Project Plan

1. Project Title

Skin Cancer Detection App Using Machine Learning

2. Team Members and Email Addresses

Neva Ildem - zildem2023@my.fit.edu
Sophia Zubieta - szubieta2023@my.fit.edu
Samuel Borges - borgess2023@my.fit.edu
Elif Dila Bayrak - ebayrak2025@my.fit.edu

3. Faculty Advisor

Dr. Zahra Nematzadeh
znematzadeh@fit.edu

4. Client

Dr. Zahra Nematzadeh
Department of Electrical Engineering and Computer Science

5. Date(s) of Meeting(s) with The Client

Our first meeting with the faculty advisor was on 08/24/2026 at 12:00 pm. During the meeting, we discussed the project goals, determined which tools are best suited for the project, and established the client's needs. Follow-up meetings are expected to take place every week at the same time. During these meetings, we will update the client on the progress made towards the established milestones through prototyping and testing.

6. Goal and Motivation

Skin cancer is one of the most common and potentially fatal cancers worldwide. Therefore, early detection is crucial for improving survival rates. However, recognizing potential metastatic skin lesions is not always easy or accessible, and individuals may not realize when a lesion requires further medical treatment. The goal of this project is to provide users with a simple, accessible approach to make a preliminary assessment of skin lesions using a mobile application. By making this study easily accessible, the application can help people become more aware of their skin condition and seek medical attention when necessary.

The purpose of this project is to create a mobile application that uses machine learning to evaluate photos of skin lesions and make real-time predictions regarding whether a lesion is benign or malignant. The application would enable users to easily upload images of skin lesions and receive preliminary predictions on their mobile devices. Furthermore, the application will allow users to track their skin health over time by displaying the progress of their skin lesion predictions on different days. The application aims to serve as an accessible tool for skin health awareness and monitoring, rather than to replace professional medical diagnosis.

7. Approach (Key Features of the System)

Image Upload and Processing

The user can upload the desired image of a skin lesion into the system. The user can choose whether they wish to upload an image from their photo gallery or files, or to take a live picture using their camera interface. To ensure the consistency of photos, the system will process the selected image. The image can be cleaned up and resized to fit the model's requirements, this includes normalizing the image to formulate better and more accurate final predictions. This allows the user to have a higher margin for acceptable photos for uploading.

Model Prediction & Visualization

The user can see a finalized result of the scan. The user will be able to see what the model predicts based on the skin lesion shown. The machine learning and AI used in the system will use previous datasets and be able to make a prediction as to what it believes is being shown, whether it is cancerous or benign, etc. The user will also be able to see a visual that indicates the confidence level of the prediction that has been made. There will also be a presentation of potential next steps following the prediction.

Disclaimers and Next Steps

The user can see helpful information regarding the next steps and suggestions. There will always be a disclaimer and a suggestion to visit a primary care provider if unsure about the result. The system does not produce a diagnosis and should be treated as a preliminary opinion. It is recommended that the user seek a professional opinion if the system predicts a cancerous or malignant diagnosis.

8. Novel Features/Functionalities

Predicting Skin Lesions and Visualizing Confidence

The user can receive a prediction of whether a skin lesion is benign or malignant by uploading or taking a picture of the lesion. An easy-to-understand visual indicator shows the user the model's confidence level along with the prediction. Instead of just displaying a benign or malignant label, this provides the user with additional information regarding the prediction. This feature helps users understand their skin lesions. Users will be encouraged to seek medical advice when needed, as the prediction is not meant to replace medical diagnosis.

Tracking the Progress of Skin Lesions

The user can save and monitor their previous skin lesion results. Users can compare how predictions have changed and view results from multiple dates in the application. Users can monitor their skin health over time by using the program, rather than just once for a single prediction. This feature provides users with an easy method to track progress and changes by combining skin lesion prediction with progress tracking.

9. Algorithms and Tools

The team aims to use tools for the data preprocessing and augmentation phase before using the skin lesion photos to train the model. For tasks such as image resizing, normalization, noise reduction, potential hair removal, and data augmentation, we plan to use and compare tools such as Python, OpenCV, and Keras. Flipping, rotating, zooming in and out, adjusting brightness, or contrast are examples of data augmentation techniques we plan to experiment with. The two datasets we will primarily use are ISIC and HAM10000.

We plan to use a deep learning algorithm that can categorize photos of skin lesions. We are considering using a Convolutional Neural Network (CNN) and comparing a simple CNN with transfer learning models such as ResNet50, EfficientNetB0, and MobileNetV2. The primary tools we plan to use to train the model are TensorFlow and Keras.

To evaluate the performance of the trained model, we require tools and evaluation techniques. We plan to compare and evaluate models using confusion matrices, F1 score, sensitivity/recall, accuracy, and precision. These evaluation metrics can be computed and visualized using Scikit-Learn. We can decide which model works best for the project using these measurements.

Tools that enable users to take or upload a picture of a skin lesion through a mobile application and receive a prediction from the trained model are needed for the application phase. We are planning to use TensorFlow Lite to integrate the trained model into the application and Flutter to build the application. For the progress-tracking feature, we plan to store past prediction results in a local storage database such as SQLite.

10. Technical Challenges

Machine Learning Model Selection and Integration

We plan to use machine learning as the basis of our system. Machine learning will be the main source of the prediction process for our system. The issue we have surrounds the determination of which machine learning model will be the best to use for this project and how to integrate it into a mobile application. As of right now, we are planning on using a Convolutional Neural Network (CNN) or transfer learning, but haven’t determined which approach will provide the best accuracy and performance. In addition, testing, training, and saving the model and integrating the model into the mobile application so it is able to process images and provide predictions is another difficult task the group must learn.

Image Processing and Preprocessing
The system is heavily dependent on images of skin lesions and malformations provided by users. As a result, images will vary in size, quality, lighting, and even resolution. A challenge we will face is determining how these images need to be processed before they are given to the machine learning model. Techniques need to be researched for image resizing, normalization, noise reduction, and data augmentation to ensure compatibility with the format and requirements of the machine learning model.

Model Training and Optimization
Another challenge arises with performance optimization and training. The team plans to use Python for developing and training the machine learning model using the selected datasets. We will need to determine training parameters, with minimizing false negatives being a major consideration for our system. In order to reduce the chances of our system producing a false negative, properly training and optimizing the model will be a challenge. We will need to determine parameters such as training epochs, learning rate, batch size, and other settings that will affect the model's performance. In addition, we will need to control problems such as overfitting and underfitting while training the model. We will need to optimize the model while maintaining reasonable training time and prediction performance.

11. Milestone 1 – September 28 (itemized tasks)

- Compare and select technical tools for image preprocessing, machine learning, model training, and app development
- Research and select skin lesion datasets, including ISIC and HAM10000
- Create small (“hello world”) demos to evaluate Python, OpenCV, Keras/TensorFlow, Flutter, TensorFlow Lite, and SQLite
- Begin implementing the image preprocessing pipeline, including image resizing, normalization, noise reduction, and data augmentation
- Research and compare CNN and transfer learning approaches, including ResNet50, EfficientNetB0, and MobileNetV2
- Identify technical challenges related to image preprocessing, model selection, model integration, and mobile deployment
- Compare and select collaboration tools for software development, documents/presentations, communication, and task management
- Create Requirements Document
- Create initial Design Document
- Create initial Test Plan

12. Milestone 2 – October 26

- Complete the initial image preprocessing and augmentation pipeline for the selected datasets
- Prepare, clean, organize, and split the dataset into training, validation, and testing sets
- Implement and train an initial CNN model
- Implement and experiment with transfer learning models such as ResNet50, EfficientNetB0, and MobileNetV2
- Compare model performance using accuracy, precision, recall/sensitivity, F1 score, and confusion matrices
- Analyze false negatives and optimize the model to reduce the risk of missing malignant lesions
- Experiment with training parameters, including learning rate, batch size, epochs, and augmentation techniques
- Identify and address overfitting, underfitting, and model performance issues
- Select the preliminary model for integration into the application
- Update Design Document based on the implemented system
- Update Test Plan with model and preprocessing test cases

13. Milestone 3 – November 23

- Integrate the selected machine learning model with the application using TensorFlow Lite
- Develop the mobile application using Flutter
- Implement image input functionality, including photo upload and camera capture
- Implement image preprocessing within the application before model prediction
- Implement prediction results and confidence-level visualization
- Implement preliminary next-step recommendations and required medical disclaimers
- Implement skin lesion progress tracking and local storage using SQLite
- Test the complete workflow from image capture/upload through preprocessing, prediction, and result visualization
- Test model performance and application functionality across supported devices/platforms
- Resolve technical challenges involving model integration, mobile performance, image compatibility, and data storage
- Conduct system-level testing using the Test Plan
- Update Requirements and Design Documents to reflect the completed system
- Demonstrate an initial end-to-end working prototype

14. Task Matrix for Milestone 1

Task	Elif	Neva	Samuel	Sophia
Compare and Select Technical Tools	Model Evaluation Tools/Scikit-Learn	CNN/Transfer Learning Models (ResNet50, EfficientNetB0, MobileNetV2)	Using Flutter to build app framework and assist with CNN training using TensorFlow and/or Keras	OpenCv/Python Image Processing
"hello world" demos	Model Evaluation Metrics Demo	Basic CNN/ Model Training Demo	Create Flutter app and CNN/TensorFlow Demo	Image Resizing/ Preprocessing Demo
Resolve Technical Challenges	Model Performance Evaluation	Machine Learning Model Selection	Integrating trained model to mobile app	Image Processing
Compare and select Collaboration Tools	Google Calendar/Github Projects (Scheduling)	Software Development-GitHub	GitHub Development	Google Docs/Slides
Requirement Document	Write 40%	Write 20%	Write 20%	Write 20%
Design Document	Write 20%	Write 20%	Write 40%	Write 20%
Test Plan	Write 20%	Write 20%	Write 20%	Write 40%