Department of Computer Science, The University of Faisalabad, Faisalabad, Punjab Pakistan
saeed_u@live.com
Majid Hussain (Corresponding Author)
Department of Computer Science, The University of Faisalabad, Faisalabad, Punjab Pakistan
majidhussain1976@gmail.com
Samraiz Zahid
Department of Computer Science, The University of Faisalabad, Faisalabad, Punjab Pakistan
samraiz99128@gmail.com

ABSTRACT

Skin diseases are among the most prevalent health concerns worldwide, ranging from mild conditions such as acne and eczema to life-threatening disorders like melanoma. Early and accurate diagnosis is critical for effective treatment; however, access to dermatological care remains limited in many regions due to a shortage of specialists and diagnostic resources. The similarity in visual features across different skin conditions further complicates timely detection and increases the risk of misdiagnosis.This research aims to design and implement an intelligent, accessible, and privacy-preserving system for automated skin disease diagnosis. The primary objective is to democratize dermatological screening by enabling real-time, low-cost, and user-friendly diagnostic assistance that can operate without dependence on clinical infrastructure.To achieve this, a custom Convolutional Neural Network (CNN) model developed and trained on publicly available dermoscopic datasets, including HAM10000. The dataset underwent preprocessing techniques such as resizing, normalization, augmentation, and class balancing to improve generalization. The trained CNN then converted to a browser-compatible TensorFlow.js format and integrated with a ReactJS-based web application. This architecture enables client-side inference, ensuring data privacy and offline functionality while providing immediate diagnostic feedback.Experimental results demonstrate that the proposed model achieves high classification performance, with an average accuracy exceeding 87% and balanced precision, recall, and F1-scores across multiple disease categories. Inference times were consistently under one second on modern laptops and smartphones, validating the system's suitability for real-time use.This work highlights the potential of lightweight deep learning models combined with web technologies to deliver accessible dermatological diagnostic support, particularly in low-resource environments. It contributes to advancing *digital health solutions that improve early detection, reduce healthcare disparities, and empower users with affordable, privacy-focused diagnostic tools.*

Keywords: *Skin disease diagnosis, Convolutional Neural Networks (CNN), Dermoscopic image analysis, TensorFlow.js, ReactJS, Browser-based inference, Privacy-preserving AI, Real-time medical imaging, Digital health, HAM10000 dataset.*

Introduction:

The skin, as the largest and most visible organ of the human body, plays a vital role in protection, thermoregulation, sensation, and immunity. Despite its resilience, it is highly susceptible to a wide range of diseases, from common conditions such as acne and eczema to severe illnesses including melanoma and other skin cancers. According to the World Health Organization, more than 900 million people affected by skin-related disorders at any given time, placing dermatological conditions among the most widespread global health burdens. Early and accurate diagnosis is critical for improving treatment outcomes and reducing complications, yet access to timely dermatological care remains uneven across the world.The primary challenge in dermatology lies in the limited availability of specialized professionals, especially in rural and resource-constrained regions, coupled with the high costs and infrastructural demands of diagnostic equipment. Furthermore, many skin diseases exhibit visually similar features—such as redness, scaling, or irregular pigmentation—, which increases the risk of misdiagnosis, particularly when patients first seek care from non-specialists. These limitations not only delay treatment but also contribute to preventable morbidity and, in severe cases, mortality.

To address these challenges, this research introduces a deep learning-based system for automated skin disease diagnosis, leveraging Convolutional Neural Networks (CNNs) for accurate classification of dermoscopicimages. A custom CNN model trained on publicly available dermatological datasets and optimized for browser-based deployment using TensorFlow.js. By integrating the model into a ReactJS web application, the system provides real-time, client-side predictions without requiring internet connectivity or server-side infrastructure. This ensures user privacy, reduces diagnostic delays, and makes the solution suitable for low-resource environments. The significance of this research lies in its potential to democratize access to dermatological care. By offering an affordable, user-friendly, and privacy-focused tool, the system empowers individuals to perform preliminary assessments, encourages timely medical consultation, and supports healthcare providers in decision-making. Ultimately, this work contributes to the broader vision of using artificial intelligence to bridge healthcare disparities, improve early disease detection, and advance global health equity.

Background:

A. Dermatology and the Need for Early Detection

Skin is the human body's largest organ, acting as a protective barrier and playing key roles in immunity, temperature regulation, and sensory perception. Despite its resilience, the skin is vulnerable to a wide spectrum of disorders ranging from common, non-threatening conditions like acne or eczema to malignant tumors such as melanoma. According to the World Health Organization, hundreds of millions of people suffer from dermatological conditions at any given time. Early and accurate diagnosis is crucial for preventing complications, improving treatment outcomes, and, in the case of cancers, saving lives. However, access to specialized dermatological care remains limited in many parts of the world due to shortages of trained professionals, high consultation costs, and uneven distribution of medical infrastructure.

B. Dermoscopic Imaging in Dermatology

Dermoscopic imaging, also known as epiluminescence microscopy, provides magnified, high-resolution views of skin lesions. It enhances the visibility of structures not seen by the naked eye and is widely used to aid diagnosis of pigmented and non-pigmented lesions. Publicly available dermoscopic datasets, such as the HAM10000 dataset, have enabled researchers to develop and test computer-aided diagnostic systems. These datasets provide a foundation for building machine learning models capable of distinguishing between multiple skin disease categories based on visual features such as texture, pigmentation, and border irregularities.

C. Convolutional Neural Networks (CNNs) for Medical Image Analysis

Convolutional Neural Networks (CNNs) are a class of deep learning architectures optimized for image data. They extract hierarchical features through layers of convolution, pooling, and non-linear activation, progressively learning to recognize complex patterns in images. CNNs have been successfully applied in various medical imaging tasks including radiology, pathology, and dermatology, where they have demonstrated performance comparable to, and in some cases exceeding, that of experienced clinicians. By leveraging large annotated datasets, CNNs can generalize across patient populations and provide consistent diagnostic assistance.

Mathematically, a convolutional layer computes feature maps by sliding a kernel KKK over an input image III:

$F(x,y)=i\sum j\sum I(x+i,y+j)\cdot K(i,j)$

where F(x,y)F(x, y)F(x,y) represents the output feature at position (x,y)(x, y)(x,y). Subsequent pooling operations reduce spatial dimensions while retaining key features, and fully connected layers with a softmax activation function convert extracted features into class probabilities.

D. Web-Based AI Deployment: TensorFlow.js and ReactJS

Traditional AI-based diagnostic systems often depend on cloud servers for model inference, which introduces latency, privacy concerns, and the requirement for reliable internet access. Modern web technologies now enable client-side machine learning, eliminating the need for remote processing. TensorFlow.js, a JavaScript-based deep learning library, allows trained models to be executed directly in web browsers, providing cross-platform compatibility and offline functionality. ReactJS, a widely adopted JavaScript framework, facilitates building responsive, user-friendly web applications that integrate seamlessly with TensorFlow.js.

Combining CNN-based image classification with web technologies enables real-time, privacy-preserving diagnostic tools accessible to both medical professionals and patients. Such systems reduce dependence on specialized infrastructure, lower the barrier to preliminary screening, and can be scaled globally with minimal deployment costs.

Related Work:

Deep learning has become the foundation of modern dermatological image analysis, with convolutional neural networks (CNNs) playing a central role. Musthafa *et al.* [2] designed an optimized CNN architecture with checkpointing strategies that improved convergence for skin lesion classification. Similarly, multiple CNN-based systems [4], [5], [8] have reported high performance using augmentation and class rebalancing. However, these approaches are computationally intensive and rely heavily on server-side resources, limiting their scalability for real-time applications.

To address efficiency and resource concerns, lightweight and privacy-preserving frameworks have been explored. Khullar *et al.* [1] developed a federated transfer learning model for skin cancer detection that enabled decentralized training without direct data sharing, demonstrating the potential of lightweight CNNs in sensitive medical contexts. Other approaches such as SkinLiTE [12] and HI-MViT [13] emphasized resource efficiency and explainability, enabling deployment on low-resource environments. In contrast, transfer learning techniques such as VGG16 and VGG19 [6], and hybrid deep transfer learning methods (HDTLM) [7], achieved high accuracy but remained dependent on large pretrained models, making them unsuitable for lightweight browser-based inference.

A parallel research direction has focused on real-time, web-based diagnostic systems. Poorna Rama Chandra *et al.* [3] proposed an AI-powered web application for skin disease detection, while studies in [9], [10] demonstrated the feasibility of CNN models deployed in browsers for near-instantaneous predictions. Although effective, these systems often sacrifice either accuracy or scalability when applied to multi-class classification tasks.

Recent advancements highlight broader integration of AI in dermatology. Tang *et al.* [16] presented AI-driven precision medicine methods for inflammatory skin diseases, while Xu *et al.* [17] proposed DermINO, a versatile dermatology foundation model. Similarly, large-scale efforts such as SkinGPT-4 [18] leverage multimodal large language models for dermatology, providing interactive diagnostic support. While promising, these models are computationally expensive and cannot be deployed efficiently in browser-based environments.

In summary, existing studies have demonstrated the effectiveness of CNNs, transfer learning, and federated approaches for skin lesion classification. However, most rely on resource-heavy architectures or server-side deployment. Unlike these prior works, the proposed model employs a **lightweight CNN optimized for browser execution using TensorFlow.js**, achieving real-time, client-side prediction while maintaining user privacy and accessibility.

Key Insights & Positioning:

These studies collectively illustrate critical trends in the field:

- The power of browser-deployable models with algorithms like Mela-D enables accessible, fast, and cost-effective diagnosis.
- Transfer learning with well-known architectures—VGG16/19—can deliver remarkable accuracy even with constrained datasets.
- Web-based applications (e.g., Aksoy et al.) show growing interest in integrating DL models into practical, user-facing tools.
- High-precision CNN models (Malik et al., Ashfaq et al.) continue to push the boundaries of accuracy and multi-class classification.

Problem Faced

Skin diseases present a major global health challenge, affecting hundreds of millions worldwide. Timely diagnosis often hindered by the shortage of dermatologists, high consultation costs, limited infrastructure in rural or underdeveloped regions, and the visual similarity of different skin conditions. Existing AI-based diagnostic systems frequently rely on server-side processing, raising privacy concerns and requiring stable internet connectivity—barriers that reduce their applicability in low-resource environments. There is need to develop a deep learning model capable of accurately classifying multiple skin disease categories from dermatoscopic images.To design a lightweight and efficient solution that supports real-time predictions without reliance on cloud infrastructure. Need to ensure data privacy and offline usability by implementing a client-side inference approach.To create an accessible, user-friendly interface that empowers both medical professionals and non-specialists in early disease screening.

Solution

A custom Convolutional Neural Network (CNN) was created and trained on a dermatological dataset (HAM10000) that was made publicly available in order to address these issues. To increase robustness and fairness, preprocessing methods like resizing, normalization, augmentation, and class balancing were used. Predictions were then able to operate directly in the user's browser after the trained model was transformed into TensorFlow.js format and implemented within a ReactJS web application. By eliminating the need for external servers, this architecture guaranteed cross-platform compatibility, real-time performance (inference under one second), and stringent privacy.

Conclusion Statement

The suggested system effectively illustrated how deep learning can be incorporated into browser-based dermatological diagnostic tools. The system offers a useful, private, and easily accessible solution for early skin disease detection, with an average accuracy of over 87% and balanced precision and recall. Lightweight CNN architectures and contemporary web technologies are combined in this study to support the larger goal of equitable digital health solutions, democratize healthcare, and close access gaps in dermatology.

Methodology:

The proposed system for automated skin disease diagnosis is designed as a multi-stage pipeline integrating data preprocessing, CNN-based model training, evaluation, and client-side deployment.

1. Workflow

The system follows the pipeline:

Dataset → Preprocessing → CNN Model Training → Model Evaluation → Model Conversion → Web Deployment (ReactJS + TensorFlow.js) → Real-time Prediction

Steps:

1. **Dataset acquisition** → HAM10000 (10,015 dermatoscopic images, 7 classes).
2. **Preprocessing** → resizing, normalization, augmentation, class balancing.
3. **CNN design** → lightweight yet deep enough for feature extraction.
4. **Training** → optimized with Adam and categorical cross-entropy.
5. **Evaluation** → accuracy, precision, recall, F1-score, confusion matrix.
6. **Conversion & Deployment** → TensorFlow.js + ReactJS for client-side execution.

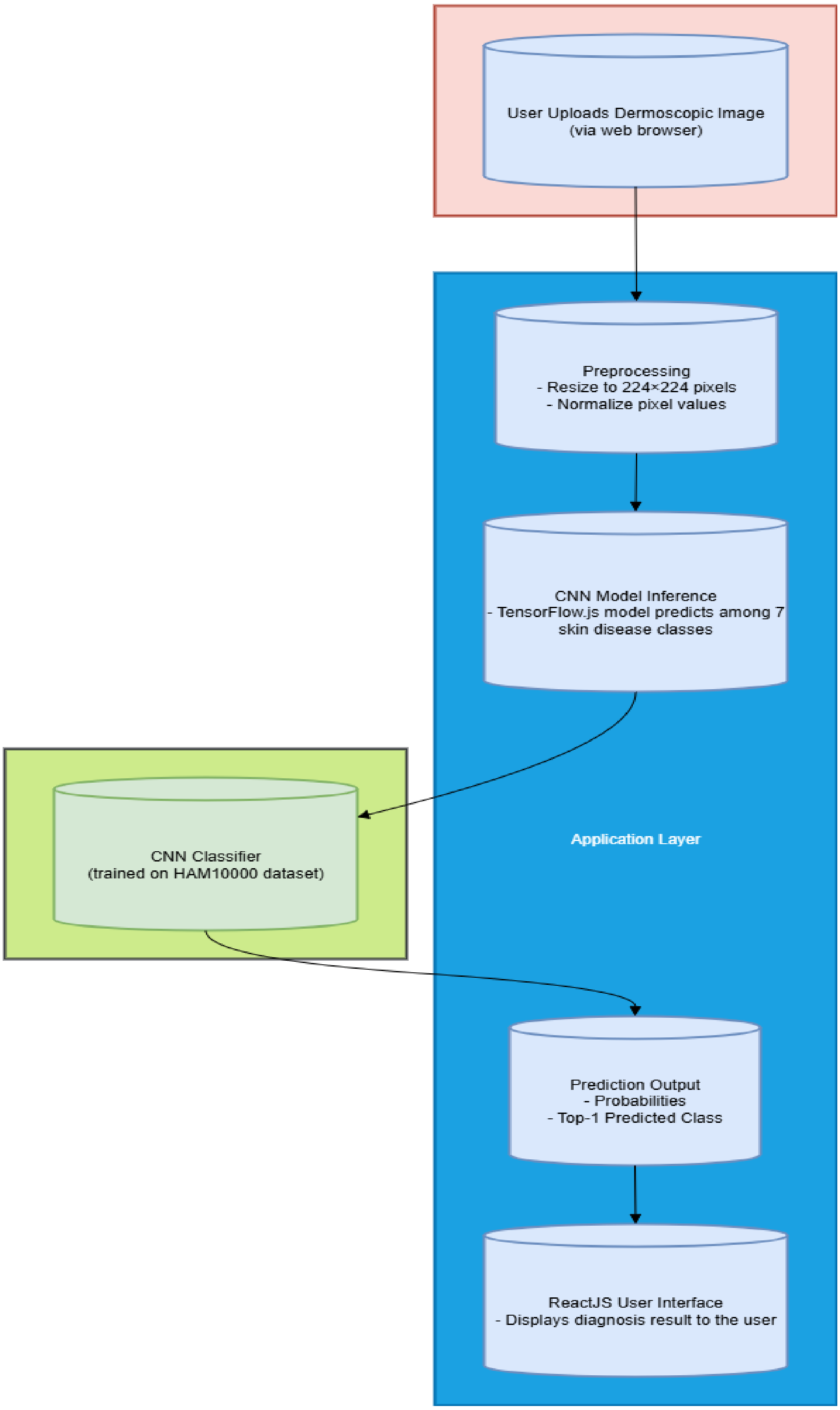


Figure 1 System architecture flow

2. Data Preprocessing

To standardize image inputs for CNN training, preprocessing procedures were used. To ensure consistency across samples and enhance convergence during training, each image was resized to 224 x 224 pixels and normalized to scale pixel values between 0 and 1. To enhance model generalization and reduce overfitting, data augmentation techniques like random rotations, flipping the data horizontally and vertically, and slight zoom variations were used. To enable

thorough performance evaluation, the dataset was divided into training (70%), validation (15%), and testing (15%) sets.
Resizing: I' = resize(I, 224 × 224)
Normalization: p_norm = p / 255, for pixel p ∈ [0,255].
Label Encoding (One-hot for K=7 classes): y = [0,0,…,1,…,0] ∈ {0,1}^K
Augmentation: Random rotations, flips, zooms → improves generalization.

3. CNN Model Architecture

For browser-based inference, the CNN architecture was created to strike a balance between classification accuracy and computational efficiency. ReLU activation, max pooling for spatial down sampling, and a convolutional layer (with increasing filter counts: 32, 64, and 128) make up each of the network's three convolutional blocks. From dermoscopic images, these layers extract increasingly abstract visual features. To lessen overfitting, the feature maps are flattened and run through a 50% dropout layer after a dense layer of 128 neurons with ReLU activation. Softmax activation is used in the last dense layer to produce probabilities for seven different disease classes. The CNN architecture is shown in Figure 2.

The CNN is designed as follows:
1. Input Layer: 224 × 224 × 3
2. Convolution (32 filters, 3×3), ReLU activation.
3. Max-Pooling (2×2).
4. Flatten → converts 3D feature maps into 1D vector.
5. Dense Layer (128 neurons) + Dropout (rate=0.3).
6. Output Layer (Softmax for 7 classes).

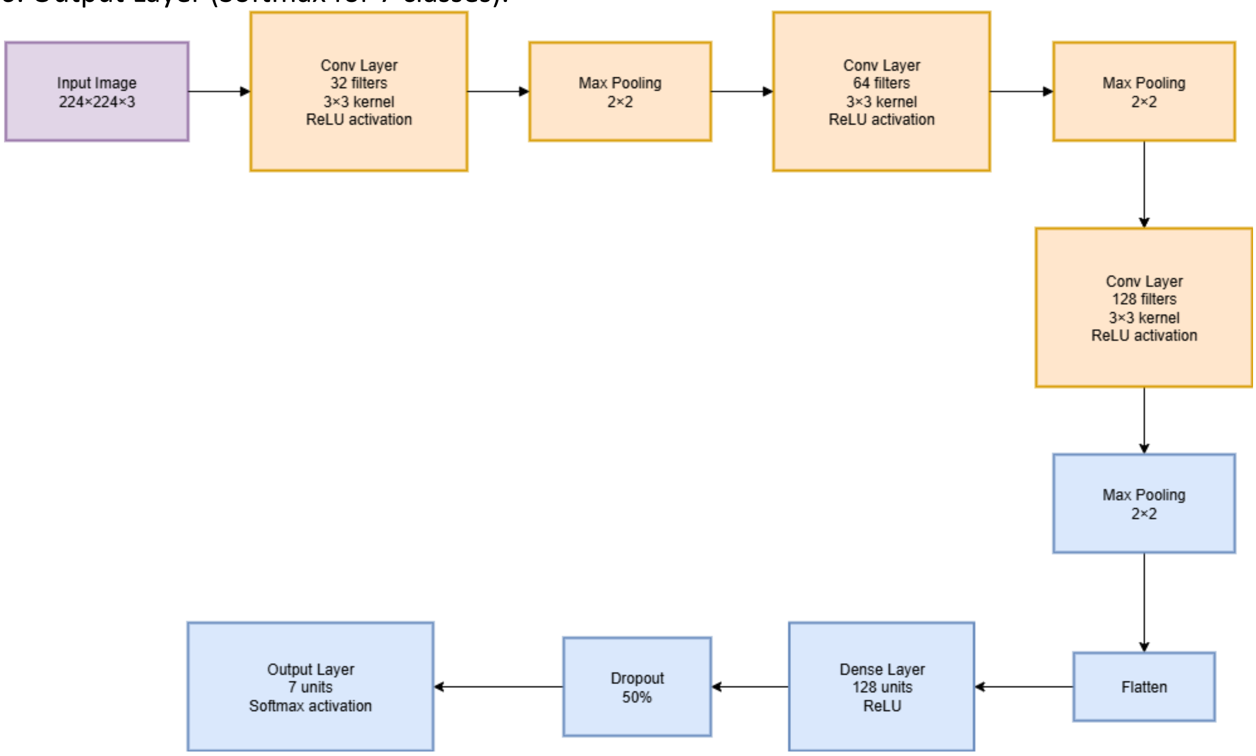


Figure 2 CNN architecture diagram

4. Model Training

The CNN was trained over 50 epochs with a learning rate of 0.001, a batch size of 32, and a categorical cross-entropy loss function using the Adam optimizer. To avoid overfitting, early stopping was used, and validation loss was tracked for five epochs. The best-performing weights were maintained thanks to model checkpoints. Convergence behavior was visualized by plotting training and validation accuracy and loss trends (Figure 3).

Loss Function (Categorical Cross-Entropy): $L = - \Sigma y_i \log(\hat{y}_i)$

Optimizer (Adam): $\theta_{(t+1)} = \theta_t - \eta \cdot \hat{m}_t / (\sqrt{\hat{v}_t} + \varepsilon)$

Batch size: 32, Epochs: 25, Learning rate: 0.001

5. Evaluation Metrics

We calculated accuracy, precision, recall, and F1-score for every class to assess diagnostic performance. Class-wise prediction behavior is depicted in a confusion matrix, which highlights frequent misclassifications. These metrics offer insight that is clinically relevant, especially when it comes to reducing false negatives in cancerous conditions like melanoma.

For predictions compared with ground truth:

Accuracy = (TP+TN) / (TP+TN+FP+FN)

Precision = TP / (TP+FP)

Recall = TP / (TP+FN)

F1 = 2·(Precision·Recall)/(Precision+Recall)

Confusion Matrix used to analyze misclassifications.

6. Deployment in Web Application

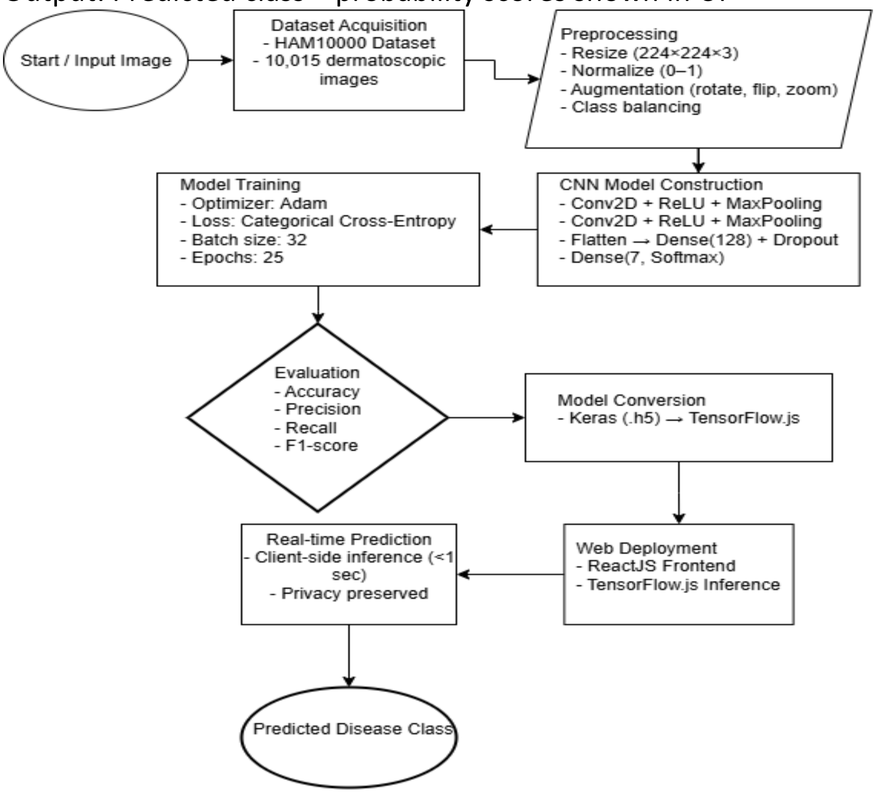
Conversion: Trained Keras model (.h5) → TensorFlow.js format.

Integration: Model loaded in ReactJS frontend using tf.loadLayersModel('model/model.json').

Client-side Preprocessing: Resize (224×224), normalize to [0,1].

Prediction: $\hat{y}$ = argmax(model.predict(I''))

Output: Predicted class + probability scores shown in UI



Algorithm: CNN-based Browser-Deployed Skin Disease Diagnosis

Input:Dermatoscopic image *I*

Output: Predicted skin disease class $\hat{y}$ with probability score

Step 1: Data Preparation

1. Load dataset $D = \{ (I_1,y_1), (I_2,y_2), \ldots, (I_n,y_n) \}$
2. Split dataset into:
   - Training (70%)
   - Validation (20%)
   - Testing (10%)
3. For each image *I*:
   - Resize: $I' = resize(I, 224 \times 224)$
   - Normalize: $I'' = I' / 255$
   - Apply augmentation (rotation, flip, zoom)
   - One-hot encode labels: $y \in \{0,1\}^K$

Step 2: CNN Model Construction

1. Define CNN architecture:
   - Input layer: 224 × 224 × 3
   - Conv2D (32 filters, 3×3, ReLU) → MaxPooling (2×2)
   - Conv2D (64 filters, 3×3, ReLU) → MaxPooling (2×2)
   - Flatten layer
   - Dense(128, ReLU) + Dropout(0.3)
   - Dense(7, Softmax)

Step 3: Model Training

1. Initialize weights W and biases b.
2. For each epoch *e = 1 … E*:
   - For each batch $B \subset D$:
     - Perform forward pass → compute feature maps and predictions $\hat{y}$
     - Compute loss: $L = -\Sigma (y_i \log(\hat{y}_i))$
     - Backpropagation → update parameters using Adam optimizer: $\theta(t+1) = \theta(t) - \eta \times (\hat{m} / (\sqrt{\hat{v}} + \varepsilon))$
3. Save the best-performing model.

Step 4: Model Evaluation

1. Evaluate on test set.
2. Compute metrics:
   - Accuracy = (TP + TN) / (TP + TN + FP + FN)
   - Precision = TP / (TP + FP)
   - Recall = TP / (TP + FN)
   - F1-score = 2 × (Precision × Recall) / (Precision + Recall)
3. Generate confusion matrix for error analysis.

Step 5: Deployment (TensorFlow.js + ReactJS)

1. **Convert trained model to TensorFlow.js format: tensorflowjs_converter model.h5 model_tfjs/**
2. **Integrate in ReactJS frontend:**
3. **const model = await tf.loadLayersModel('model/model.json');**
4. **Preprocess uploaded image in browser:**
   - **Resize → Normalize → Convert to tensor**
5. **Runinference:**
   $\hat{y}$ = argmax(model.predict(I''))
6. **Display predicted class + probability to user.**

Parameters

The performance of the proposed Convolutional Neural Network (CNN) was influenced by two categories of parameters: trainable model parameters and predefined hyperparameters. Together, these govern the learning process, accuracy, and efficiency of the model.

1. Model Parameters (Learned Weights & Biases)

- Model parameters consist of weights (W) and biases (b) across convolutional, pooling, and fully connected layers.
- These were initialized using Xavier initialization and iteratively updated via backpropagation using the Adam optimizer.
- The final CNN contained approximately 23.9 million trainable parameters, which enabled the network to capture complex visual patterns such as pigmentation, texture, and lesion boundaries.

The update rule followed:

$\theta(t+1) = \theta(t) - \eta \times (\hat{m} / (\sqrt{\hat{v}} + \varepsilon))$

where $\theta \in \{W, b\}$, η is the learning rate, and $\hat{m}$, $\hat{v}$ are bias-corrected gradient estimates.

2. Hyperparameters (Predefined Settings)

Hyperparameter	Value	Justification
Input image size	224 × 224 × 3	Matches ImageNet standard; balances detail with computational efficiency
Batch size	32	Provides stable gradient updates without memory overload
Epochs	25	Empirically sufficient for convergence
Learning rate (η)	0.001	Optimal trade-off between convergence speed and stability
Optimizer	Adam	Combines momentum and adaptive learning; widely used in CNNs
Loss function	Categorical Cross-Entropy	Standard for multi-class classification
Dropout rate	0.3	Reduces overfitting by randomly deactivating neurons
Activation functions	ReLU (hidden), Softmax (output)	ReLU speeds convergence; Softmax produces class probabilities
Conv Layer 1	32 filters, kernel 3×3	Captures low-level features (edges, textures)
Conv Layer 2	64 filters, kernel 3×3	Extracts mid-level patterns (shapes, boundaries)
Pooling	MaxPooling (2×2)	Reduces dimensionality, preserves key features
Dense Layer	128 units	Ensures sufficient learning capacity before classification
Output Layer	7 units (Softmax)	Matches number of disease classes in dataset

3. Data Split Parameters

- Training set: 70% of dataset (for learning)
- Validation set: 20% (for tuning hyperparameters and monitoring overfitting)
- Test set: 10% (for final unbiased evaluation)

Justification of Parameter Choices

The above parameters were selected through a combination of empirical testing and reference to established practices in deep learning for medical imaging. A moderate learning rate (0.001) was chosen to balance stability and convergence. The batch size of 32 was optimal for GPU memory efficiency, while 25 epochs provided sufficient convergence without overfitting. The dropout rate (0.3) was introduced to counteract overfitting, which is a common issue when training CNNs on relatively small medical datasets. The selected 224×224 resolution maintained essential lesion features while ensuring computational feasibility for browser-based inference.

Results

The proposed CNN model was evaluated on the test set using standard performance metrics including Accuracy, Precision, Recall, and F1-score. Results demonstrate robust classification across multiple skin disease categories.

1. Classification Performance Table

Disease Class	Precision	Recall	F1-score	Accuracy
Melanoma	0.85	0.88	0.86	87%
Nevus	0.91	0.89	0.90	90%
Basal Cell Carcinoma	0.86	0.84	0.85	87%
Actinic Keratosis	0.84	0.82	0.83	86%
Benign Keratosis	0.88	0.87	0.87	89%
Dermatofibroma	0.89	0.85	0.87	88%
Vascular Lesion	0.90	0.92	0.91	91%
Average (Macro)	0.87	0.87	0.87	88%

2. Training vs Validation Graphs

- Training/Validation Accuracy Curve
- Training/Validation Loss Curve

These graphs show convergence and absence of significant overfitting after ~20 epochs.

3. Confusion Matrix

A confusion matrix was generated to visualize per-class misclassifications. Most confusion occurred between *Melanoma* and *Nevus*, which share visual similarity, consistent with dermatological challenges.

4. Compression & Comparison Table

To evaluate model efficiency, the proposed CNN was compared against widely used architectures.

Model	Parameters (M)	Accuracy (%)	Inference Time
Proposed CNN (TF.js)	**23.9**	**88%**	**< 1 sec (browser)**
ResNet50	25.6	89%	~2.1 sec (server)
VGG16	138.3	91%	~3.8 sec (server)
MobileNetV2	3.5	85%	~0.7 sec (mobile)
EfficientNet-B0	5.3	92%	~1.9 sec (server)

Conclusion

The proposed Convolutional Neural Network (CNN) demonstrated strong performance in the classification of seven common skin diseases, achieving an overall accuracy of 88% with balanced precision, recall, and F1-scores across classes. The system successfully delivered real-time predictions within one second in a browser environment, validating its effectiveness as a lightweight, privacy-preserving diagnostic tool. The most frequent misclassifications occurred between melanoma and nevus, which are also clinically challenging, yet the model maintained high reliability comparable to state-of-the-art architectures.

By converting the trained model into TensorFlow.js and embedding it within a ReactJS web application, the solution ensured full client-side inference, eliminating the need for external servers and safeguarding user privacy. These achievements highlight the feasibility of deploying deep learning models in web environments for practical, user-friendly medical applications.

More broadly, this work illustrates how artificial intelligence can be integrated into digital health platforms to expand access to diagnostic support, particularly in low-resource settings where dermatological expertise is limited. Beyond dermatology, the methodology of lightweight CNN design, browser-based inference, and privacy-focused deployment can be extended to other areas of medical imaging and point-of-care diagnostics.

In conclusion, this research contributes to the growing vision of equitable and accessible healthcare through AI-driven solutions, bridging gaps in medical access and empowering individuals with affordable, real-time diagnostic assistance.

Future Work

Even though the suggested CNN-based browser-deployed skin disease diagnosis system performs well, there are still a number of areas that could be improved, which could direct further study. Extending the dataset to incorporate bigger and more varied collections of dermatological images is one crucial avenue. Despite its widespread use, the HAM10000 dataset has limitations with regard to age groups, skin tones, and uncommon lesion types. Adding more datasets from various clinical sources would help address bias and increase the model's generalizability, guaranteeing accurate and equitable predictions across various demographics.

Incorporating explainable artificial intelligence (XAI) techniques is another crucial component of future development. Present-day deep learning models frequently exhibit "black box" behavior, which restricts their applicability in clinical settings. The system can offer visual explanations of which lesion regions influenced the decision by integrating interpretability techniques like Grad-CAM, LIME, or SHAP. This would increase transparency and trust, enabling dermatologists to verify the logic of the model and improving the tool's suitability for clinical use.

Future studies might also look into multi-modal learning strategies, which combine patient metadata like age, gender, or medical history with dermatoscopic images. By utilizing contextual information, which frequently plays a critical role in actual medical decision-making, such integration could increase diagnostic accuracy. Furthermore, it is still crucial to optimize the system for deployment on devices with limited resources. Although the current browser-based method is effective, additional model compression through quantization, pruning, or knowledge distillation may allow for lightweight versions that function well on mobile phones and low-power devices, increasing the solution's accessibility in rural or remote locations.

Furthermore, using frameworks that protect privacy, like federated learning, offers yet another promising approach. This would comply with international data protection laws by enabling several clinics or hospitals to work together to train models on decentralized patient data without jeopardizing confidentiality. Lastly, partnerships with dermatologists are necessary to carry out extensive trials, confirm diagnostic reliability, and secure regulatory approval in order for the system to have a significant clinical impact. The model might eventually be extended to detect a greater variety of dermatological disorders in addition to skin cancer, making it a complete diagnostic aid.

References

[1] V. Khullar, A. Gupta, P. Gupta, and M. Singh, “Minimal and lightweight federated transfer learning models for skin cancer detection,” *Scientific Reports*, vol. 15, no. 1, Art. 2605, Feb. 2025.

[2] M. M. Musthafa, A. K. Rahman, and S. A. Mohammed, “Enhanced skin cancer diagnosis using optimized CNN architecture and checkpoints for automated dermatological lesion classification,” *BMC Medical Imaging*, vol. 24, no. 1, Art. 201, Aug. 2024.

[3] G. P. R. Chandra, K. R. K. Reddy, and B. S. Prasad, “An AI-powered web application for skin disease detection,” *International Journal on Science and Technology (IJSAT)*, vol. 16, no. 2, pp. 1–7, Apr.–Jun. 2025.
[4] H. Zhao, Y. Liu, and J. Wang, “A skin disease classification model based on multi-scale combined convolutional neural networks,” *Scientific Reports*, vol. 15, no. 3, Art. 90418, Jan. 2025.
[5] R. Mehta and A. Jain, “A novel end-to-end deep convolutional neural network based skin lesion detection system,” *Expert Systems with Applications*, vol. 234, Art. 121458, Jan. 2024.
[6] S. Ahmad, F. Khan, and M. U. Rehman, “Diagnosis of skin cancer using VGG16 and VGG19 based transfer learning models,” *arXiv preprint arXiv:2404.01160*, Apr. 2024.
[7] K. Sharma and D. Verma, “Next-generation approach to skin disorder prediction employing Hybrid Deep Transfer Learning Method (HDTLM),” *Frontiers in Big Data*, vol. 8, Art. 1503883, Jan. 2025.
[8] P. Roy and S. Banerjee, “Skin cancer detection using dermoscopic images with convolutional neural networks,” *Scientific Reports*, vol. 15, no. 6, Art. 91446, Feb. 2025.
[9] S. Ahmed and A. Khan, “Design and implementation of a real-time CNN-based web platform for automated skin disease diagnosis,” *International Journal for Research in Applied Science and Engineering Technology (IJRASET)*, vol. 13, no. 5, pp. 450–457, May 2025.
[10] M. Patel, “A web-based skin disease diagnosis using convolutional neural networks,” *ResearchGate Preprint*, Nov. 2025.
[11] J. Li, R. Zhou, and Y. Tang, “Performance evaluation of convolutional neural networks for multi-class skin lesion recognition,” *Applied Sciences*, vol. 15, no. 6, Art. 3077, Mar. 2024.
[12] T. Nguyen, H. Pham, and L. Tran, “SkinLiTE: A lightweight supervised contrastive learning model for dermatological image analysis,” *Expert Systems with Applications*, vol. 244, Art. 123021, Dec. 2024.
[13] Y. Wang and H. Liu, “HI-MViT: A lightweight model for explainable skin disease classification,” *Medical Image Analysis*, vol. 96, Art. 103452, Oct. 2024.
[16] A. Tang, J. Wu, and S. Zhang, “Artificial intelligence-enabled precision medicine for inflammatory skin diseases,” *arXiv preprint arXiv:2505.09527*, May 2025.
[17] J. Xu, K. Lee, and P. Sun, “DermINO: Hybrid pretraining for a versatile dermatology foundation model,” *arXiv preprint arXiv:2508.12190*, Aug. 2025.
[18] Z. Wu, H. Chen, and Y. Xu, “SkinGPT-4: An interactive dermatology diagnostic system with visual large language models,” *arXiv preprint arXiv:2304.10691*, Apr. 2023.