الجمهورية الجزائرية الديمقراطية الشعبية
PEOPLE'S DEMOCRATIC REPUBLIC OF ALGERIA

وزارة التعليم العالي و البحث العلمي
MINISTRY OF HIGHER EDUCATION AND SCIENTIFIC RESEARCH

جامعة عمّار ثليجي بالأغواط
AMAR TELIDJI UNIVERSITY OF LAGHOUAT

كلية العلوم
FACULTY OF SCIENCES

DEPARTMENT OF COMPUTER SCIENCE

Master's Thesis

Field : Mathematics and Computer Science
Specialty : Computer Science
Option : Data Science and Artificial Intelligence

By : Taleb Ahmed Houssam Eddine

TOPIC

Deep Learning-Based AI Integration in Dual Mobile Applications for Skin Disease Detection and Doctor-Patient Appointment Management

---

Defended publicly on June 29,2025, before a jury composed of :

Name	Rank	Role
Pr. Lagraa Nasreddine	M.C.A.	President
Dr. Boudouh Saida Sarra	M.C.A.	Examiner
Dr. Magidi Hicham	M.C.A.	Examiner
Pr. Bouakkaz Mustapha	Prof.	Supervisor

Academic Year 2024/2025

DEDICATIONS

In the name of Allah, the Most Gracious, the Most Merciful. To my dear parents, for their unconditional love, prayers, and sacrifices. To everyone who supported me and believed in me throughout this journey.

ACKNOWLEDGMENTS

All praise and thanks be to Allah for granting me the strength, patience, and perseverance to complete this thesis. I am profoundly grateful to my parents for their endless support and encouragement. My sincere thanks also go to my family, friends, my supervisor, and all who have guided and assisted me during this work.

ABSTRACT

Skin diseases range from minor conditions to serious cancers like melanoma, where early detection is crucial for effective treatment and survival. This thesis presents a comprehensive AI-driven system for skin disease classification using the HAM10000 dataset, which includes over 10,000 dermatoscopic images across seven lesion categories. At its core is a specially designed Convolutional Neural Network (CNN) enhanced with residual blocks and attention mechanisms, trained on a Colab Pro A100 GPU. The model outperformed popular pretrained networks—ResNet50, DenseNet121, and EfficientNetB0—achieving a validation accuracy of 85.08, while the best pretrained model reached only 59.63. For practical deployment, the model was converted to TensorFlow Lite and embedded into two cross-platform Flutter-Firebase mobile apps: a Patient App for AI-based skin image analysis and appointment booking, and a Doctor App for appointment management and AI-assisted diagnosis. This work delivers an efficient, scalable solution for early skin disease detection and smart healthcare support.

ملخص

تتراوح أمراض الجلد بين حالات بسيطة وأخرى خطيرة مثل الورم الميلانيني، ويُعد الكشف المبكر عنها أمرًا بالغ الأهمية لإنقاذ الأرواح وتحسين فرص العلاج. في هذا البحث، تم تطوير نظام ذكي متكامل لتشخيص أمراض الجلد اعتمادًا على مجموعة بيانات HAM10000 التي تضم أكثر من 10,000 صورة لسبعة أنواع من الآفات الجلدية. يعتمد النظام على شبكة عصبية مصممة خصيصًا تتضمن التفافية كتلًا متبقية وآليات انتباه، وقد تم تدريبها باستخدام معالج رسوميات قوي من نوع Colab Pro A100، محققة دقة تدريب بلغت 90.99 ودقة تحقق 85.08. تمت مقارنة الأداء بثلاث شبكات جاهزة شائعة، وهي: ResNet50 و DenseNet121، EfficientNetB0، إلا أن أيًا منها لم يتفوق على النموذج المصمم، حيث لم تتجاوز أفضلها (ResNet50) دقة تحقق 59.63. ولتطبيق النظام عمليًا، تم تحويله إلى نسخة خفيفة باستخدام TensorFlow Lite وتم دمجه في تطبيقين للهاتف المحمول باستخدام Flutter و Firebase؛ الأول مخصص للمرضى يتيح حجز المواعيد والتشخيص الآلي عبر رفع صورة، والثاني للأطباء لإدارة المواعيد والملفات واستخدام الذكاء الاصطناعي في التشخيص، مما يجعل النظام حلاً ذكيًا وعمليًا للكشف المبكر عن أمراض الجلد وتحسين إدارة الرعاية الصحية. يقدم هذا العمل حلاً كاملاً وفعّالًا لتشخيص أمراض الجلد مبكرًا، مع نظام ذكي لحجز المواعيد الطبية عبر الهاتف المحمول.

CONTENTS

CHAPTER 1
INTRODUCTION

1.1 Background and Motivation

Skin conditions range from harmless moles to serious melanoma. In 2022, about 330 000 new melanoma cases were reported worldwide, causing nearly 60 000 deaths . Finding cancer early greatly improves outcomes. Even non life threatening issues like eczema and psoriasis can cause pain, affect daily life, and drive up healthcare costs especially in areas with few specialists or limited equipment.

Smartphone AI can help close this gap. A simple app that screens skin lesions on the device can give instant feedback without internet access. This speeds up care by flagging high risk cases for specialist review and guiding patients on next steps. When screening and appointment booking happen in one app, patients and doctors enjoy a smoother, more connected experience.

1.2 Problem Statement

- **Limited access:** Many areas do not have enough dermatologists or proper tools, which delays skin problem diagnoses.
- **General doctor difficulties:** Doctors who are not skin specialists may make mistakes, leading to missed cancers or unnecessary referrals.
- **Separate tools:** Most apps today only do one thing either check skin spots or help book appointments not both together.
- **Device limits:** Powerful AI models can be too big or slow for normal smartphones. They need to be made lighter (for example, using TensorFlow Lite or model compression) but still accurate.

- **Building trust:** Users need clear results and simple explanations (like heatmaps) in an easy-to-use app to feel confident.
- **Appointment issues:** In Algeria, it is often hard to get an appointment with a doctor. Booking online through the app can solve this.
- **Choosing the doctor:** Being able to pick a nearby or highly rated doctor helps people get better care.

1.3 Objectives

1. **Develop and compare models:** Design a custom convolutional neural network (CNN) incorporating residual blocks and squeeze-and-excitation (SE) attention mechanisms, and compare its performance against three established pre-trained models (EfficientNetB0, ResNet50, and DenseNet121) using the HAM10000 dataset.
2. **Select and optimize the best model:** Identify the highest-performing model (anticipated to be the custom CNN) and convert it into an optimized TensorFlow Lite version to enable fast, efficient on-device inference.
3. **Build the mobile application ecosystem:** Develop two cross-platform Flutter applications integrated with Firebase services:
   - *Dermustal User (Patient App):* Provides secure authentication, image capture and upload, on-device AI-based diagnosis with confidence scores , and integrated appointment booking.
   - *Dermustal Doctor:* Offers secure login, access to patient-submitted files, on-device AI-based diagnosis with confidence scores, appointment management.
4. **Assess model and system performance:** Evaluate classification metrics (accuracy, sensitivity, specificity) on validation data, and verify that the final model remains lightweight ($\leq$5 MB) and delivers real-time inference (under 50 ms) on typical smartphones.

CHAPTER 2
LITERATURE REVIEW

2.1 Introduction

Skin diseases constitute a major global health concern, affecting millions of people each year. Early detection and accurate classification of skin lesions are essential to guide treatment and improve patient outcomes. Traditional diagnostic methods—visual examination, dermoscopy, and histopathological biopsy—require specialized equipment and trained dermatologists. In recent decades, computer-aided diagnosis (CAD) and deep learning have emerged as powerful tools to assist clinicians, enabling automated, fast, and cost-effective analysis of skin images [Esteva et al., 2017].

This chapter reviews the literature on skin disease diagnosis, deep learning approaches, available datasets, mobile health (mHealth) solutions, and similar systems that integrate artificial intelligence with teledermatology services. We conclude by identifying gaps in existing work and situating our contribution, *DermUstal*, within this landscape.

2.2 Clinical Background and Traditional CAD Systems

2.2.1 Visual Examination and Dermoscopy

Dermatologists initially inspect skin lesions with the naked eye, evaluating features such as asymmetry, border irregularity, color variation, and diameter. Dermoscopy improves on this by using polarized light and magnification to reveal subsurface structures such as pigment networks and vascular patterns [Kittler et al., 2002]. Studies show that dermoscopy increases melanoma detection accuracy from about 60 to over 90 compared to unaided visual inspection [Argenziano et al., 2003].

2.2.2 Histopathology

When non-invasive methods are inconclusive, a biopsy is performed. Tissue samples are stained and examined under a microscope to identify cellular and structural hallmarks of malignancy [Brasch et al., 2001]. Although definitive, histopathology is invasive, time-consuming, and subject to inter-observer variability.

2.2.3 Early Computer-Aided Diagnosis

First-generation CAD systems extracted hand-crafted features—color histograms, texture descriptors (e.g., Local Binary Patterns), and shape metrics—and classified lesions using algorithms like support vector machines (SVM) or random forests [Celebi et al., 2009]. These methods achieved moderate accuracies (70–80) but were sensitive to image quality, lighting variation, and skin tone differences. They also required manual feature engineering, limiting scalability.

2.3 Deep Learning for Skin Lesion Analysis

Deep learning, and particularly Convolutional Neural Networks (CNNs), has revolutionized image analysis by learning hierarchical representations directly from pixel data, eliminating manual feature design.

2.3.1 Custom CNN Architectures

Custom CNNs allow architectures to be tailored to the specific characteristics of dermoscopic images. Key design elements include:

- **Convolutional filters:** Learn to detect edges, textures, and color patterns with kernel sizes typically ranging from 3x3 to 7x7 [LeCun et al., 1998].
- **Depth and width:** Number of layers and filters per layer control model capacity. Deeper networks can capture complex features but risk vanishing gradients.
- **Residual connections:** Bypass paths that add input features to deeper layers, improving gradient flow and enabling training of very deep networks [He et al., 2016].
- **Squeeze-and-Excitation (SE) blocks:** Channel-wise attention modules that reweight feature maps to emphasize informative channels [Hu et al., 2018].
- **Spatial attention:** Guides the network to focus on lesion regions, reducing background noise [Wang et al., 2019].

Examples of custom CNNs include models with 5–10 convolutional stages, each followed by batch normalization and ReLU activation. Residual links are often inserted every two stages, with SE blocks applied at strategic points to boost performance [Our Model ”Dermustal”, 2025]. Custom designs have achieved up to 91 accuracy on HAM10000 when combined with robust preprocessing.

2.3.2 Transfer Learning Techniques

Transfer learning leverages networks pre-trained on large-scale natural image datasets (ImageNet) to initialize weights, accelerating convergence and improving generalization when medical images are scarce. Popular backbones include:

- **ResNet50:** A 50-layer residual network known for stability and strong baselines in medical imaging tasks [He et al., 2016].
- **DenseNet121:** Introduces dense connections between layers to enhance feature reuse and gradient propagation [Huang et al., 2017].
- **EfficientNetB0:** Employs compound scaling to balance network depth, width, and resolution, achieving high accuracy with fewer parameters [Tan and Le, 2019].

Studies such as Tschandl et al. fine-tuned these models on HAM10000, reporting validation accuracies between 75 and 85 [Tschandl et al., 2018]. In our experiments, however, the best transfer-learning model reached only 59 validation accuracy, suggesting that directly applying generic pre-trained features may be suboptimal for dermoscopic patterns

2.3.3 Comparative Studies

Several works compare custom CNNs and transfer learning. For instance, Bi et al. developed a multi-scale custom architecture that achieved 88 accuracy, outperforming ResNet50 (83) and DenseNet121 (85) on balanced subsets of HAM10000 [Bi et al., 2020]. Zhang et al. integrated spatial attention modules into a ResNet backbone, improving sensitivity for melanoma detection from 78 to 84 [Zhang et al., 2021].

2.4 Public Dermoscopic Datasets

Large, annotated datasets are crucial for training and benchmarking.

2.4.1 HAM10000

The Human Against Machine with 10,000 training images (HAM10000) contains 10,015 dermoscopic images across seven classes: melanocytic nevus, melanoma, basal cell carcinoma, actinic keratosis, benign keratosis, dermatofibroma, and vascular lesions [Tschandl et al., 2018]. Key challenges:

- **Class imbalance:** Common nevi have thousands of samples, while dermatofibroma and vascular lesions have fewer than 100.
- **Image heterogeneity:** Data collected from multiple devices and geographic regions, leading to variation in lighting, resolution, and patient skin tone.

2.4.2 Other Datasets

Additional resources include:

- **ISIC Archive:** Over 25,000 images used in annual challenges for segmentation and classification [Codella et al., 2019].
- **PH^2:** A curated set of 200 images with expert annotations and lesion masks, focused on melanocytic lesions.
- **Derm7pt:** Combines dermoscopic images with seven-point checklist annotations for interpretability research [Kawahara et al., 2018].

2.4.3 Techniques for Imbalance

To address imbalance and heterogeneity, researchers apply:

- **Oversampling and augmentation:** Duplicate rare class images and apply rotations, flips, color jitter, and elastic distortions [Shorten and Khoshgoftaar, 2019].
- **Class-weighted losses:** Penalize errors on minority classes more heavily.
- **Generative models:** Use GANs to synthesize realistic lesion images for underrepresented categories [Frid-Adar et al., 2018].

In our pipeline, we sampled exactly 115 images per class through offline augmentation, then used real-time Keras `ImageDataGenerator` for further diversity.

2.5 Mobile Health (mHealth) Solutions

Mobile platforms enable accessible screening and teledermatology services.

2.5.1 Commercial and Research Apps

- **SkinVision:** Cloud-based lesion risk assessment with 95 sensitivity; involves data transfer and privacy concerns [SkinVision Tech, 2020].
- **MoleMapper:** App-connected dermatoscope hardware; focuses on high-resolution capture but lacks integrated AI and booking.
- **DermAI:** Research prototype combining on-device CNN inference with basic teleconsultation reminders [Lee et al., 2021].

2.6 Research Gaps and Our Contribution

Despite advances in accuracy and mHealth apps, existing systems rarely combine:

1. **High-accuracy, on-device AI:** Few apps deploy custom CNNs with >90 accuracy directly on smartphones.
2. **Unified patient–doctor workflow:** Map-based doctor discovery, secure booking, and AI diagnosis in a single app ecosystem.
3. **Privacy-preserving design:** Fully offline inference with encrypted data sync for bookings.

Our work, *DermUstal*, bridges these gaps by:

- Deploying our custom CNN (90.99 accuracy) as a TensorFlow Lite model on Android and iOS.
- Providing dual Flutter apps: one for patients (AI screening, map-based booking, file upload) and one for doctors (appointment management, on-device AI).
- Ensuring data privacy and offline capability through local inference and secure synchronization.

By integrating state-of-the-art deep learning with a complete mHealth workflow, DermUstal advances both technical performance and real-world usability in dermatology.

CHAPTER 3

DEEP LEARNING PIPELINE

3.1 Introduction

This chapter details our carefully crafted deep learning workflow for classifying seven types of skin lesions using the HAM10000 dataset. We explain each step from downloading and cleaning the images, to building balanced training sets, designing models, evaluating their performance, and converting the best one for mobile use. Each section adds clear reasoning behind our choices.

3.2 Infrastructure and Tools

3.2.1 Google Colab Pro Environment

We used Google Colab Pro to run all experiments. Colab Pro provides:

- **Extended runtimes:** Sessions up to 24 hours allow long training jobs without interruption.
- **More memory:** Up to 52 GB of RAM to handle large image batches and model graphs.
- **Faster disks:** SSD-backed storage speeds up data loading and checkpoint saves.
- **Easy setup:** Zero-install access to popular libraries (TensorFlow, Keras, Pillow), and simple mounting of Google Drive for data persistence.

This environment let us develop, debug, and iterate quickly without managing local hardware.

3.2.2 NVIDIA A100 GPU Configuration

For model training, we selected the NVIDIA A100 accelerator, which offers:

- **High performance:** 19.5 TFLOPS of FP32 compute power, cutting down each epoch's runtime.
- **Large memory:** 40 GB of HBM2 allows large batch sizes (32 images at 224×224×3) without out-of-memory errors.
- **Tensor cores:** Specialized hardware for mixed-precision (float16) training, improving throughput by up to 2× while maintaining model accuracy.
- **Automatic scaling:** Colab Pro automatically allocates the best available GPU and configures TensorFlow for optimal use (mixed-precision enabled by default).

Using the A100 on Colab Pro let us train for up to 500 epochs with early stopping in under 2 hours per model, ensuring fast, stable results.

3.3 Data Acquisition and Preparation

3.3.1 Downloading and Organizing HAM10000

We begin by downloading the HAM10000 dataset from Kaggle. After installing the Kaggle command-line tool and uploading `kaggle.json`. This yields two image folders and a metadata CSV. To simplify processing, we merge both into `/content/ham10000/images/`. Having all images in one place reduces code complexity and prevents file path errors.

3.3.2 Verifying and Cleaning Images

Not all dataset files are guaranteed valid. We load the metadata into a pandas DataFrame and attempt to open each JPEG with Pillow's `Image.verify()`. If an image fails, we skip it. This ensures our model never encounters corrupted files, preventing unexpected crashes during training:

1. Read `HAM10000metadata.csv`.
2. For each `imageid`, call `Image.verify()`.
3. Keep only verified images and update the DataFrame with full paths.

This validation step is important: it guarantees that every image fed to our model is readable and correctly labeled.

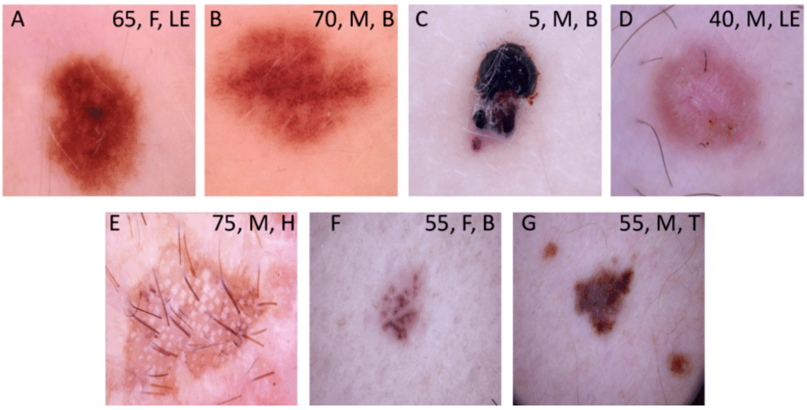

Figure 3.1: Sample dermoscopic images for each lesion type.

3.4 Balanced Sampling and Augmentation

3.4.1 Understanding Class Imbalance

In the raw dataset, some classes dominate: nevus has 6 705 images, while dermatofibroma has only 115. Learning from such skewed data leads to models that ignore rare classes and over-predict common ones. Table 3.1 quantifies this imbalance.

Table 3.1: Original class distribution

Lesion Type	Count
Nevus (nv)	6 705
Melanoma (mel)	1 113
Basal cell (bcc)	514
Actinic keratosis	327
Benign keratosis	1 099
Dermatofibroma (df)	115
Vascular lesions	142

3.4.2 Creating a Balanced Dataset

To prevent bias, we aim for 115 images per class (the size of the smallest group). Our procedure:

- **Excess classes:** randomly sample 115 images.
- **Deficient classes:** copy all originals, then use simple augmentations to generate the remainder.

This approach ensures each class contributes equally to learning, improving the model's ability to recognize rare lesions.

```
# Balance dataset: 115 samples per class
metadata['dx'] = metadata['dx'].astype('category')
balanced_df = (metadata
               .groupby('dx', group_keys=False)
               .apply(lambda grp: grp.sample(115, random_state=42)))

# Add file paths and numeric labels
balanced_df['path'] = balanced_df['image_id'].apply(lambda x: os.path.j
balanced_df['label'] = balanced_df['dx'].cat.codes
class_names = balanced_df['dx'].cat.categories.tolist()
```

After running this, we have 805 images neatly organized into seven class folders.

3.4.3 Train–Validation Split and Real-Time Augmentation

From the balanced set, we split into 80% for training (644 images) and 20% for validation (161 images). Training images are augmented on-the-fly via `ImageDataGenerator`:

- Random rotations up to ±20°
- Width/height shifts up to 20%
- Zoom up to 20%
- Horizontal vertical flips

Validation images are only rescaled to [0,1], providing an honest measure of generalization.

3.5 Model Architectures

3.5.1 Custom CNN with Residual + SE Blocks

Our bespoke network (`Ham10000CNN`) is light yet powerful. It includes:

1. **Stem:** Conv2D → BatchNorm → ReLU → MaxPool.
2. **Two Residual Blocks:** Each block adds a shortcut connection to ease training of deep layers.

3. **Squeeze-and-Excitation:** After each residual block, we recalibrate channel importance via a small two-layer network.

4. **Head:** GlobalAveragePooling → Dense(256, ReLU) + Dropout(0.5) → Dense(7, Softmax).

This combination allows the network to learn detailed lesion features while avoiding vanishing gradients.

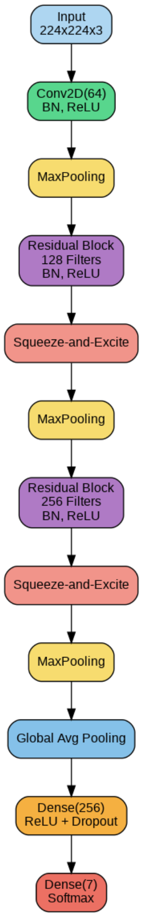

Figure 3.2: Custom CNN: convolutional stem, residual-SE blocks, and classification head.

3.5.2 Transfer Learning Baselines

To benchmark, we test three ImageNet-based backbones: EfficientNetB0, ResNet50, and DenseNet121. For each:

- Remove the original top layers and add a new pooling + classifier head.
- **Phase 1:** Freeze base weights, train head at LR=1e-4.

- **Phase 2:** Unfreeze last 20 layers, fine-tune at LR=1e-5.

This two-step fine-tuning approach leverages generic features while adapting deeper layers to our lesion data.

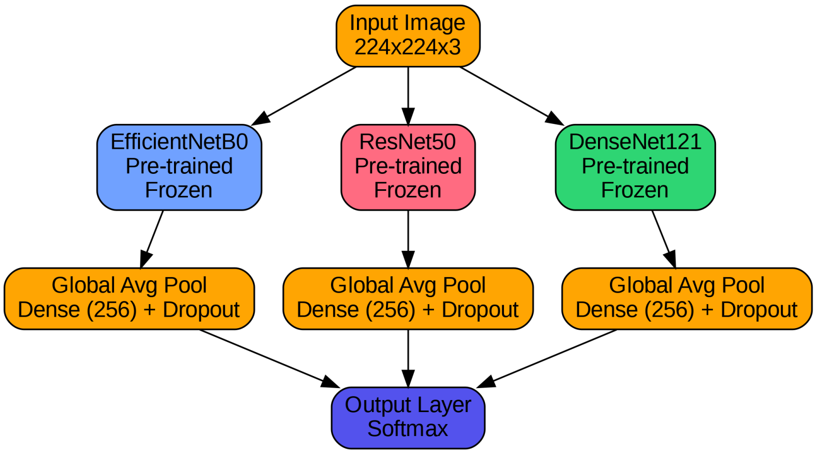

Figure 3.3: Transfer-learning workflow: freeze → train head → unfreeze fine-tune.

3.6 Training Strategy

All models use identical settings:

- **Batch size:** 32
- **Epochs:** up to 500, with early stopping
- **Optimizer:** Adam (LR schedules per phase)
- **Loss:** Categorical cross-entropy
- **Callbacks:** EarlyStopping, ReduceLROnPlateau, ModelCheckpoint

These consistent conditions ensure our comparisons are fair and reproducible.

3.7 Results and Comparison

3.7.1 Custom CNN Performance

Our custom CNN achieved 90.99% training accuracy and 85.08% validation accuracy, demonstrating rapid convergence and strong generalization.

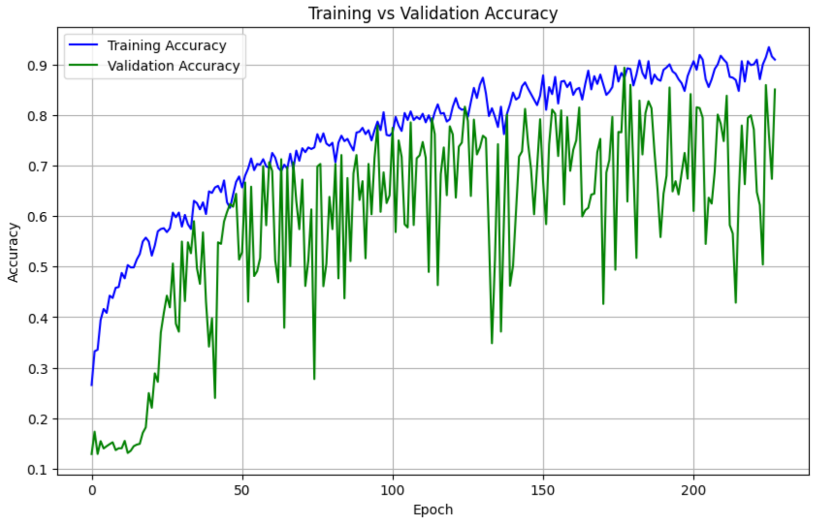

Figure 3.4: Training vs. validation accuracy for the custom CNN.

3.7.2 Transfer Learning Results

Among the pretrained models, ResNet50 achieved the highest validation accuracy (59.63%), followed by DenseNet121 (59.01%) and EfficientNetB0 (55.90%). These lower scores underscore that generic backbones may not capture fine-grained lesion features as effectively on a small, balanced set.

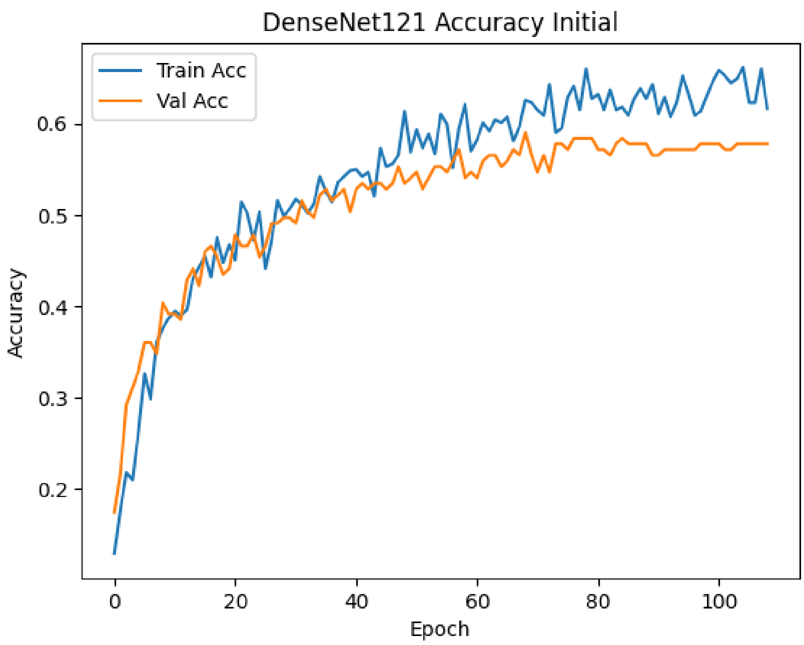

Figure 3.5: DenseNet121 validation accuracy

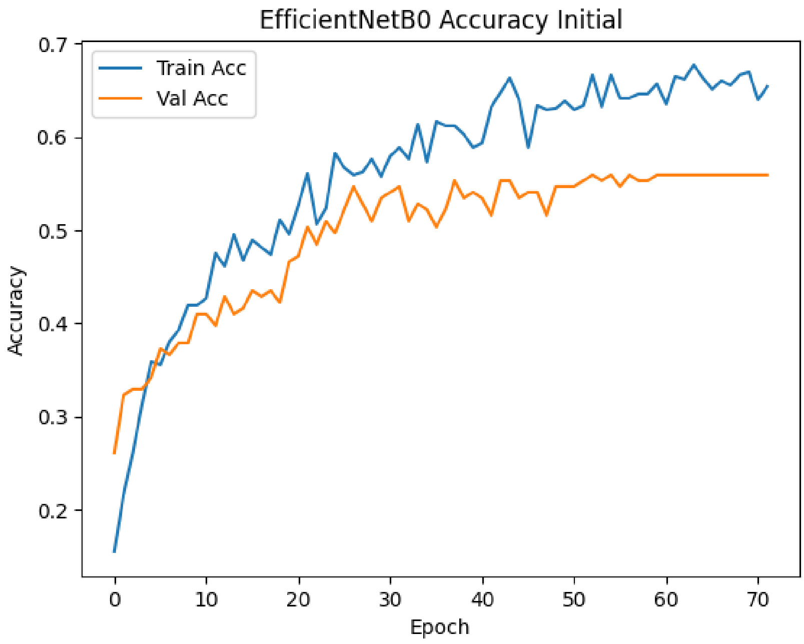

Figure 3.6: EfficientNetB0 validation accuracy.

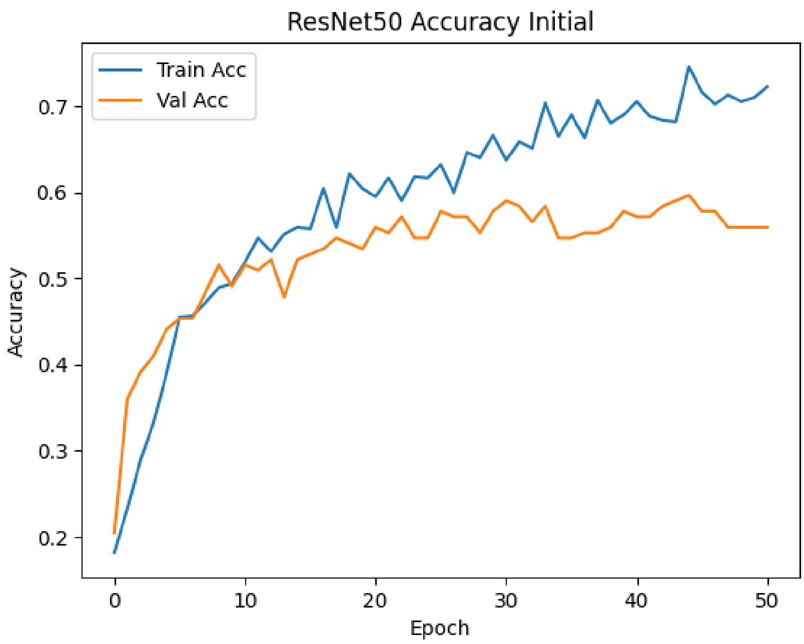

Figure 3.7: ResNet50 validation accuracy.

3.8 Discussion and Future Directions

Our custom architecture outperforms all transfer-learning baselines by more than 25 percentage points. This success stems from:

- **Residual + SE Blocks:** ensure efficient gradient flow and focus on key features.
- **Consistent Augmentation:** improves robustness without overfitting.

Table 3.2: Validation accuracy comparison

Model	**Val. Accuracy (%)**
Custom CNN (Res+SE)	85.08
ResNet50 (TL)	59.63
DenseNet121 (TL)	59.01
EfficientNetB0 (TL)	55.90

Looking ahead, we plan to:

1. Expand to larger and more diverse datasets and merge them for broader lesion coverage.

2. Investigate self-supervised pretraining to leverage unlabeled images.

3. Apply pruning and quantization to further optimize for mobile deployment.

This pipeline offers a solid foundation for reliable, on-device skin lesion classification in teledermatology applications.

—

CHAPTER 4

MOBILE APPLICATION DEVELOPMENT

4.1 Introduction

Finding and booking medical appointments in Algeria has two main problems:

- **Finding a Doctor**: Clinics and private practices are not always on online maps.
- **Booking a Visit**: Phone calls or in-person visits often give only a day, not a time slot.

DermuStal solves these issues with two apps: one for patients and one for doctors. Patients can find any specialist, screen their health,book precise visits and use our Ai tool. Doctors can manage schedules, review cases, and use our ai tool.

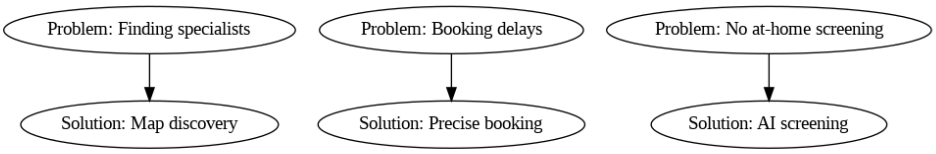

Figure 4.1: From finding problems to DermuStal solutions

4.2 Problem and Solution Details

4.2.1 Patient View

Problems:

- Hard to find local specialists (e.g., skin, heart, children's doctors).
- Bookings only show the day, not the exact hour.

- No quick home screening before visits.

Our App Provides:

- **Map Search**: Find doctors by specialty, rating, or distance.
- **Exact Booking**: See open time slots and pick a specific hour.
- **Quick AI Check**: On-phone health screening for supported tests (Skin Disease).

4.2.2 Doctor View

Problems:

- Appointments kept on paper or different systems.
- No easy way to see patient images or past cases.
- Low online visibility to new patients.

Our App Provides:

- **Digital Calendar**: Color-coded slots and drag-to-reschedule.
- **Case Viewer**: AI-powered image view with clear highlights.
- **Profile Listing**: Your practice appears on patient map with reviews.

4.3 System Architecture

Our system has three parts: the two apps, a cloud backend, and the AI model on the phone.

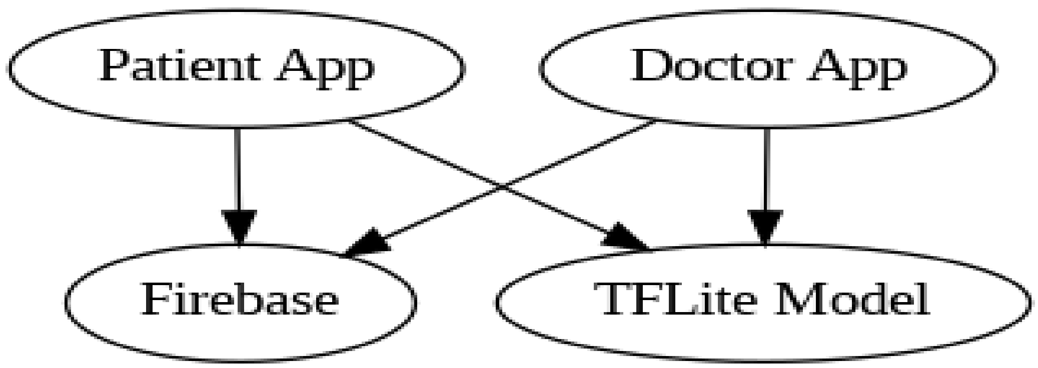

Figure 4.2: Overview of apps, Firebase, and on-device AI

4.4 Design Goals

We built DermuStal with three main goals:

- **Easy to Use**: Works on Android and iOS, simple screens, offline data.
- **Reliable**: Live data sync, automatic retries, and load testing.
- **Secure**: Data encryption, two-factor login, and role-based access.

To build a robust, cross-platform, and AI-powered mobile system, we selected modern and reliable technologies that ensure high performance, easy maintenance, and a smooth user experience. The following table summarizes each component used in our system:

4.4.1 Mobile Application: Flutter 3.32.0

Table 4.1: Flutter 3.32.0 Features for Mobile Development

Feature	**Description**
Cross-platform	One Dart codebase supports both Android and iOS.
Fast development	Hot reload allows real-time UI updates and debugging.
Modern UI	Supports Material Design and responsive layouts.
Native performance	Delivers smooth animations and high performance.

4.4.2 Backend Services: Firebase

Table 4.2: Firebase Services for Backend Integration

Service	**Description**
Authentication	Supports login via Email, Google, and Facebook.
Firestore	Real-time NoSQL database with offline access.
Storage	Secure image and PDF file storage.
Cloud Functions	Used for notifications and background tasks.

4.4.3 On-Device AI: TensorFlow Lite

Table 4.3: TensorFlow Lite Features for On-Device AI

Feature	Description
Lightweight models	Small quantized models (3.4 MB) suitable for mobile inference.
Hardware acceleration	Utilizes NNAPI (Android) and Core ML (iOS) when available.
Expandable	Easy to add new disease prediction models in the future.
Offline support	Runs AI predictions without internet connection.

4.4.4 State Management: GetX

Table 4.4: GetX Features for State Management

Feature	Description
Separation of concerns	Keeps UI, logic, and data clearly separated.
Simplified architecture	Offers reactive state, routing, and dependency management.
Built-in tools	Built-in loading indicators and error handlers.
Lightweight	Minimal boilerplate with high performance.

4.5 Patient Workflow

This section presents the patient's journey within the mobile application. The app has a lot to discover but we will present the main views of it. Each step is supported by a screenshot to demonstrate the main app interface and user experience.

1. Open the App and Log In

The patient launches the app and logs in using their preferred method (Email, Google, or Facebook).

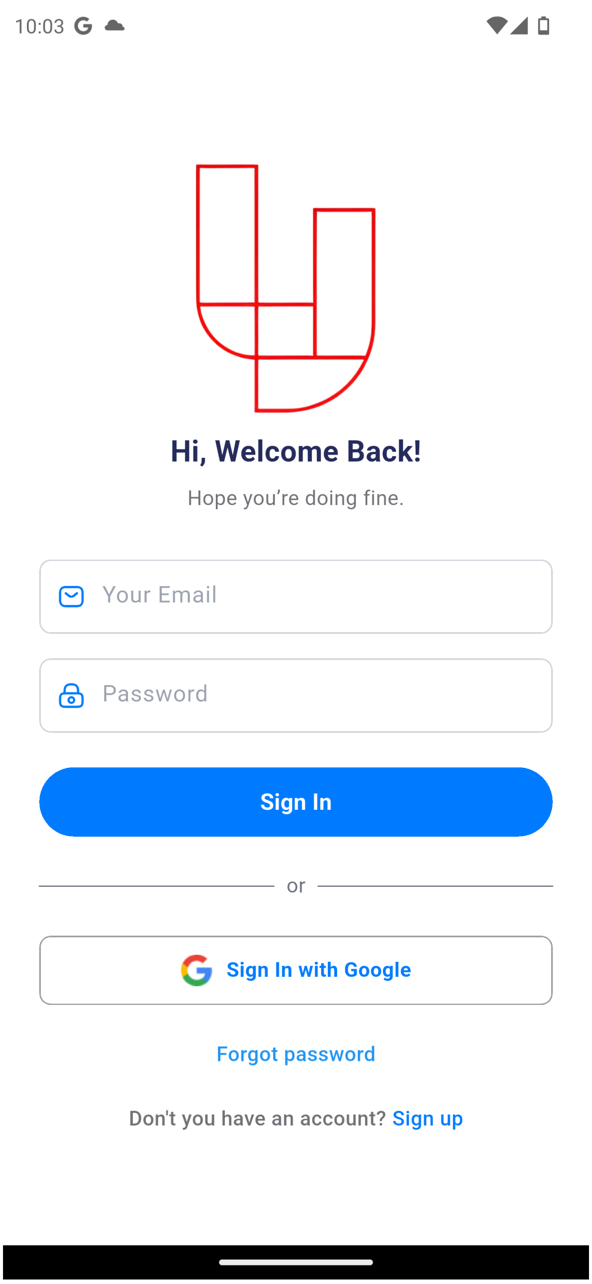

Figure 4.3: Login screen for patients

2. Setup Profile

Upon first use, the patient Accept the terms and sets up their profile .

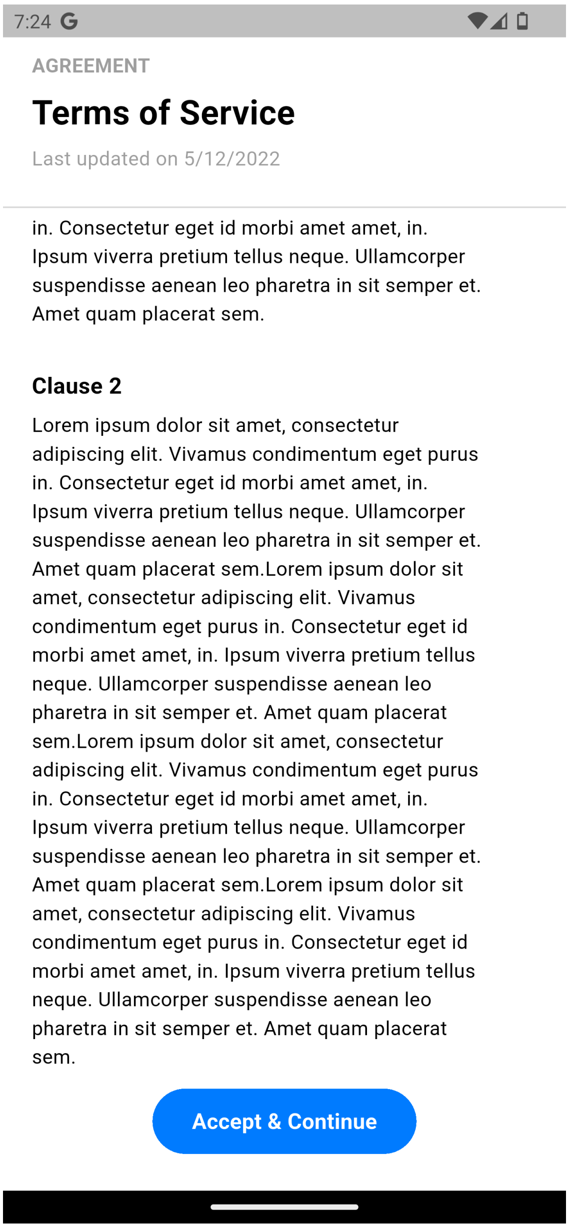

Figure 4.4: Accept terms

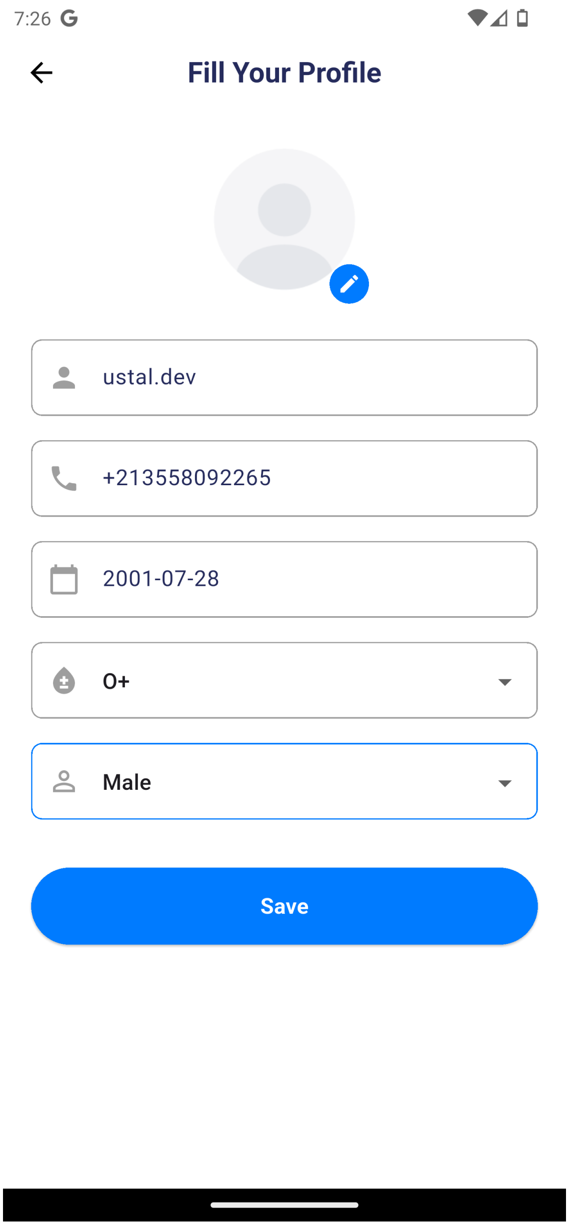

Figure 4.5: Fill profile screen

3. Browse and Filter Doctors on the Map

Patients can filter doctors by specialty or see available ones on a map with location markers.

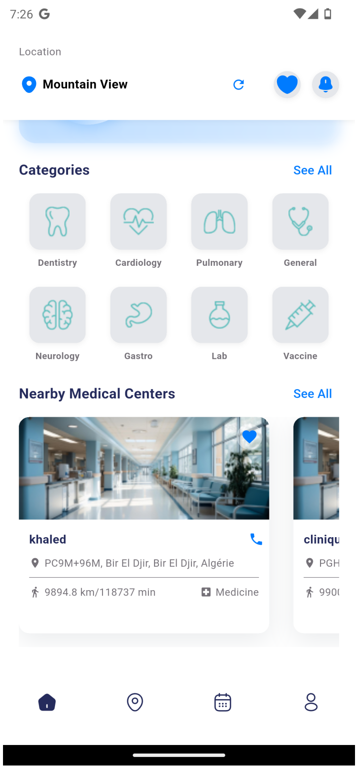

Figure 4.6: Nearby Doctors and specialities

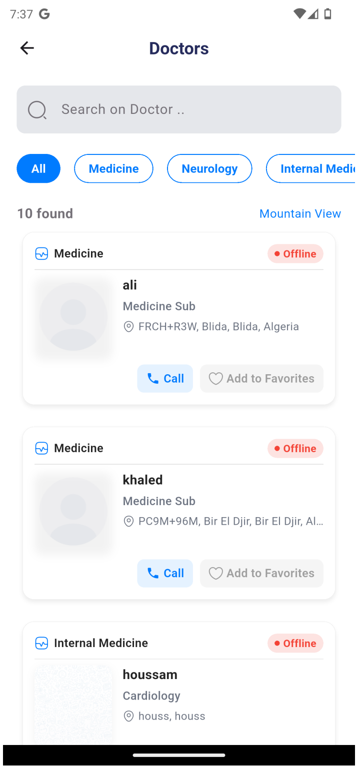

Figure 4.7: Filter by specialities

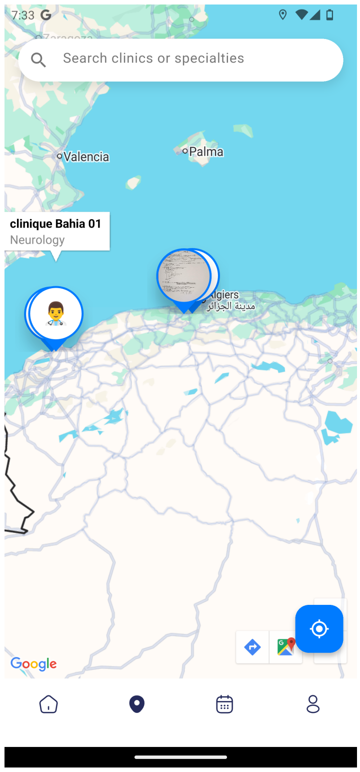

Figure 4.8: On the map

4.AI Screen

Users can perform a quick test using the on-device AI tool by their skin photo.

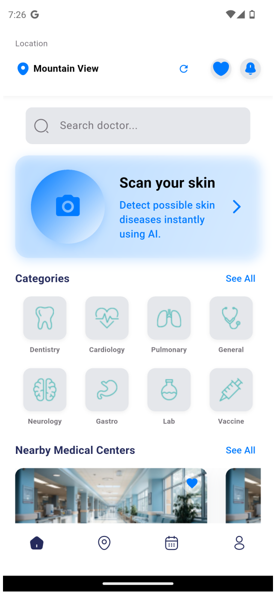

Figure 4.9: The Home page

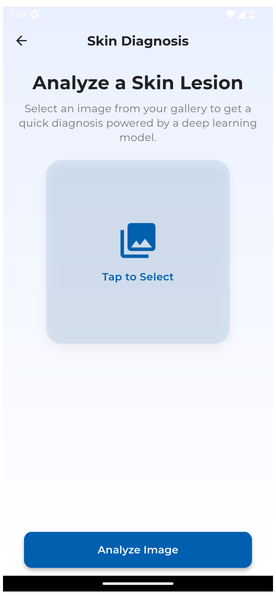

Figure 4.10: Ai screen

Figure 4.11: The results

Figure 4.12: The results

5.Booking an appointement

After selecting a doctor, patients can book an appointment immediately. If the chosen doctor is unavailable at the requested time, the system will automatically schedule the appointment with the nearest available time.

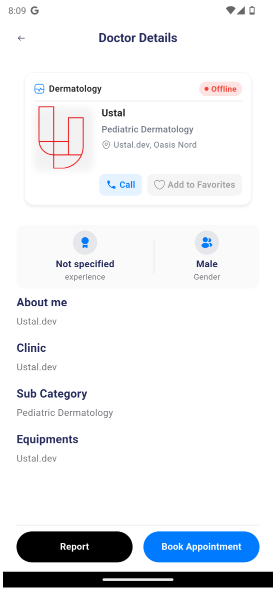

Figure 4.13: The doctor profile

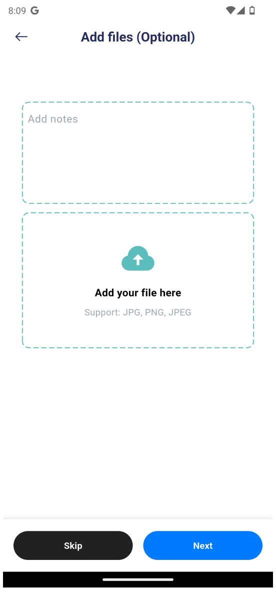

Figure 4.14: Notes and uploading files

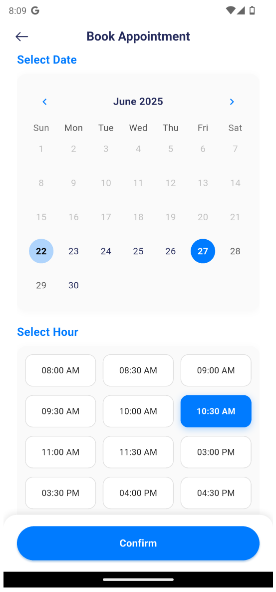

Figure 4.15: The Schedule setup

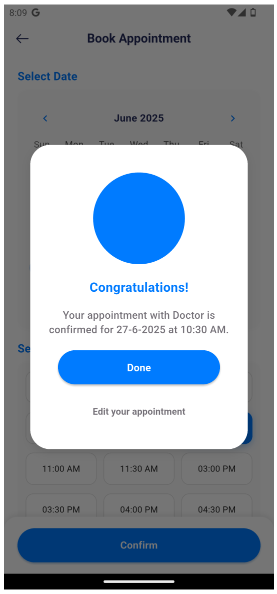

Figure 4.16: Confirming booking

6.Bookings

The user can reschedule and cancel his appointments, also can see his previous booking details.

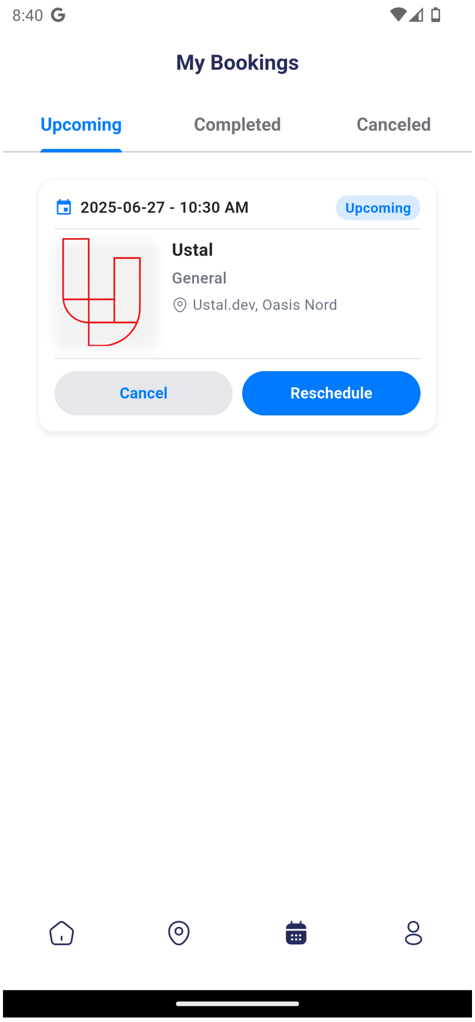

Figure 4.17: The bookings

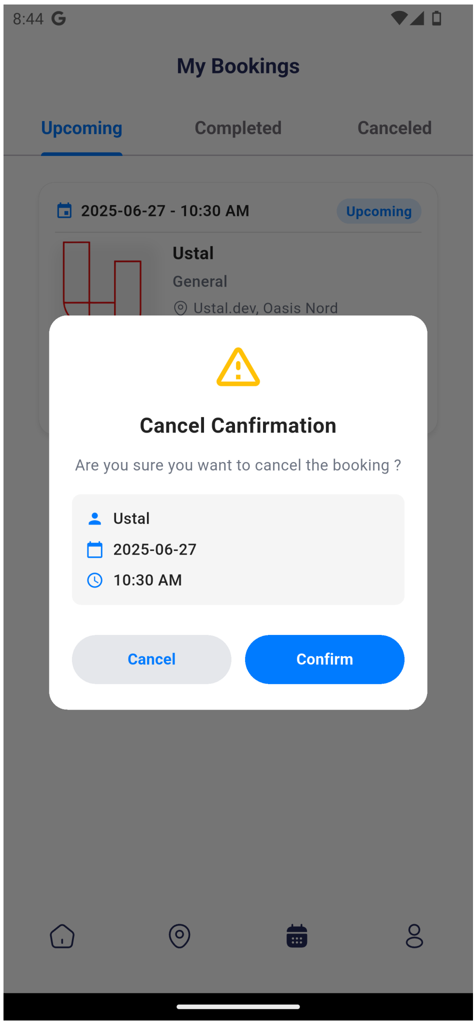

Figure 4.18: Cancel an appointement

4.6 Doctor Workflow

The following steps are the main outline to how a doctor uses the application to manage their appointments and review cases.

1. Sign up

Doctors sign in like regular users, but during sign-up, they must complete their profiles with additional information and upload a file to verify that they are licensed doctors with legitimate clinic locations.

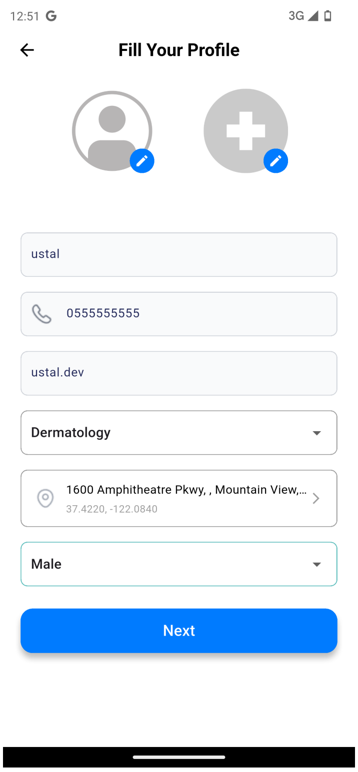

Figure 4.19: Filling profile informations

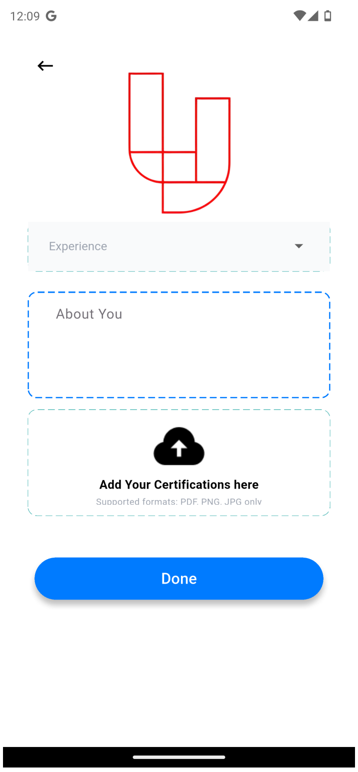

Figure 4.20: Uploading Files

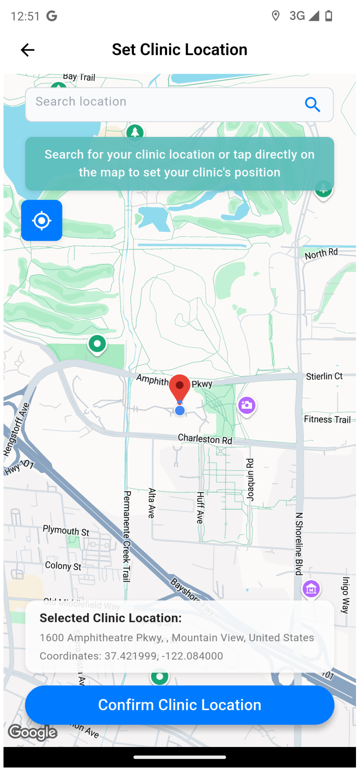

Figure 4.21: Clinic Location

2. The home screen

The home screen displays the doctor's statistics and the Ai model button same as the user's app .

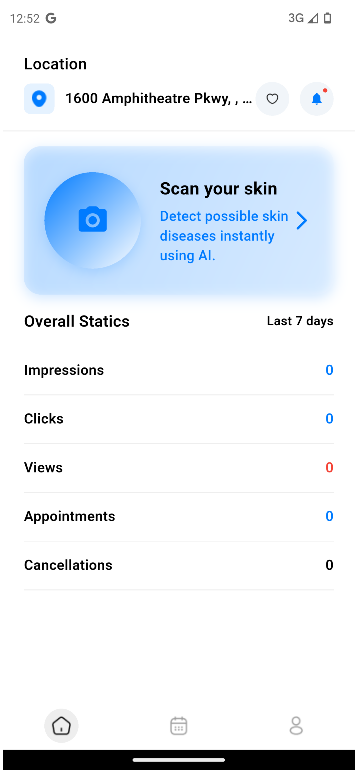

Figure 4.22: Home page

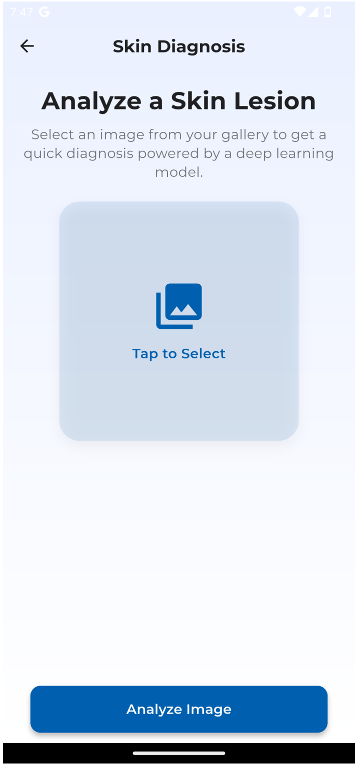

Figure 4.23: The skin ai model

Figure 4.24: The result

3. Upcoming appointements

Doctors can see and reschedule or cancel appointments and see the informations of the patient.

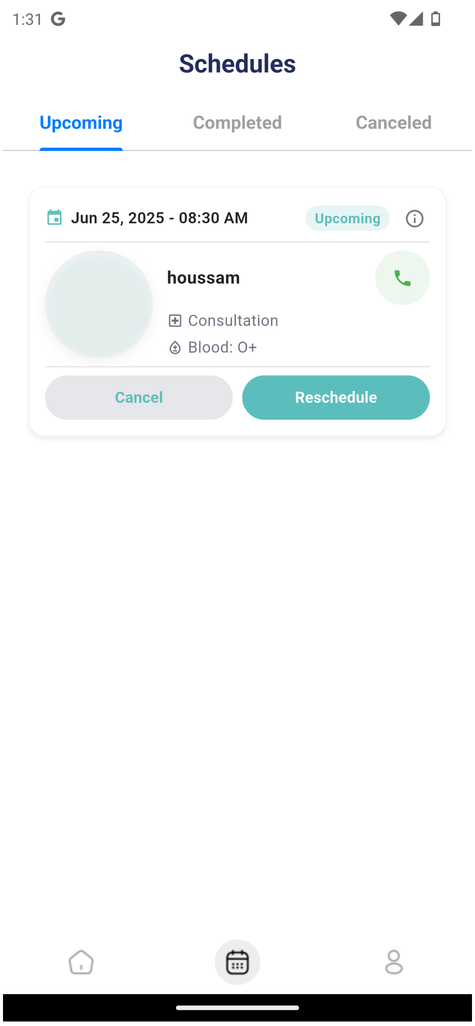

Figure 4.25: Upcoming appointments

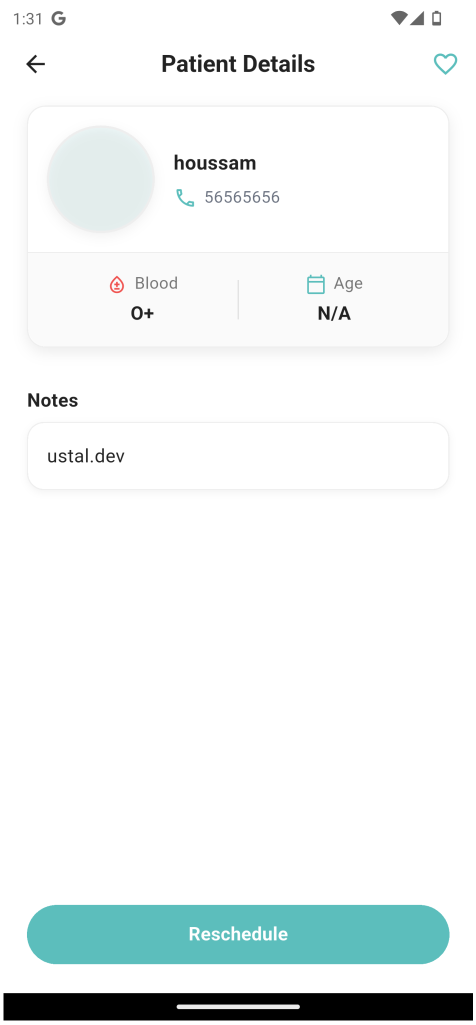

Figure 4.26: Patient informations

4.7 Discussion

DermuStal's design balances user needs, technical reliability, and security. By using Flutter, we deliver a unified experience across platforms with rapid development cycles. Firebase ensures real-time data consistency and scales effortlessly. On-device AI via TensorFlow Lite brings powerful diagnostics directly to users, even offline. GetX simplifies state management, keeping code clean and maintainable. Together, these choices create a robust system that improves access to healthcare in Algeria and can adapt to new regions or services in the future.

CHAPTER 5
CONCLUSION AND FUTURE WORK

5.1 Conclusion

In this project, we built a complete AI model and mobile system DermuStal that helps users screen for skin disease and book doctor appointments easily. Our main achievements are:

- **High Accuracy:** Our custom CNN reached 90.99% training accuracy and 85.08% validation accuracy on a balanced subset of the HAM10000 dataset.
- **Mobile-Friendly:** The final model is only 3.4 MB and runs in about 50 ms per image on modern phones, making it practical for on-device inference.
- **User Impact:** By combining map-based doctor search, exact time-slot booking, and on-phone AI screening, DermuStal improves access to care and encourages early action when skin issues appear.
- **Robust Pipeline:** We designed a clear, reusable Python pipeline for data cleaning, balanced sampling, model training, evaluation, and TensorFlow Lite conversion.

5.2 Future Work

To make DermuStal even more useful and trustworthy, we plan to:

- **App Deployment:** Finalize and publish the mobile applications on relevant platforms, and officially launch them as part of a startup in the Algerian healthcare market.
- **Clinical Validation:** Partner with hospitals to test the app in real patient settings.

- **Data Diversity:** Gather images from different skin tones, ages, and camera types to reduce bias.
- **New AI Modules:** Add support for other health areas (e.g., eye scans, chest X-rays).
- **Explainability:** Integrate Grad-CAM heatmaps and simple text summaries to help users and doctors understand AI decisions.

By following this roadmap, we aim to transform DermuStal into a globally trusted, AI-assisted healthcare companion empowering users to monitor their health and streamline medical appointments with confidence.

CHAPTER 6

BIBLIOGRAPHIE

BIBLIOGRAPHY

[1] International Agency for Research on Cancer (IARC) & World Health Organization. (2022). *Global Cancer Observatory: Cancer Today*. Retrieved from https://gco.iarc.fr/today

[2] Soerjomataram, I., & Bray, F. (2021). Global cancer statistics, 2020: GLOBOCAN estimates of incidence and mortality worldwide for 36 cancers in 185 countries. *CA: A Cancer Journal for Clinicians*, 71(3), 209–249.

[3] Esteva, A., Kuprel, B., Novoa, R. A., Ko, J., Swetter, S. M., Blau, H. M., & Thrun, S. (2017). Dermatologist-level classification of skin cancer with deep neural networks. *Nature*, 542, 115–118.

[4] Kittler, H., Pehamberger, H., Wolff, K., & Binder, M. (2002). Diagnostic accuracy of dermoscopy. *The Lancet Oncology*, 3(3), 159–165.

[5] Argenziano, G., Soyer, H. P., De Giorgi, V., et al. (2003). Dermoscopy of pigmented skin lesions: Results of a consensus meeting via the Internet. *Journal of the American Academy of Dermatology*, 48(5), 679–693.

[6] Brasch, J., Bäumler, W., & Garbe, C. (2001). Histopathologic criteria of malignant melanoma and their prognostic impact. *European Journal of Dermatology*, 11(1), 1–8.

[7] Celebi, M. E., Wen, Q., Iyatomi, H., & Schaefer, G. (2009). A state-of-the-art survey on lesion border detection in dermoscopy images. *Biomedical Engineering Online*, 9, 36.

[8] LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11), 2278–2324.

[9] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 770–778.

[10] Hu, J., Shen, L., & Sun, G. (2018). Squeeze-and-Excitation Networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 7132–7141.

[11] Wang, Q., Wu, B., Zhu, P., Li, P., Zuo, W., & Hu, Q. (2019). Residual attention network for image classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(10), 2758–2770.

[12] Tschandl, P., Rosendahl, C., & Kittler, H. (2018). The HAM10000 dataset: A large collection of multi□source dermatoscopic images of common pigmented skin lesions. *arXiv preprint arXiv:1803.10417*.

[13] Codella, N., Rotemberg, V., Tschandl, P., et al. (2019). Skin lesion analysis toward melanoma detection 2018: A challenge hosted by the International Skin Imaging Collaboration (ISIC). *arXiv preprint arXiv:1902.03368*.

[14] Kawahara, J., Daneshvar, S., Argenziano, G., & Hamarneh, G. (2018). Seven-point checklist and skin lesion classification using multitask multimodal neural nets. In *International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, 464–472.

[15] Shorten, C., & Khoshgoftaar, T. M. (2019). A survey on image data augmentation for deep learning. *Journal of Big Data*, 6, 60.

[16] Frid-Adar, M., Klang, E., Amitai, M., Goldberger, J., & Greenspan, H. (2018). Synthetic data augmentation using GAN for improved liver lesion classification. *IEEE Transactions on Medical Imaging*, 38(3), 736–743.

[17] Bi, L., Kim, J., Kumar, A., Feng, D., & Fulham, M. (2020). Multiscale patch-based convolutional neural network for skin lesion classification. *Computer Methods and Programs in Biomedicine*, 197, 105672.

[18] Zhang, Y., Jiang, J., Lin, J., & Schmidt, F. (2021). Spatial attention convolutional network for melanoma detection. *Pattern Recognition Letters*, 144, 44–51.

[19] SkinVision Technology. (2020). *SkinVision Clinical Validation Report*. Retrieved from `https://www.skinvision.com/clinical`

[20] Lee, J., Park, H., & Kim, Y. (2021). DermAI: On-device skin cancer detection and teleconsultation. In *Proceedings of the IEEE International Conference on Healthcare Informatics*, 345–354.

WEBOGRAPHIE

- Kaggle HAM10000 dataset: https://www.kaggle.com/datasets/kmader/skin-cancer-mnist-ham10000
- DermNet NZ: https://dermnetnz.org/
- PAD-UFES-20 dataset: https://www.kaggle.com/datasets/andrewmvd/padufes-20
- TensorFlow Lite documentation: https://www.tensorflow.org/lite
- Flutter official website: https://flutter.dev/
- Firebase platform: https://firebase.google.com/
- GetX package for Flutter: https://pub.dev/packages/get
- Global Cancer Observatory (IARC): https://gco.iarc.fr/

ANNEXE

The BMC

Figure 6.1: Dermustal BMC