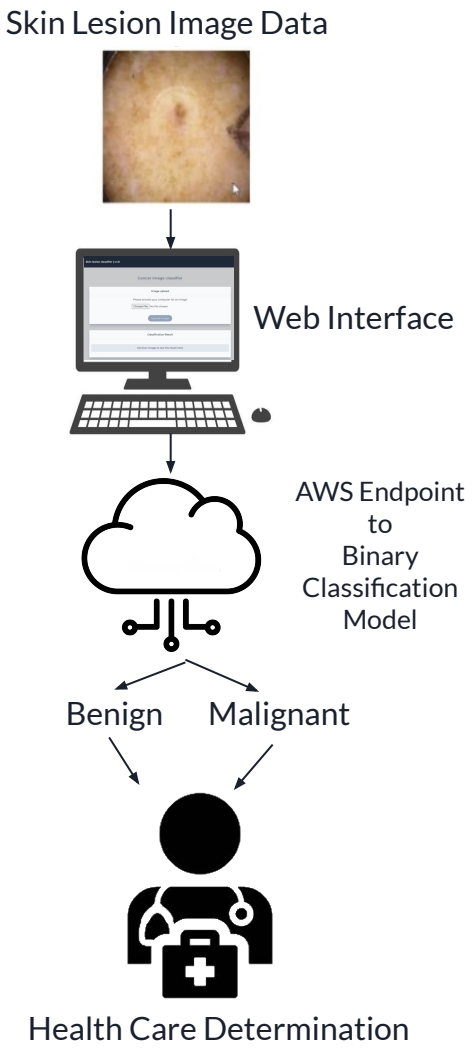
Skin Lesion Classification

Team sdmay23-05

Client/Advisor: Dr. Ashraf Gaffar, Teaching Professor [E CPE]

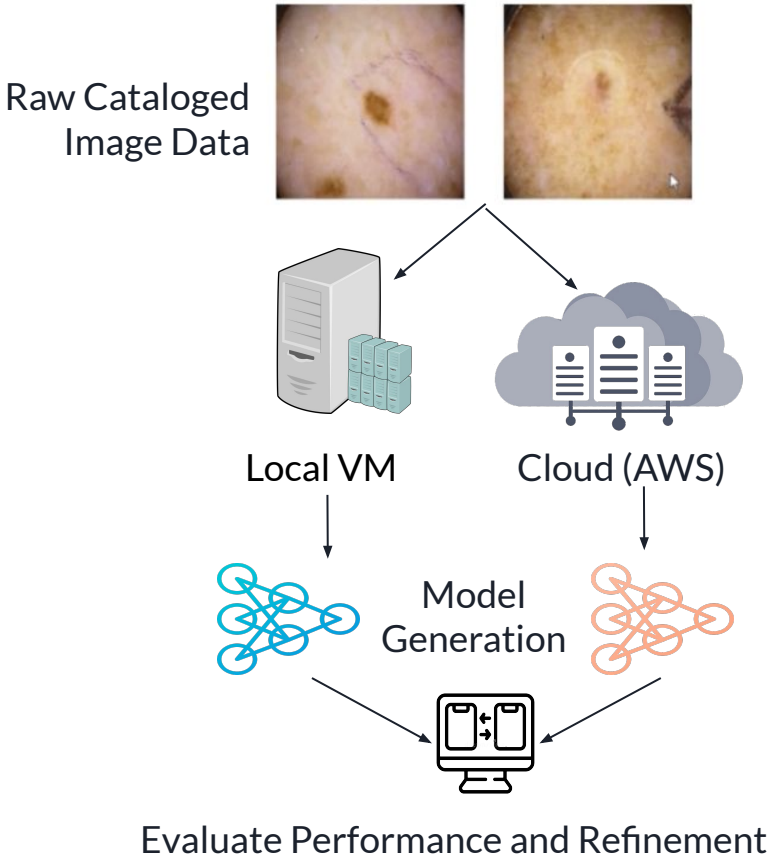
Introduction

- Classification of skin lesions
  - Benign
  - Malignant
- Develop binary image classification AI model
- Evaluate and Compare Performance on
  - Local Environment
  - Cloud Environment
- Graphical User Interface for easy utilization by medical professionals
  - AWS Endpoint (back end) for harnessing web gateway into generated model
  - Web interface (front end) for uploading and requesting prediction from model



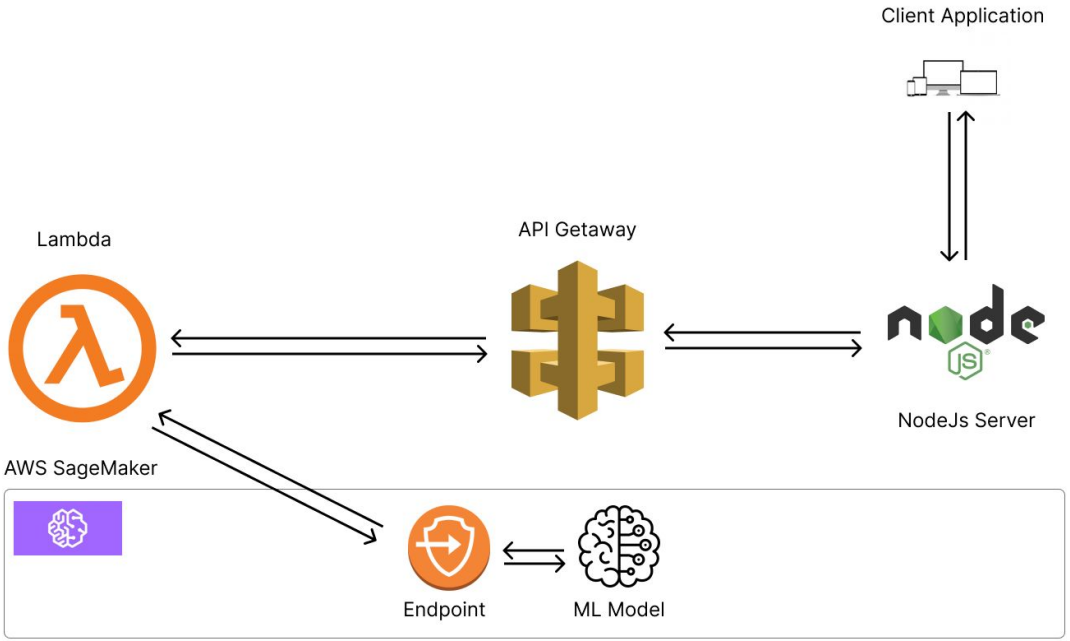
Implementation Architecture

- AI Model
  - Binary image classification neural network
- Virtual Machine (Ground)
  - CPU and GPU resources
  - Conda TensorFlow environment for training
- Amazon Sagemaker Notebook (Cloud)
  - CPU resource
  - JupyterLab IDE
  - Conda TensorFlow environment for training
- User Interface
  - AWS Endpoint
  - AWS Lambda Function
  - Node.JS
    - Front end with React.JS
    - Helper back end for image conversion with Heroku



Deployment

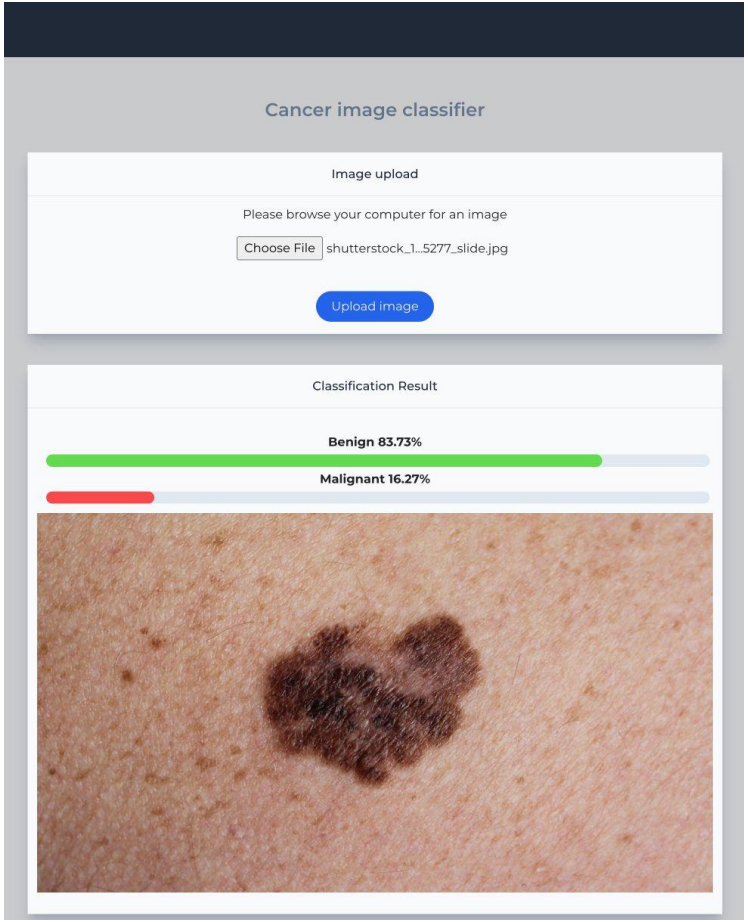
- Saved model.h5
- Sagemaker and endpoi
- Lambda function
- Api gateway and post n
- Nodejs server



Graphical User Interface

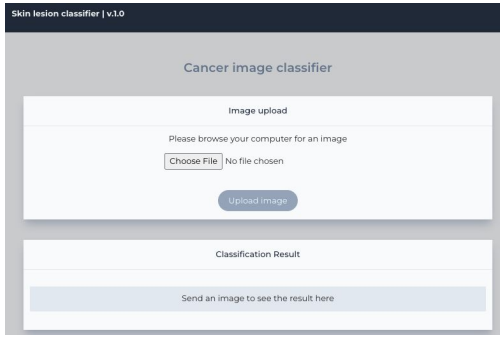
- SPA (Single Page App)
- Simple and Easy to use UI
- Responsive UI
- Web based UI
- Built with React.js

Result screenshot

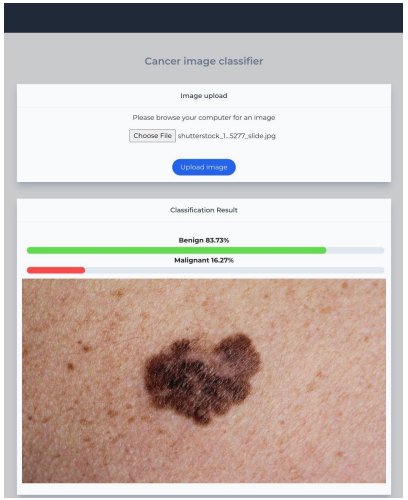


Work Accomplishments: User Interface

- AWS Endpoint built upon existing trained model [On-cloud training]
- AWS Lambda Function created to link API gateway and endpoint
- Single-Page Application Web-based Graphical User Interface for ease of interaction between user and the backend
  - NodeJS backend for server calls and image conversion
  - ReactJS for front end GUI components



Initial Landing Page on UI



Prediction Result

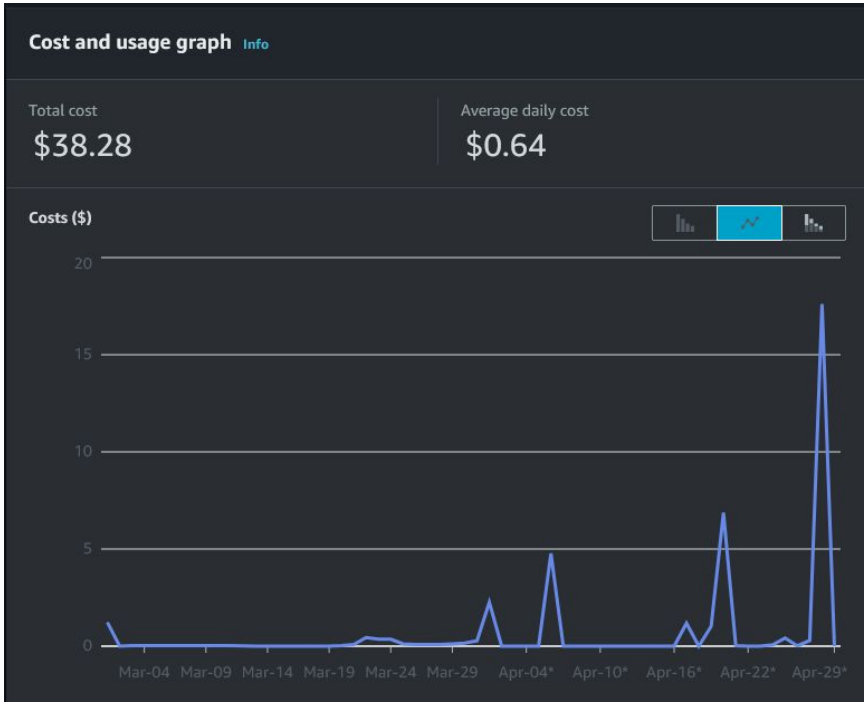
Work Accomplishments: On-Ground Training

- Binary image classification model for identifying benign/malignant
  - Dataset scaled to over 67,000 images
  - Achieved accuracy of 92%
- 27-Class model for specific diagnosis
  - Scaled to over 48,000 images
  - Achieved accuracy of 75%
- Documentation of training results

Work Accomplishments: On Cloud Training

- Binary image classification model for benign/malignant mirroring On Ground Team
  - Achieved accuracy of 86%
- Achieved comparable results to equipment used at Iowa State ETG (VM)
- We were able to analyze On-cloud training cost-effectiveness, scalability, and how it can reduce reliance on local resources
- Move ISIC Model training to the cloud

On-Cloud Training Cost for 8,000 Images



Key Contributions

- Richard Gonzalez (On-Cloud Team)
  - Configuration,Training, Analyzing, for AWS SageMaker.
- Adam Sweiger (On-Ground Team)
  - Developed and configured models, trained on Virtual Machine, worked with UI team to deploy trained model.
- Yannick Fumukani (UI Team)
  - Created UI components, and deployed the frontend to netlify for production
- Mohammed (UI Team)
  - Styled all UI components, hooked the frontend to the backend by using axios module, and deployed the frontend with Yannick

Key Contributions

- Asad Abdalla (UI Team)
  - Worked closely with On-Ground team to deploy the trained model, created the NodeJs server and API gateway
- Rashed Alyammahi (UI Team)
  - Assisted in deployment of AWS endpoint and research into various bugs encountered during process
- Meet Patel (UI Team)
  - Connected frontend with backend and implemented component

Challenges and Solutions

- **Challenge 1:** Limited Resources (storage space, computing power, time)
  - **Solution:** Increased disk size on VM, upgraded computing resources on AWS
- **Challenge 2:** Frontend team unable to send uploaded image from the client to the backend directly
  - **Solution:** Encode image to 64-bit before sending to the server for model prediction
- **Challenge 3:** Keras data format error only when training on AWS
  - **Solution:** Sagemaker's TensorFlow backend requires input data to be 'channels_last'
- **Challenge 4:** Sagemaker kernel randomly terminating in the middle of training
  - **Solution:** Ran appropriate python code in corresponding conda environment on command line

Future Work

- More training: Expand dataset, more epochs, faster computing
- Further optimize models by changing architecture and tuning hyperparameters
- New models: Classify based on other attributes (age, sex, diagnosis type, etc.)
- Host other models on UI
- Use training documentation to assist transition of large machine learning projects from on-ground to on-cloud
- Bring entire project to the Cloud

Conclusion

- Developed and trained AI model to classify skin lesions as benign or malignant with high accuracy
- Developed User Interface to run predictions on user-uploaded images
- Trained model in local and cloud environments and documented comparisons of results
- Result of comparisons: training on AWS with similar computing resources produces comparable results at a fraction of the cost
- By leveraging cloud services like AWS, we effectively scaled our research, reducing barriers such as cost and scalability.