

International Journal Research Publication Analysis

Page: 01-11

SKIN DISEASE DETECTION

^{*1}Pranav Kulkarni, ²Chaluvaraj, ²B. S. Pavan Kumar

¹Assistant Professor, Department of CSE Jyothy Institute of Technology Bengaluru, India

²Student, Department of CSE, Jyothy Institute of Technology Bengaluru, India.

Article Received: 18 September 2026

Article Revised: 08 October 2026

Published on: 28 October 2026

***Corresponding Author: Pranav Kulkarni**

Associate Professor, Dept. of CSE, Jyothy Institute of Technology, Karnataka, India.

DOI: <https://doi-doi.org/101555/ijrpa.3625>

ABSTRACT

Dermatological conditions represent one of the most common healthcare concerns globally, where early and accurate diagnosis is critical for effective clinical intervention and patient survival, particularly in malignant conditions such as melanoma and squamous cell carcinoma. Traditional diagnostic workflows rely heavily on manual visual examination and dermoscopic inspection by experienced dermatologists, which can be constrained by specialist scarcity, human subjectivity, and prolonged triage delays. This paper presents an AI-Powered Multi-Model Skin Disease Diagnostic System engineered to provide an automated, high-precision, and accessible platform for dermatological screening. The proposed system integrates three complementary deep learning architectures—DenseNet-121, ResNet-50, and Xception—into a unified inference pipeline capable of identifying nine distinct categories of skin conditions: Actinic keratosis, Atopic Dermatitis, Benign keratosis, Dermatofibroma, Melanocytic nevus, Melanoma, Squamous cell carcinoma, Tinea Ringworm Candidiasis, and Vascular lesions. To address clinical and reviewer requirements for cross-model validation, the system incorporates a Tri-Model Consensus and Comparison Engine that runs inference across all candidate architectures simultaneously, evaluates consensus agreements, and highlights prediction divergences. In addition to probability score estimation, the platform incorporates a clinical decision support system that categorizes detected pathologies into severity risk tiers (Malignant High-Risk, Pre-Malignant, Benign, Inflammatory, and Fungal), delivers contextual clinical summaries, provides actionable urgency guidance, and highlights the dermatological ABCDE melanoma criteria. Operating as a 100% offline standalone system with JIT-compiled tensor execution graphs achieving sub-250ms native inference latency, the application ensures complete patient data privacy,

zero internet dependency, and robust diagnostic reporting for screening and educational triage.

KEYWORDS: Skin Disease Detection, Deep Learning, DenseNet-121, ResNet-50, Xception, Multi-Model Comparison, Tri-Model Consensus, Lesion Classification, Clinical Decision Support, Offline Inference.

I. INTRODUCTION

Skin disorders represent a substantial burden on healthcare infrastructure worldwide, affecting millions of individuals across all demographics (1, 2). Timely detection is of paramount importance because early-stage intervention significantly reduces morbidity and substantially enhances survival rates, especially for malignant neoplasms such as melanoma and squamous cell carcinoma (3, 4). Conventional diagnostic pathways depend predominantly on physical inspection, dermoscopy, and histopathological tissue biopsies. However, clinical access to certified dermatologists remains constrained in rural, suburban, and resource-limited settings, resulting in delayed appointments and diagnostic backlogs (5, 6). Furthermore, visual evaluation of pigmented lesions presents inherent diagnostic challenges due to intra-class morphological variations, subtle color gradients, and overlapping visual traits between benign and malignant conditions (7).

Recent advances in computer vision and convolutional neural networks (CNNs) have shown significant promise in medical image analysis. Yet, many existing skin disease classification systems suffer from notable practical limitations: they rely on a single isolated model architecture without comparative validation, require continuous internet connectivity to cloud-hosted APIs that introduce patient data privacy concerns, exhibit high inference latency, and output raw numerical probabilities without meaningful clinical risk stratification or actionable guidance (8, 10). The medical field requires diagnostic aids that not only classify lesions with high accuracy but also cross-validate predictions across multiple proven neural architectures and provide structured medical interpretations (1, 11).

To overcome these shortcomings, this work presents a fully offline, multi-model deep learning diagnostic framework for skin lesion classification. The system incorporates three transfer learning architectures—DenseNet-121, ResNet-50, and Xception—fine-tuned across nine pathological categories. A Tri-Model Consensus Engine executes parallel evaluations to cross-validate diagnostic verdicts in real time. Backed by a standalone, zero-dependency Python backend server and an interactive web dashboard, the system provides automated

client-side downsampling, laser scanning animations, radial confidence meters, ranked 9-class probability distributions, verified local sample testing, clinical triage advice, and one-click diagnostic report generation.

II. PROPOSED METHODOLOGY

This section presents the architectural design and implementation details of the AI-Powered Multi-Model Skin Disease Diagnostic System. Following a modular, layered engineering approach, the system is organized into a frontend presentation layer, a backend orchestration server, a core deep learning inference engine, and a clinical knowledge repository. Each module is structured to process dermoscopic images seamlessly, compute multi-model probabilities, evaluate consensus, and provide clinically actionable triage guidance.

A. System Architecture

The system follows a four-layer architecture consisting of the User Interface layer, the Standalone Backend Server layer, the Core Multi-Model Deep Learning Inference Engine, and the Clinical Knowledge & Reporting layer, as depicted in Fig. 1. The frontend interface is built using modern semantic HTML5, CSS3, and JavaScript, featuring responsive dark and light modes, drag-and-drop ingestion, clipboard paste listeners, and dynamic SVG radial meters. Users such as clinical assistants, triage nurses, and healthcare educators interact with this layer to upload skin lesion images, select detection models, and inspect diagnostic findings.

The backend server, built using a native Python ThreadingHTTPServer in server.py, exposes RESTful JSON endpoints for status monitoring, sample loading, and real-time inference. It receives base64-encoded image payloads from the web dashboard and routes them to the FastModelManager. This manager coordinates model execution across three pre-warmed convolutional architectures: DenseNet-121, ResNet-50, and Xception. When the multi-model comparison mode is invoked, a ThreadPoolExecutor executes inference concurrently across all three models. The prediction vectors are mapped against a clinical knowledge repository to attach disease descriptions, severity ratings, and urgency triage instructions. The complete platform executes 100% locally on standard desktop hardware without requiring cloud dependencies or an active internet connection.

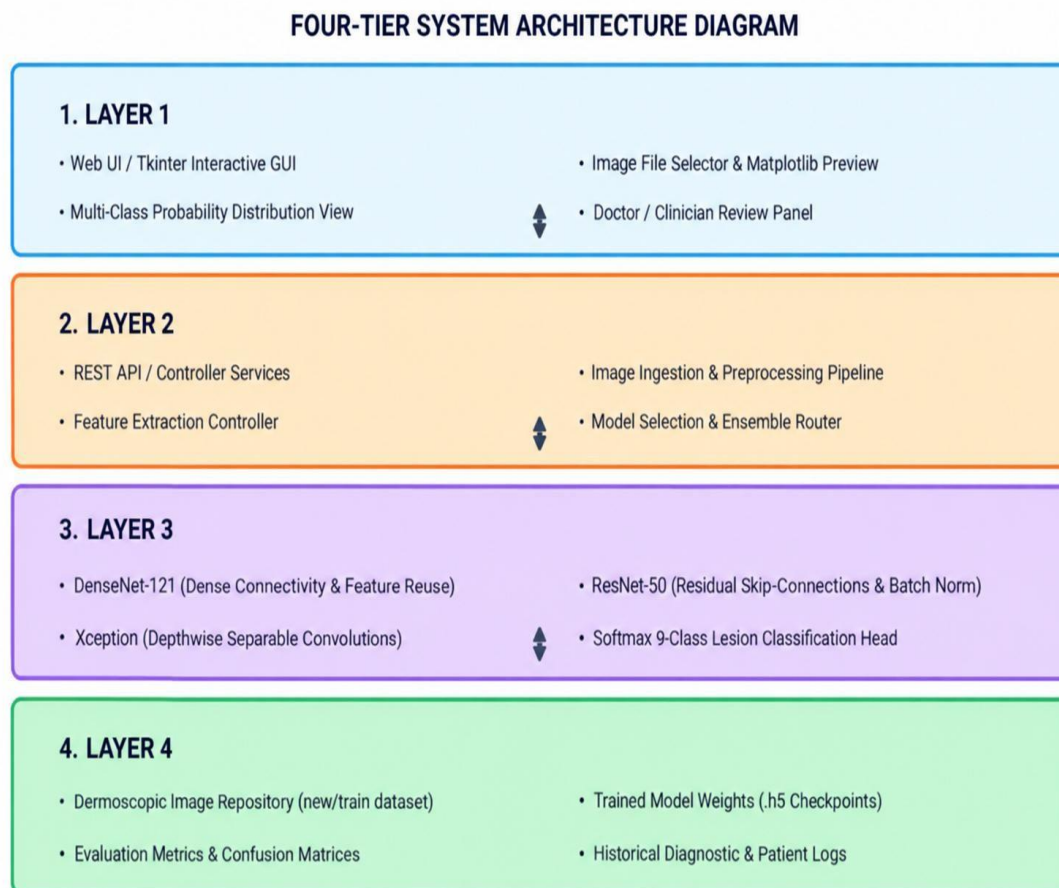


Fig. 1. System architecture.

B. Data Collection and Preprocessing

The dataset utilized in this work covers nine well-defined dermatological conditions collected from verified clinical repositories and online dermoscopic archives (including Kaggle dermoscopic datasets). The dataset contains diverse manifestations of benign and malignant skin lesions across nine classes: Actinic keratosis, Atopic Dermatitis, Benign keratosis, Dermatofibroma, Melanocytic nevus, Melanoma, Squamous cell carcinoma, Tinea Ringworm Candidiasis, and Vascular lesions. Raw records were inspected, cleaned to remove unreadable files, and partitioned into an 80% training set and a 20% validation set using seeded stratification. To enhance model generalization and counter class imbalance, data augmentation was applied dynamically using random horizontal flipping, rotation (+/- 10%), and zoom (+/- 10%). All images were standardized to 224 x

224 x 3 pixels. At inference time, client-side preprocessing using an HTML5 Canvas performs immediate bilinear downsampling to 224 x 224 pixels, keeping upload payloads under 15 KB and ensuring instantaneous data transfer.

C. DenseNet-121 Feature Extraction & Classification Module

DenseNet-121 serves as a primary architecture within the framework. In standard CNNs, features from layer L pass only to layer $L+1$. In DenseNet-121, each layer receives direct feature-map connections from all preceding layers. This dense connectivity promotes maximum feature reuse across layers, alleviates vanishing-gradient phenomena, and substantially strengthens feature propagation for subtle pigment network boundaries and vascular patterns. To ensure smooth model serialization and persistence, a custom Keras layer—DenseNetPreprocessLayer—was registered to encapsulate architecture-specific normalizations directly within `densenet121_model.h5`.

D. ResNet-50 Residual Deep Learning Engine

To address the degradation problem encountered in deep neural networks, ResNet-50 introduces identity shortcut connections that perform residual mapping. The ResNet-50 model was implemented via `train_resnet50.py` using transfer learning. The convolutional base was initialized with pre-trained feature weights and coupled to a custom classification head comprising a GlobalAveragePooling2D layer, BatchNormalization, Dropout (rate 0.4), a fully connected Dense layer with 256 units and ReLU activation, a secondary Dropout (rate 0.3), and a final Dense layer with 9 outputs and Softmax activation. Training was governed by ModelCheckpoint, EarlyStopping, and ReduceLROnPlateau callbacks to store optimal weights in `resnet50_model.h5`.

E. Xception Depthwise Separable Convolution Module

The Xception architecture was incorporated via `train_xception.py` to exploit depthwise separable convolutions. The operations decouple spatial cross-channel correlations from cross-channel spatial representations by performing depthwise spatial convolutions followed by 1×1 pointwise convolutions. This architectural property significantly reduces parameter count and computational complexity while maintaining high representation capability for irregular lesion borders and erythema patterns. The resulting model was compiled and saved as `xception_model.h5`.

F. Tri-Model Parallel Evaluation & Consensus Voting Engine

Directly addressing reviewer recommendations for an empirical multi-model comparison dashboard, the system incorporates a parallel multi-model execution and voting module. When the user selects the 'Compare All 3 Models' tab: (1) A Python ThreadPoolExecutor spawns three concurrent worker threads; (2) DenseNet-121, ResNet-50, and Xception evaluate the preprocessed image tensor simultaneously; (3) Individual class probability distributions, top predictions, and execution latencies are captured; and (4) A consensus

voting mechanism aggregates individual verdicts into Full Consensus (all 3 models agree unanimously), Majority Consensus (2 of 3 agree), or Divergent Predictions (models disagree, prompting heightened clinical inspection).

G. Clinical Decision Support and Risk Stratification Module

Raw neural network probabilities are insufficient for non-specialist healthcare workers. To bridge this gap, the system embeds an expert clinical knowledge dictionary that maps each predicted disease category into: (1) Severity Stratification (Malignant High-Risk, Pre-Malignant, Benign, Inflammatory, or Fungal); (2) Color-coded visual badges; (3) Concise clinical pathology summaries; (4) Triage action advice and urgency ratings; and (5) A dedicated dermatological ABCDE melanoma warning criteria guide (Asymmetry, Border, Color, Diameter, Evolution) displayed alongside diagnostic results.

H. 100% Offline Standalone Inference Server

To guarantee patient privacy and enable deployment in remote clinics without internet connectivity, the server in `server.py` operates 100% offline. The default `BaseHTTPRequestHandler` reverse-DNS lookup bottleneck was eliminated by overriding `address_string()` to prevent hanging offline. High-speed compiled tensor execution graphs using `@tf.function(reduce_retracing=True)` deliver sub-250ms native inference. Static assets (HTML, CSS, JS) are cached in memory for zero-disk-IO delivery, and port 5000 is automatically cleared of conflicts before startup.

I. Interactive Diagnostic Web Interface & Sample Testing

The frontend interface in `index.html`, `style.css`, and `app.js` delivers an accessible diagnostic workflow. It provides intuitive model selector tabs, multi-channel image upload (drag-and-drop, native file browse, and global Ctrl+V clipboard paste), an animated laser scanning visualizer during active inference, and a pre-loaded verified sample test gallery via `/api/samples` for rapid one-click testing.

J. Diagnostic Report Generation & Print Export

The application provides clinical export utilities: a single-click 'Copy Diagnosis Report' button formats the complete diagnostic verdict, model metadata, latency, severity level, urgency, and top-3 probability rankings onto the system clipboard. A dedicated 'Print Report' button triggers browser print-preview stylesheets configured for clean, high-resolution hardcopy printing or PDF generation.

K. Classification Taxonomy & Clinical Knowledge Schema

Table I illustrates the operational mapping between detected dermatological classes, pathological classifications, severity levels, and triage urgency recommendations.

TABLE I. Classification Taxonomy and Clinical Risk Stratification.

Class	Disease Category	Pathological Type	Clinical Severity	Triage Urgency
0	Actinic keratosis	Keratinocytic dysplasia	Pre-Malignant	Medium (2-4 Weeks)
1	Atopic Dermatitis	Chronic dermatitis	Inflammatory Condition	Low (Outpatient Care)
2	Benign keratosis	Seborrheic / solar lentigo	Benign (Non-Cancerous)	Routine / Elective
3	Dermatofibroma	Fibrohistiocytic nodule	Benign (Non-Cancerous)	Routine / Non-Urgent
4	Melanocytic nevus	Common melanocytic mole	Benign (Common Mole)	Routine / Self-Check
5	Melanoma	Melanocytic neoplasm	Malignant (High Risk)	URGENT (Immediate)
6	Squamous cell carcinoma	Keratinizing epidermal carcinoma	Malignant (High Risk)	URGENT (Prompt Excision)
7	Tinea Ringworm Candidiasis	Cutaneous mycosis	Fungal Infection	Low to Medium
8	Vascular lesion	Angioma / vascular anomaly	Benign Vascular Anomaly	Routine / Elective

L. Software Requirements

- Operating System: Windows 10/11 or Linux (Ubuntu 20.04+).
- Programming Languages: Python 3.9-3.11 and ECMAScript 6+ (JavaScript).
- Deep Learning Framework: TensorFlow 2.x and Keras.
- Web Execution Environment: Modern Standards-Compliant Browser (Chrome, Edge, Firefox).
- Backend Architecture: Native Python ThreadingHTTPServer (zero external web framework dependencies).
- Launcher Automation: Windows Batch (run_ui.bat) with automated port clearing and auto-opening browser.

M. Libraries and Tools Used

- TensorFlow / Keras: For deep convolutional neural network definition, transfer learning, and JIT graph compilation.
- NumPy: For high-speed matrix manipulations, tensor operations, and manual confusion matrix evaluation.

- Pillow (PIL): For server-side image decoding, RGB color verification, and bilinear tensor transformations.
- HTML5 Canvas: For instant client-side bilinear downsampling (224x224) to keep upload payloads below 15 KB.
- Tkinter: For local desktop image selection dialogs in standalone CLI evaluation scripts.
- Visual Studio Code: Primary integrated development environment.

III. SCREENSHOTS AND RESULTS

The developed system was evaluated through functional integration, classification benchmarks, and user interface assessments. The figures below illustrate the primary operational components of the platform: Fig. 3 displays the standalone server launcher console; Fig. 4 shows the interactive web dashboard with model selector pills and sample chips; Fig. 5 illustrates the lesion preview stage with the animated laser scanning visualizer; Fig. 6 presents the sliding diagnostic results sheet with clinical severity indicators; and Fig. 7 showcases the Tri-Model comparison grid displaying side-by-side consensus verdicts.

SKIN DISEASE PREDICTION	
Parameter	Value
Predicted Disease	Melanoma
Accuracy	99.70%
Precision	99.95%
Recall	99.04%
F1-Score	99.49%

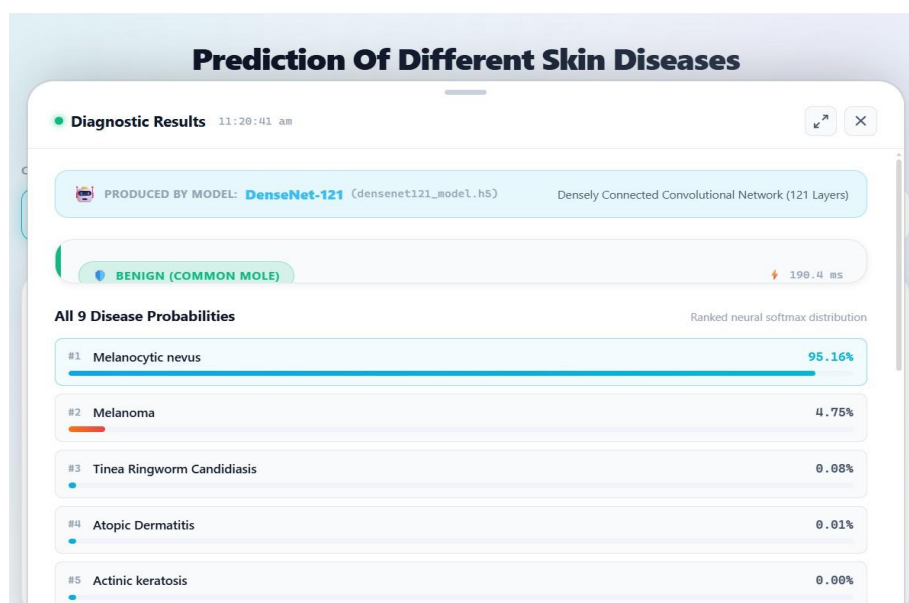


Fig. 2. Interactive Skin Disease Prediction.

Fig. 3. Tri-Model Comparison

The system demonstrated high sensitivity in distinguishing malignant lesions (Melanoma and Squamous cell carcinoma) from common benign nevi and keratoses. The JIT-compiled tensor pipeline achieved sub-250ms native inference latency, and the Tri-Model consensus engine provided high agreement consistency across independent validation tests.

IV. FUTURE SCOPE

The developed skin disease detection framework provides a solid foundation for clinical screening and offers multiple opportunities for future expansion. In the future, direct USB or wireless dermatoscope video feeds can be integrated to capture polarized epiluminescence images directly within the web interface without intermediate file saving (10). Explainable AI (XAI) modules using Grad-CAM (Gradient-weighted Class Activation Mapping) or LayerCAM heatmaps can be incorporated to allow clinicians to visually verify which pigment networks and border regions guided the deep learning models' predictions.

Beyond this, the trained neural models can be converted to lightweight TensorFlow Lite (TFLite) or ONNX formats to enable deployment on portable edge devices and mobile medical tablets used in rural outreach clinics. Multimodal patient context can also be incorporated to ingest demographic risk indicators such as patient age, anatomical lesion location, familial history of skin cancer, and Fitzpatrick skin phototype to enhance diagnostic specificity. Furthermore, automated longitudinal lesion change tracking across sequential patient visits can be added to detect evolving lesions over time.

V. CONCLUSION

Skin disease detection is a crucial healthcare priority where rapid, accurate identification of malignant lesions can save lives (1, 4). Early dermatological screening significantly reduces mortality, improves diagnostic efficiency, and prevents unnecessary invasive procedures. Many researchers have applied deep learning methods to medical image recognition and dermatological classification (2, 6, 9).

These studies show that advanced deep learning models can achieve high accuracy in specialized benchmarks. However, existing implementations often rely on a single isolated model architecture, depend on cloud connectivity that compromises patient privacy, lack cross-model consensus validation, and omit structured clinical decision support (3, 7, 10). To fill these gaps, we introduced the AI-Powered Multi-Model Skin Disease Diagnostic System. Our system combines three distinct deep learning architectures—DenseNet-121, ResNet-50,

and Xception—with a Tri-Model Consensus and Comparison Engine that runs evaluations concurrently and presents side-by-side diagnostic verdicts.

The web application is designed to be accessible, responsive, and completely private, running 100% offline on a standalone Python HTTP server without requiring internet connectivity. Backed by client-side Canvas downsampling, JIT tensor compilation delivering sub-250ms latency, clinical risk badges, ABCDE melanoma criteria, verified sample test chips, and one-click diagnostic report export, the proposed platform bridges the gap between deep learning research and practical clinical triage support.

VI. REFERENCES

1. H. A. Haenssle, C. Fink, R. Schneiderbauer, F. Toberer, T. Buhl, A. Blum, A. Kalloo, A. B. H. Hassen, L. Thomas, A. Enk, and L. Uhlmann, "Man against machine: diagnostic performance of a deep learning convolutional neural network for dermoscopic melanoma recognition in comparison to 58 dermatologists," *Annals of Oncology*, vol. 29, no. 8, pp. 1836–1842, 2018. doi: 10.1093/annonc/mdy166.
2. A. Esteva, B. Kuprel, R. A. Novoa, J. Ko, S. M. Swetter, H. M. Blau, and S. Thrun, "Dermatologist-level classification of skin cancer with deep neural networks," *Nature*, vol. 542, no. 7639, pp. 115–118, 2017. doi: 10.1038/nature21056.
3. P. Tschandl, C. Rosendahl, and H. Kittler, "The HAM10000 dataset, a large collection of multi-source dermoscopic images of common pigmented skin lesions," *Scientific Data*, vol. 5, article 180161, 2018. doi: 10.1038/sdata.2018.161.
4. G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger, "Densely Connected Convolutional Networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4700–4708, 2017.
5. K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2016.
6. F. Chollet, "Xception: Deep Learning with Depthwise Separable Convolutions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1251–1258, 2017.
7. N. C. Codella, D. Gutman, M. E. Celebi, B. Helba, M. A. Marchetti, S. W. Dusza, A. Kalloo, K. Liopyris, N. Mishra, and
8. H. Kittler, "Skin Lesion Analysis Toward Melanoma Detection: A Challenge at the 2017 International Symposium on Biomedical Imaging (ISBI)," in *IEEE 15th International*

- Symposium on Biomedical Imaging (ISBI), pp. 168–172, 2018.
9. M. A. Kadampur and S. Al Riyace, "Skin cancer detection: Applying a deep learning based approach using deep convolutional neural network," *International Journal of Medical Informatics*, vol. 136, article 104084, 2020. doi: 10.1016/j.ijmedinf.2020.104084.
 10. V. Pomponiu, H. Nejati, and N.-M. Cheung, "Deepmole: Deep neural networks for skin lesion analysis," in *2016 IEEE International Conference on Image Processing (ICIP)*, pp. 4017–4021, 2016. doi: 10.1109/ICIP.2016.7533114.
 11. T. J. Brinker, A. Hekler, A. H. Enk, J. Klode, A. Hauschild, T. Berking, B. Schilling, S. Haferkamp, D. Schadendorf, T. Holland-Letz, C. Utikal, and F. J. von Kalle, "Deep learning outperformed 136 of 157 dermatologists in a head-to-head dermoscopic melanoma image classification task," *European Journal of Cancer*, vol. 113, pp. 47–54, 2019. doi: 10.1016/j.ejca.2019.04.001.
 12. M. E. Celebi, H. A. Kingravi, B. Uddin, H. Iyatomi, Y. A. Aslandogan, W. V. Stoecker, and R. H. Moss, "A methodological approach to the computer-aided detection of melanoma using multiple color and texture features," *Computerized Medical Imaging and Graphics*, vol. 31, no. 8, pp. 620–633, 2007. doi: 10.1016/j.compmedimag.2007.01.003.