Skin Cancer diagnosis using Artificial Intelligence on the Cloud

Team 45
Bariture Ibaakee
Breann Grant
Abigail Thompson
Megan Eberle
Evan Nim
Alexander Lafontaine

Clients/Advisors: Ashraf Gaffar, Ashfaq Khokhar

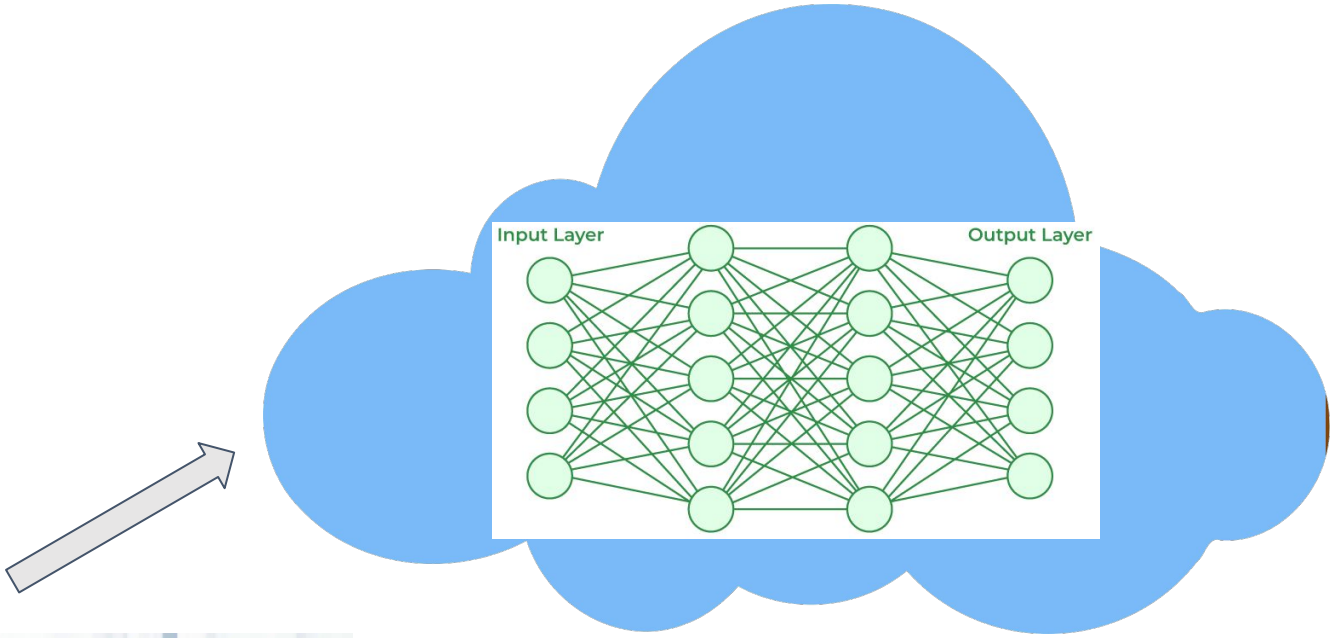
Project Vision

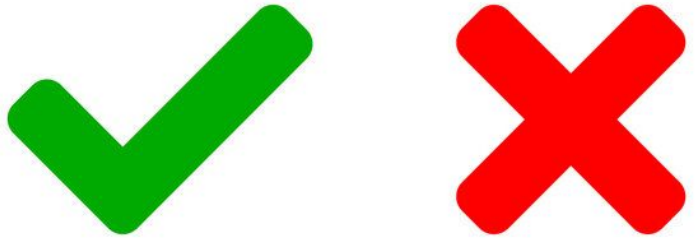
The goal of this project is to train an Artificial Intelligence (AI) model to determine whether skin cancer is present in an area of skin using just an image.

Project Vision - Use Case

A patient has a concern about an area of skin. Instead of performing a biopsy, a health professional takes a photo of the affected area, uploads it to the web interface, and within seconds receives a result predicting whether the area is cancerous or not.

Conceptual/Visual Sketch





Requirements

Functional Requirements

- UI shall allow only registered users to log in
- UI shall allow users to upload an image from their device
- When an image has been submitted, UI shall tell the user the results determined by the AI model

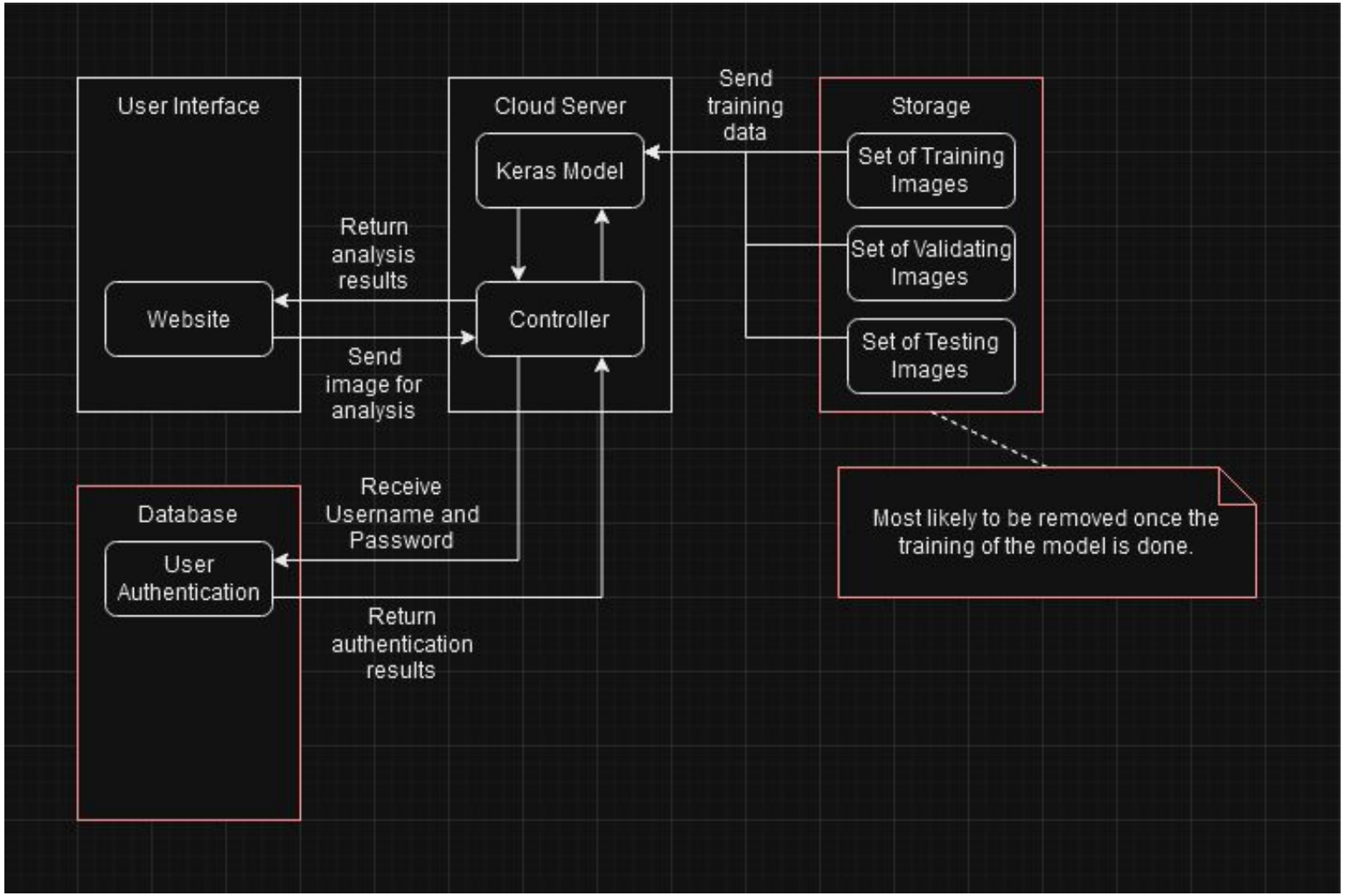
Nonfunctional Requirements

- UI shall be easy to understand and use
- AI model shall be hosted on the cloud
- AI model shall produce results with 85% accuracy
- AI model shall output results within seconds
- AI model shall gracefully degrade

Constraints

- The project must cost $0

Conceptual Final Design Diagram



System Design - Technology

- AWS or GCP
  - Used to store the A.I. model and the website’s controller.
- Keras API
  - Used to create an A.I. model that will be trained with cancer images.
- MySQL or MongoDB
  - Used to store user authentication for website.

System Design - Languages and Frameworks

- Python
  - A.I. Model
  - Controller
- React.js, Bootstrap or Angular
  - Website

System Design - Standards

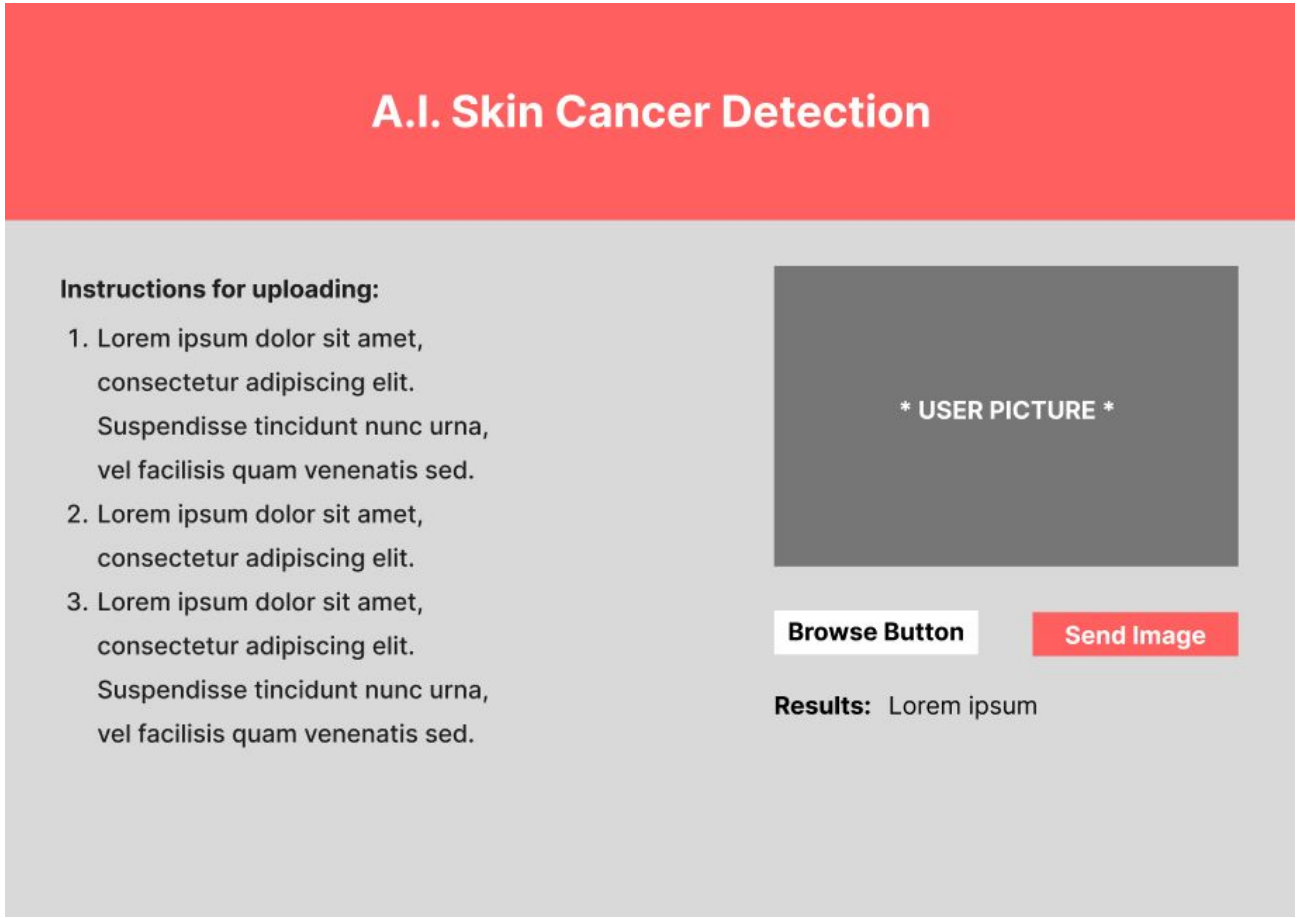
- IEEE 730
  - This is a software project and therefore should adhere to typical software quality assurance processes, similar to those laid out in IEEE 730.
- IEEE 1012
  - We'll need to be able to identify whether or not the given problem (detecting skin cancer) was able to be solved by our software as well as the consumer using it has their problem solved.
- IEEE 16326
  - Since this is a group project it'll be important to identify key points in relation to project management, specifically planning, monitoring, quality management, documentation, etc.

System Design - Standards

- IEEE 24748
  - Since the project will have a user interface for non-expert audience, it will require documentation or a tutorial.
- IEEE 29148
  - There will be various development stages required in order to create our model, IEEE 29148 talks about the provisions and requirements for these different stages.
- ISO/IEC 29119
  - Since the software project involves A.I, rigorous testing will be needed to both train and test the A.I. Therefore, great documentation and testing standards are needed.

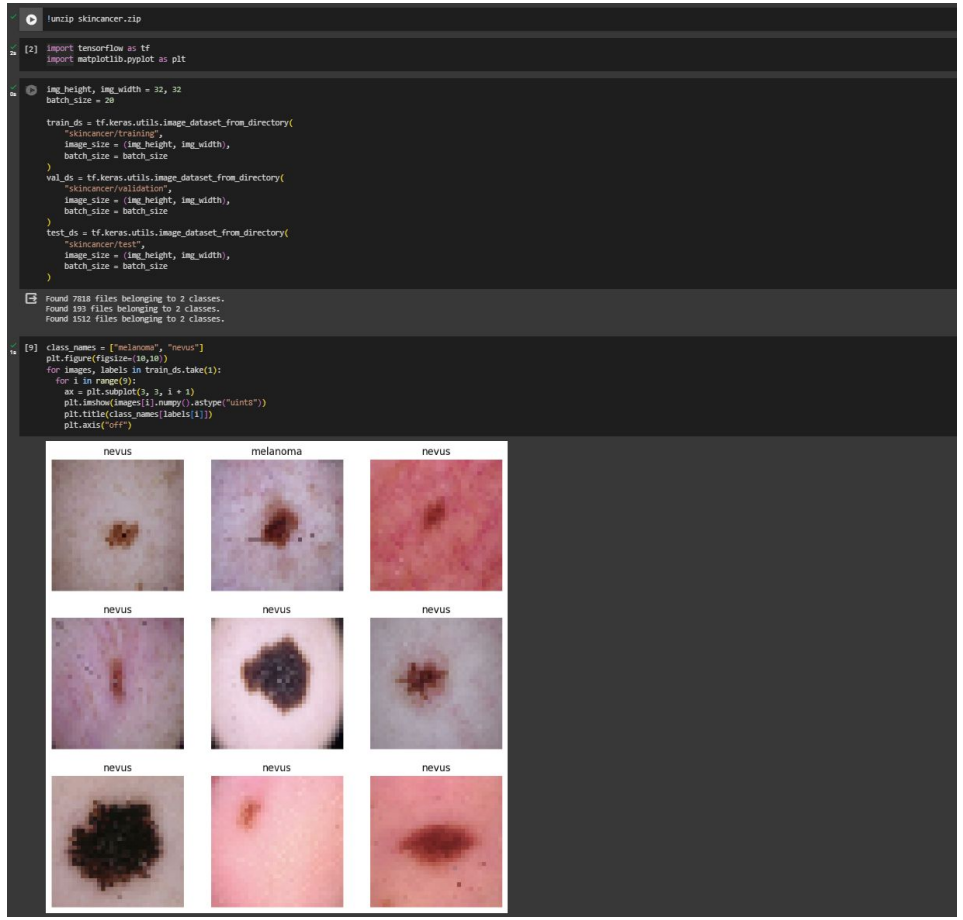
System Design - Website Design Draft

- Early draft of the website with key components.
  - Title
  - Instructions to guide user in the process
  - Reflect the picture the user is sending.
  - Results

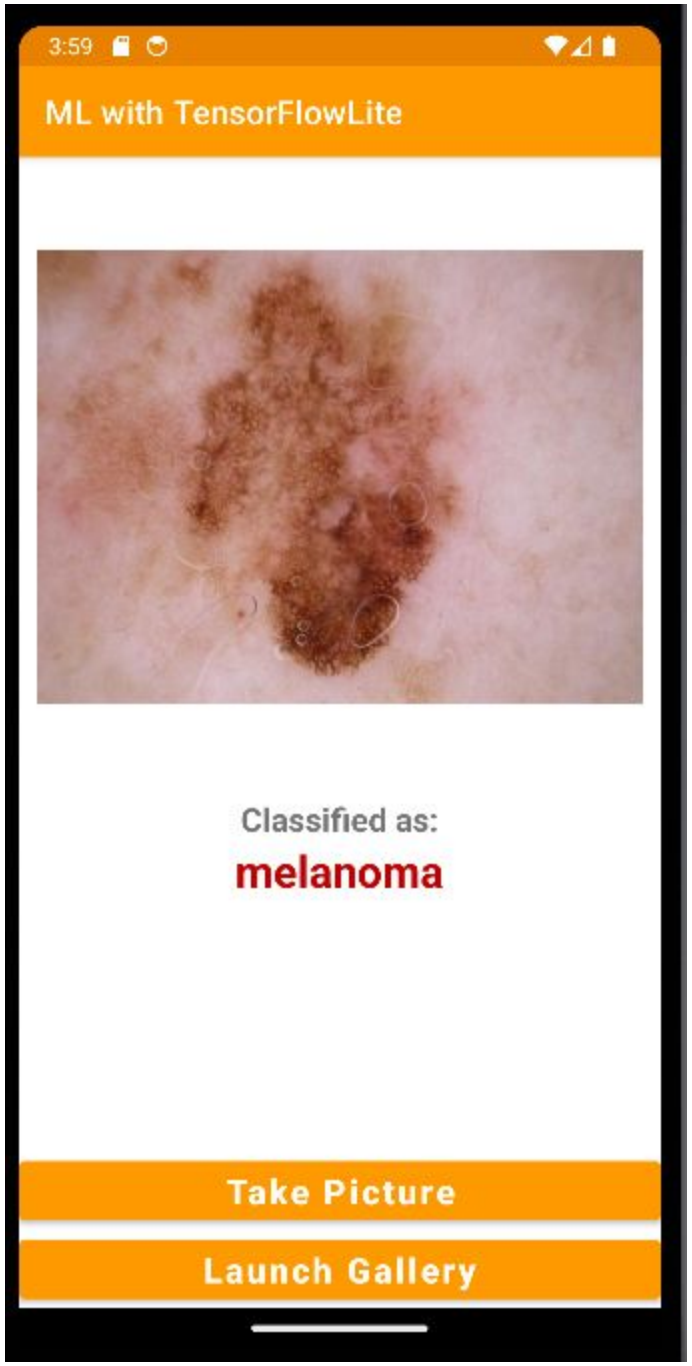


Prototype Implementations (1-3 slides)

- The current implementation of a prototype currently consists of a model being ran on Google Colab.
- The current prototype has an accuracy of about 80% right now which is on par with the current models of our size as of now.
- It currently is being trained off of 8000 images right now and is expected to increase once we move onto a bigger model.
  (Note: It's only been trained on images of nevus (moles) and melanoma)



Quick Demo


3:59
ML with TensorFlowLite
Classified as:
melanoma
Take Picture
Launch Gallery

Design Complexity

- Components/Subsystems
  - Database: Contains all the images for training the model.
  - Cloud Server: Contains the Deep Learning A.I.
  - User Interface: Sends input images and receives output from the cloud server.
  - Model (neural networks): Structure of the A.I.
  - Website: Website for users to read, send images, and receive results.
  - Controller: Communicates between modules and controls the flow.

Project Plan: Schedule

week	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
Train on AI and cloud computing																			
Introduction to AI skin cancer practices																			
Gain knowledge about our domain and investigate solutions																			
Small AI model																			
Improve model accuracy																			
Port model to cloud and train																			

Project Plan: Risks and Mitigation

Agile project can associate risks and risk mitigation with each sprint.

1. Difficulty transferring AI model to the cloud - 0.3
2. The training for our AI could be off resulting in bad outputs - 0.5 - We would prevent this by starting with smaller models to prevent time being wasted if the model was trained incorrectly.
3. Slow performance - 0.3
4. Refactor UI code - 0.1
5. Difficulty understanding some of the tools - 0.2
6. Not reaching the desired level of accuracy - 0.4

Design Context

Area	Description	Examples
Public health, safety, and welfare	Our project impacts any person who may have skin cancer as well as doctors by giving them a less invasive option to identify skin cancer.	Reduces need for invasive procedures.
Global, cultural, and social	People who live in regions that get more sun may be more impacted by this project.	People closer to the equator may get more sun than those who live further away.
Environmental	Our project could contribute to climate change by burning fossil fuels and increasing greenhouse gasses.	Since AI uses a lot of computing, it uses a significant amount of energy.
Economic	Our project could decrease the financial responsibility of patients and insurance providers.	Our project provides a non-invasive method of diagnosing skin cancer, which is generally less expensive.

Test Plan

- Unit Testing
  - Create test cases using Python for all the major Keras functions.
  - We will also test units within our UI, such as API calls.
- Interface Testing
  - We will test the interface between the UI and the AI to confirm that the images are successfully being sent.
  - We will also ensure that the UI receives the results from the AI model.

Test Plan

- System Testing

  Planned Implementations
- Check if A.I. is identifying cancer images 85% of the time.
- Check if A.I. is not identifying cancer images as cancer images.
- Check if A.I. is identifying a group of cancer images and not others.
- Check if A.I. is identifying images with peculiar characteristics instead of cancer images.
- Check if A.I. can identify cancer images outside of the training images.

Test Plan

- Regression Testing
  - We will have a branching strategy within Gitlab to ensure that we do not break our old functionality.
  - Once we have tested that new implementations do not break the old functionality we can merge the branch into the main.
- Acceptance Testing
  - We will demonstrate the design requirements, both functional and non-functional, are met by thorough testing.
  - We will also have videos that walk through our design and show that it is functional.

Conclusions

- We have gotten acquainted with how AI works and how the models are trained and used.
- We have our dataset downloaded and have started training smaller subsets of the dataset with a Keras sequential model.
- Our goal is to train a model using the entire dataset and set up an environment on the cloud.
- We are also planning on making a UI to be used by medical professionals.
- Our plan now is to train more models and start testing, while also working on the UI in parallel.