AI system for melanoma recognition

1st Jakub Grelowski
Wrocław University of Science and Technology
Wroclaw, Poland
262754@student.pwr.edu.pl

2nd Kamil Kochan
Wrocław University of Science and Technology
Wroclaw, Poland
259141@student.pwr.edu.pl

3rd Maksym Malicki
Wrocław University of Science and Technology
Wroclaw, Poland
259216@student.pwr.edu.pl

***Index Terms*—melanoma, artificial intelligence, computer vision, machine learning**

I. Description of the designed system

The main goal of this project was a RESTful web API that uses previously developed machine learning models. The API should allow users to classify melanoma from benign based on the uploaded image and return an appropriately formatted response.

In the final development phase of the project, a mobile application was developed to interact with the API, allowing users to upload images and receive classification results directly through their phones. This application serves as a front-end interface with the melanoma detection API back-end. To extend the functionality of our application, we implemented a feature that allows users to upload and store pictures to monitor the development of their birthmarks. To achieve that, we used a cloud file storage container and an external identity provider.

A. Tools

React Native will be used to develop a mobile application with which to take and upload photos. Communication between the client application and the model is done via dedicated API implemented with help of the Python Fast API framework.

II. Short review of used literature

The literature review performed in the first part covered general issues related to technology in medicine, the use of artificial intelligence in diagnosis, and the process of diagnosis itself and the benefits of making a diagnosis in advance. Acquiring knowledge in the medical field allowed us to focus more on technical matters, such as image classification methods and data augmentation techniques. In order to better understand the problem, we also reviewed already existing systems for automatic melanoma recognition.

III. Conclusions

In a previous phase of the project, research has shown that Gradient Boost model demonstrates good performance for simple image classification tasks, however, CNNs outperform it in more demanding contexts where high noise robustness is required. Also, data augmentation techniques such as rotations, scaling and color jittering improved performance of both models.

These aspects led us to choose CNNs to work with the web application, as this is particularly important when dealing with images taken, with the usual cameras available on phones.

IV. The Purpose of the Project

A. The purpose

The purpose of this project is to develop a mobile app for early melanoma detection. The application allows users to perform preliminary melanoma screenings by taking photos of their skin using their mobile device's camera. The application utilises machine learning to analyse images for potential signs of melanoma, providing users with accessible, early detection insights. If the system detects any moles defects, it encourages the user to get the medical follow-up. The project includes all phases of application development, including the design phase, implementation of data layer (REST API and machine learning algorithms for melanoma detection), visual layer (mobile application) and deployment.
The projects main objectives are:

1) **Increase Early Detection Rates (must have):**
2) **Enhance Accessibility to Preventive Screening (must have):**
3) **Increase Public Health Awareness (could have):**
4) **Apply machine learning algorithms in medical diagnostics (must have):**
5) **Monitor skin changes over time (should have):**

B. The stakeholders

The project outlines various types of stakeholders:

1) **End Users:** People concerned about their health, wanting to monitor their skin condition over time

2) **Healthcare Providers:** Dermatologists, who's diagnostics might benefit from the data present in the application
3) **Medical Researchers:** Institutions or research teams, who can utilize anonymized data to improve their solutions or patents.

Among the stakeholders the following groups were distinguished:

- **Client:**
  - A healthcare organization
  - A research institution
  - A government institution
  - Companies investing in app development
- **Customer:**
  - People who download and use the app for monitoring their skin health, including those at higher risk for melanoma.
  - Caretakers of people with potential skin defects (seniors, children)
- **Other Interested Parties:**
  - Public health agencies - necessary for potential advertising and popularisation of the product, knowledge level: high, impact on the project: low
  - Skin cancer awareness organizations - necessary for potential advertising and popularisation of the product, knowledge level: high, impact on the project: low
  - Technology experts - necessary for technical consultations regarding project design and implementation, knowledge level: high, impact on the project: high
  - Experts in the field of melanoma - necessary for technical consultations regarding melanoma detection, knowledge level: high, impact on the project: high

C. Hands-on users of the product

1) **Patients Concerned with Skin Health**
   - **User Role:** Individuals who are proactive about their skin health, regularly monitor moles, and want a quick way to assess the risk of melanoma.
   - **Domain Knowledge:** Beginners or people with an intermediate level of knowledge about health risks related to skin changes.
   - **Technology Experience:** Mixed levels – ranging from beginners to advanced smartphone users.
   - **Other Characteristics:** Health-conscious, aware of skin cancer risks, interested in using the app to track changes in their skin over time.
   - **Priority:** Primary user
2) **Dermatologists and Other Medical Specialists**
   - **User Role:** Medical professionals who use the app as an additional tool when working with patients.
   - **Domain Knowledge:** Experts with strong knowledge in skin cancer and related health issues.
   - **Technology Experience:** Moderate to high – familiar with diagnostic technology and health applications.
   - **Other Characteristics:** Professional approach, high clinical awareness, requiring accuracy and reliability in results.
   - **Priority:** Secondary user
3) **Caregivers for Individuals at Risk of Skin Cancer**
   - **User Role:** Caregivers for elderly people or individuals at higher risk of melanoma, monitoring changes in the patients' skin.
   - **Domain Knowledge:** Intermediate understanding of skin health.
   - **Technology Experience:** Varies – depending on the caregiver's familiarity with technology.
   - **Other Characteristics:** Empathetic, responsible, and needing easy access to photos and tracking of skin changes.
   - **Priority:** Primary user

D. User story - Patient Persona

- **Personal information:** Jan Kowalski, 31 years old, Programmer, lives in Wrocław (Poland), his hobby is bouldering
- **Likes:** optimizing and automating things around him, using digital solutions, keeping healthy diet
- **Doesn't like:** wasting his time, standing in long queues for doctor appointment
- **Motives:** Jan recently noticed a new mole on his shoulder and is a bit concerned because he likes to keep finger on the pulse regarding his health. He's looking for an easy-to-use app that can help him monitor this and other moles for any changes over time. Jan is looking for an app that gives him clear, visual feedback on his skin health and an easy way to track his skin changes.

V. CONSTRAINS

*A. Technological **[Must have]***

1) Use React Native for the mobile app development.
2) Use Python with FastAPI for REST API implementation.
3) Use of previously developed CNN model.
4) Use of an environment that provides adequate computing power.

*B. Temporary limitation **[Must have]***

1) The initial prototype design must be delivered before 26 November 2024.
2) The project must be implemented and ready for use before 16 January 2025.

*C. Financial **[Should have]***

1) Using only free or open sourced solutions for development to keep costs to a minimum.
2) Using free cloud solutions to provide the computing resources needed.

VI. Functional requirements

A. Requirement: Machine Learning Model for Melanoma Recognition

ID F001

Description

The system shall provide a machine learning model specifically trained for recognizing melanoma in uploaded images, delivering a classification result with a confidence score.

Priority

Must have

Rationale

Accurate detection of melanoma is critical to the application's purpose, assisting users in early diagnosis and prompt medical consultation.

Fit Criterion

The model should achieve at least 80% accuracy on a benchmark melanoma dataset and provide a confidence score for each classification.

Supporting Materials

Performance evaluation metrics.

Dependencies

None

B. Requirement: RESTful API for Image Classification

ID F002

Description

The system shall provide a RESTful API that enables users to upload images and receive a classification result based on the trained model.

Rationale

API is the core functionality, allowing external applications to utilize the classification model.

Priority

Must have

Fit Criterion

The API accepts a valid image input and returns a classification response in JSON format within 2 seconds for images less than 2MB in size.

Supporting Materials

API documentation outlining endpoint usage and request/response formats.

Dependencies

F001

C. Requirement: Mobile Application Interface

ID F003

Description

The system shall provide a mobile application interface that allows users to interact with the API and access the image classification model.

Priority

Must have

Rationale

A mobile interface enhances accessibility, allowing users to classify images directly from their devices.

Fit Criterion

The application must load and be fully functional on Android and iOS, providing a clear interface for uploading images and viewing classification results within 3 seconds.

Supporting Materials

UI design prototypes, usage instructions.

Dependencies

F002

D. Requirement: Image Storage on Server

ID F004

Description

The system should provide a feature for storing images on a server, allowing users to save and access a history of classified images.

Priority

Should have

Rationale

Storing images enables users to maintain records of classifications, improving usability for frequent users.

Fit Criterion

Users can view and retrieve past classifications via the server or cloud, with storage maintained for at least 30 days.

Supporting Materials

Details of self-hosted or cloud storage solutions, data retention policies.

Dependencies

F002

E. Requirement: Third-Party Authentication for User Access

ID F005

Description

The system shall support user authentication via third-party services, providing enhanced security for user access.

Priority

Could have

Rationale

Using third-party authentication adds a layer of security and simplifies access control.

Fit Criterion

Authentication should be implemented through Firebase or OAuth2 protocols.

Supporting Materials

List of supported authentication providers, data protection policy.

Dependencies

F004

F. Requirement: Integration with Healthcare Providers

ID F006

Description

The system shall support direct integration with healthcare systems and electronic medical records (EMR),

allowing sharing of classification results with healthcare providers.

Priority

Won't have

Rationale

Integration with healthcare systems would allow classification results to be shared with medical professionals, facilitating patient follow-up and clinical decision-making.

Fit Criterion

The system integrates with EMR systems, using standardized APIs to transmit classification results directly into patient records, with a secure authentication mechanism to ensure data privacy.

Supporting Materials

API integration guide for healthcare systems, security and compliance policies.

Dependencies

F002

G. Requirement: Advanced Analytics and Reporting

ID F007

Description

The system shall include advanced analytics and reporting features that provide detailed insights beyond basic classification results, enabling trend analysis and visualizations for user insights.

Priority

Won't have

Rationale

Advanced analytics would allow users to view patterns over time, enhancing the application's value for frequent monitoring and health tracking.

Fit Criterion

The system generates visual reports, trend analyses, and heatmaps, allowing users to view and export detailed insights over a specified time period.

Supporting Materials

Design documentation for the analytics dashboard, user guide for data interpretation, and API documentation for data export options.

Dependencies

F002

VII. Nonfunctional requirements

A. Aesthetic Requirements

ID NF001

Description

The mobile application and web interface shall have a modern and intuitive design, using a color scheme that is easy on the eyes.

Priority

Should have

Rationale

Aesthetic quality improves user satisfaction and increases the likelihood of adoption.

Fit Criterion

The user interface should use a color palette of cool colors (e.g., blue, green) with high contrast for readability. UI elements should follow the Material Design or Human Interface guidelines for mobile applications.

Supporting Materials

Mockups and UI design guidelines.

B. Performance Requirements

ID NF002

Description

The system shall respond to user actions, such as uploading an image and receiving a classification result, within a maximum of 2 seconds for images less than 2MB in size.

Priority

Should have

Rationale

Fast response times are critical for usability, especially in a medical context where users expect immediate feedback.

Fit Criterion

The system shall be tested with various network speeds and device types to ensure a response time of less than 2 seconds.

Supporting Materials

Performance testing and benchmarks.

C. Security Requirements

ID NF003

Description

The application shall ensure that all user data, including uploaded images and user information, is encrypted both in transit and at rest.

Priority

Must have

Rationale

Protecting sensitive user data is essential, particularly in a healthcare-related application where privacy and data protection are especially important.

Fit Criterion

Data should be encrypted using at least AES-256.

Supporting Materials

Security protocols.

D. Usability Requirements

ID NF004

Description

The application shall be intuitive and easy to use, requiring no more than three steps to complete a typical task, such as uploading an image and viewing the classification result.

Priority

Should have

Rationale

High usability increases user engagement and decreases the risk of errors, particularly for non-technical users.

Fit Criterion

The application interface provides a clear, labeled workflow allowing users to complete the primary task in no more than three or four actions.

Supporting Materials

Usability tests, sample user feedback.

VIII. Prioritisation of requirements

A. Must have

1) **Detection of most cases of melanoma:** The system must accurately detect melanoma using machine learning algorithms with an accuracy above 80%.
2) **Technological constraints:** The project must use React Native for the mobile app and Python with FastAPI for the REST API.
3) **Data security:** Data must be secured against unauthorized access.
4) **Mobile application:** A functional mobile application that allows users to interact with the API and classification model.
5) **API for image classification:** A RESTful API that enables image upload and returns classification results.
6) **Project deadlines:** The initial prototype must be delivered before 26 November 2024, and the project must be ready for use before 16 January 2025.

B. Should have

1) **Storing images on a server:** Provide functionality for users to save and access their image history.
2) **System performance:** Ensure quick API response times, ideally up to 2 seconds.
3) **Interface utility:** Develop an intuitive and user-friendly mobile interface.
4) **Financial constraints:** Utilize free or open-source solutions and free cloud resources to minimize costs.

C. Could have

1) **Authorization:** Implement user authentication using third-party services for enhanced security.
2) **Scalability:** Design the system to handle an increasing number of users efficiently.
3) **Increase public health awareness:** Incorporate features to educate users about melanoma and skin health.

D. Won't have

1) **Integration with healthcare providers:** Direct integration with healthcare systems or medical records is beyond the scope of this project.
2) **Advanced analytics and reporting:** Detailed analytics beyond basic classification results will not be included.

IX. Prototype design

A. Selection of functionalities included in the prototype

Based on the project requirements and priorities, the prototype includes the following key functionalities:

- **User Interface:** A mobile application developed using React Native, providing basic interface for users to interact with the system.
- **Authentication:** Basic user authentication mechanisms with login and password (stored in API's database).
- **Camera Access:** The ability for users to capture images of their skin lesions directly through the mobile application.
- **Image Upload:** Functionality to upload captured images to the backend server for analysis.
- **RESTful API:** A backend API built with Python and FastAPI to handle image submissions and communicate with the machine learning model.
- **Image Classification Model:** A convolutional neural network (CNN) model that processes the uploaded images and classifies them as melanoma or benign.
- **Image Storage:** Storage of uploaded images on the server to allow users saving uploaded images.

Functionalities not included in the prototype at this stage are:

- **Cloud Services:** Integration with cloud storage and third-party services is not included.
- **Image comparison:** Allowing the user to compare (by overlaying) images and observe changes over time.

B. Use cases

For the prototype, we wanted to focus mainly on the core functionalities of our system. The application should be able to send benign pictures made by users through REST API to the classification model and receive the result. Figure 1 presents use cases that our minimum viable product (MVP) should handle.

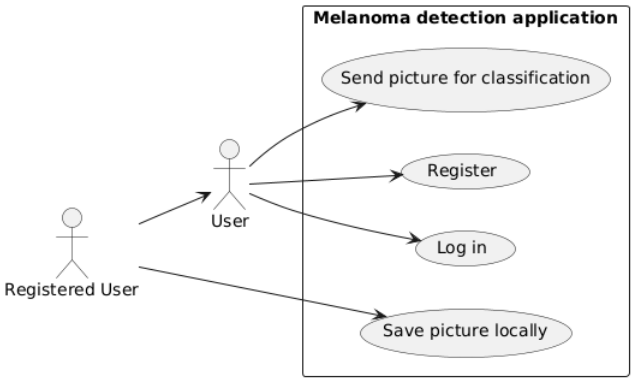


Fig. 1. Use case diagram

1) Send Picture for Classification:

- **Actors:** User
- **Goal:** Upload a picture for melanoma classification.
- **Process:**
  1) The user captures an image within the application.
  2) The application sends the image to the backend for analysis.
  3) The classification result is displayed to the user.

- **Outcome:** The user receives diagnostic feedback to decide on further action.

2) Register:

- **Actors:** User
- **Goal:** Create an account to enable advanced features such as saving images locally.
- **Process:**
  1) The user provides required details, such as email and password.
  2) The application validates and stores the user information securely.
  3) A new account is created, enabling access to additional functionalities.
- **Outcome:** The user becomes a Registered User.

3) Log In:

- **Actors:** Registered User
- **Goal:** Access the application's advanced features by logging in.
- **Process:**
  1) The Registered User provides their login credentials.
  2) The application verifies the credentials and grants access.
- **Outcome:** The Registered User accesses personalized functionalities.

4) Save Picture Locally:

- **Actors:** Registered User
- **Goal:** Save analyzed images for future reference.
- **Process:**
  1) The Registered User selects an analyzed image.
  2) The application saves the image locally on the device or in the user's account.
- **Outcome:** The user retains a record of their skin health over time.

C. Identification of Key System Components

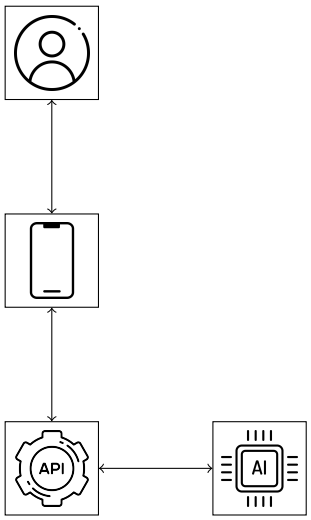
The key components of the system are essential for providing the core functionalities. These components include:

- **Registration and Login (Authentication):** This feature lets users sign up and sign in to the app. It keeps user info and only lets people with the right credentials use the app's special tools. The login system checks user details and handles safe login sessions.
- **Image Upload for Classification:** This feature lets users upload images of skin spots from the app. These pictures are then sent to the API for checking. This is the most important part of the app.
- **RESTful API Execution:** The API receives pictures from the app. It processes these images, sends them to the AI model, and sends the results back. This server is the central part of the system, connecting the app's front-end and back-end.

These parts work together to make the app easy to use, from uploading/taking pictures to getting results. The login part keeps things secure, while the picture upload and server parts handle the main job of checking for melanoma.

D. System architecture

The architecture of our system consists of a mobile application, a backend server, and a machine learning model. The mobile app allows users to capture and upload images, register, log in, and save images locally, while the backend processes these images via a RESTful API using a trained classification model.



E. Component diagram

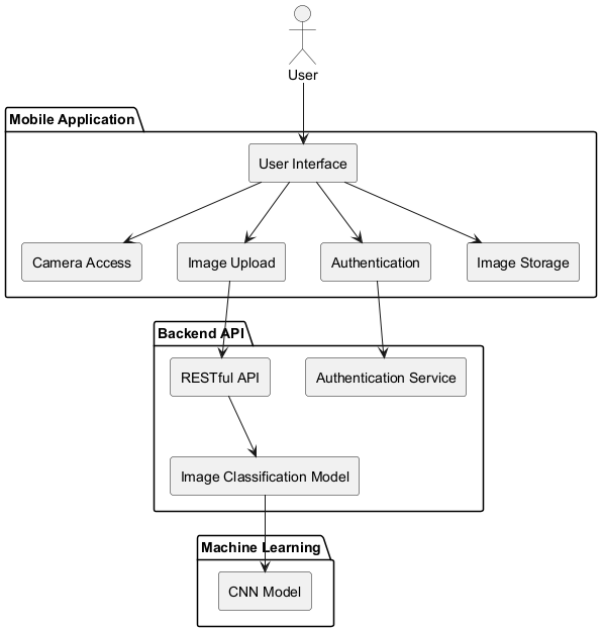


Fig. 2. Component diagram

X. Schedule of the Project

A. Tasks

1) **Design and Planning:**
   - Analysis of application requirements.

- Research on system architecture (division into app layers, designing the communication between layers, choosing the technologies and providers).

2) **Frontend:**
   - Creating registration view.
   - Creating login view.
   - Creating the camera view, responsible for capturing the images.
   - Creating the analysis view, with the results of diagnosis.
   - Creating the gallery view, responsible for viewing historical images.
   - Integration with the backend.
3) **Backend (REST API):**
   - User registration
   - User login.
   - Handling image uploads to the AI model.
   - Storing images in the database.
   - Endpoints for saving and retrieving user data.
4) **ML Model:**
   - Improving the ML model for melanoma diagnosis.
   - Assesing the model performance using metrics
   - Integration of the model with the backend.
5) **Testing and Deployment:**
   - Testing the application functionalities.
   - Optimizing application performance.

B. Task Dependencies

- The REST API backend must be partially ready before full ML model integration.
- The REST API backend must be fully ready before full ML model integration.
- The works on Frontend, Backend and ML layers can be run in parallel.
- Testing can begin after all the integrations are done.

C. Task Duration Estimates

Task	O (days)	M (days)	P (days)	E (days)
Requirements analysis	1	2	3	2
System architecture design	1	2	3	2
Registration view	0.5	1	1.5	1
Login view	0.5	1	1.5	1
Camera view	0.5	1	1.5	1
Analysis view	0.5	1	1.5	1
Gallery view	0.5	1	1.5	1
Integration with backend	1	2	3	2
Registration view	1	2.5	4	2.5
Login view	1	2.5	4	2.5
Integration with AI	2	2.5	10	2.66
Image management	2	3	5	3.33
AI Model training	4	6	10	6.67
AI Model performance	1	2	3	2
AI Model API integration	2	3	5	3.33
Testing	3	5	7	5
Performance	1	2	4	2.33

TABLE I
TASK DURATION ESTIMATES USING THE PERT METHOD

The duration of each task was estimated using the PERT method:

$$E = \frac{O + 4M + P}{6} \tag{1}$$

D. Network Diagram

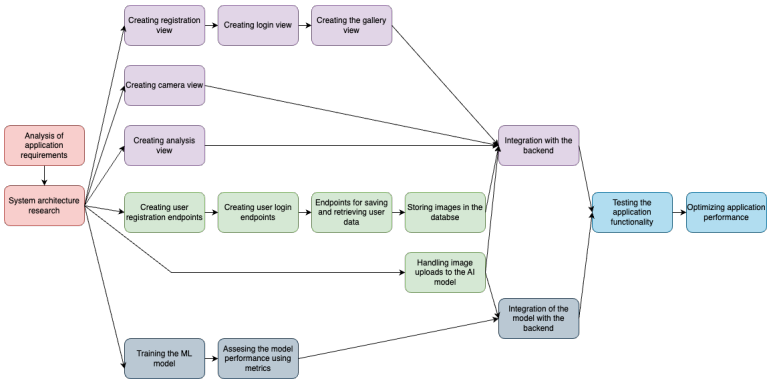


Fig. 3. Network diagram of dependencies in the schedule

E. Critical Path

The longest time path in the project is:

1) Requirements analysis
2) System architecture design
3) AI Model: training
4) AI Model: performance
5) AI Model: integration with API Testing Performance

The total duration of the critical path is **23.33 days** (Aproximately 4 weeks).

F. Task Assignment to Team Members

- **Jakub Grelowski:** Creating views and integrating with the backend.
- **Maksym Malicki:** REST API implementation and AI model integration.
- **Kamil Kochan:** Training and integrating the AI model.