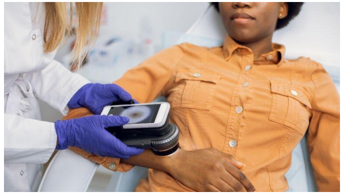
AI & Skin Cancer Detection On the Cloud

By: Evan Hanson
Anirudh Ambore
Wonjun Choi
Ziyad Alqahtani
Abdelrahman Mohammed
Mishari Alharbi

Project Goals and Vision

- **Goals**
  - Classify skin lesion images as malignant or benign
  - Compare the accuracies of different models
    - *MobileNetV2, Inceptionv3
  - Utilize AI/ML and Cloud computing
  - Benchmarking on the Cloud
  - Create an application with Flask and deploy it on AWS/GCP

- **Project Vision**
  - **Patients**: Allows users to test for skin cancer
  - **Healthcare Providers:** Presents new avenue for professional testing
  - **Healthcare Organizations**: Benchmarking to help set a baseline for future research in cloud computing
  - **System Performance**: Available to users regardless of hardware restrictions

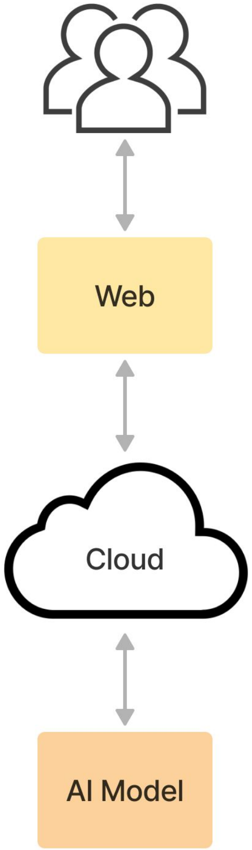


Simplified Conceptual Sketch

- **User Interface & Accessibility**
  - Simple web app created with Flask

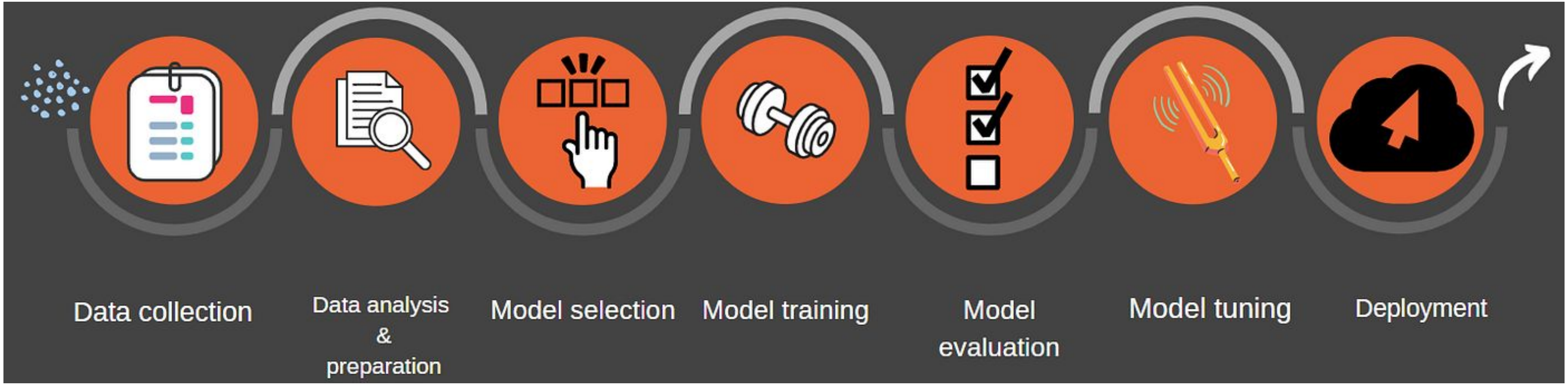
- **AI Model for Accurate Diagnosis**
  - Trained on the Cloud to classify malignant/benign images

- **Cloud Deployment**
  - Hosted on AWS/GCP
  - Improves scalability, accessibility, sustainability



System Design: AI/ML

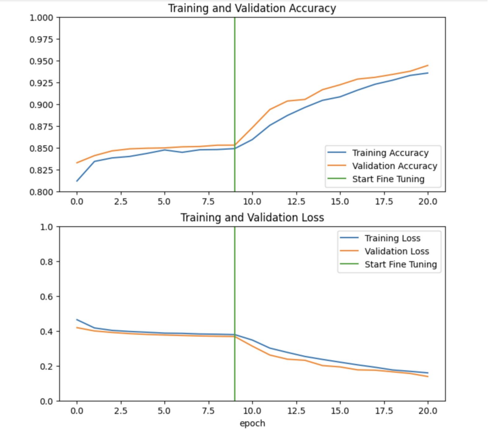
- Our system design comprises the following:

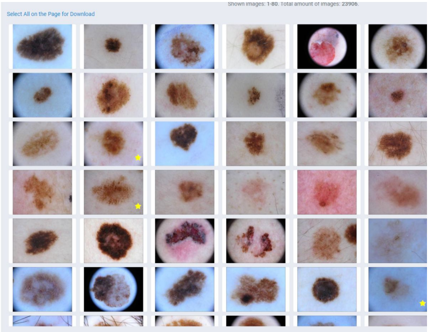


Prototype Implementation - AI Transfer Learning

For the implementation of AI model, we use the **TensorFlow/Keras** framework and train it on **AWS ec2/GCP VM**

- **Data Preparation**: ~11K images are loaded from **ISIC** and processed for testing and validation.
- **Model Building**: A pre-trained convolutional neural network (**MobileNetV2**) is introduced to learn new patterns.
- **Model Training:** The model is trained on the Cloud t o recognize cancerous features from images.
- **Evaluation:** The model's metrics are assessed on unseen data and and fine-tuned.
- **Deployment:** The model is integrated into the web application to make predictions.





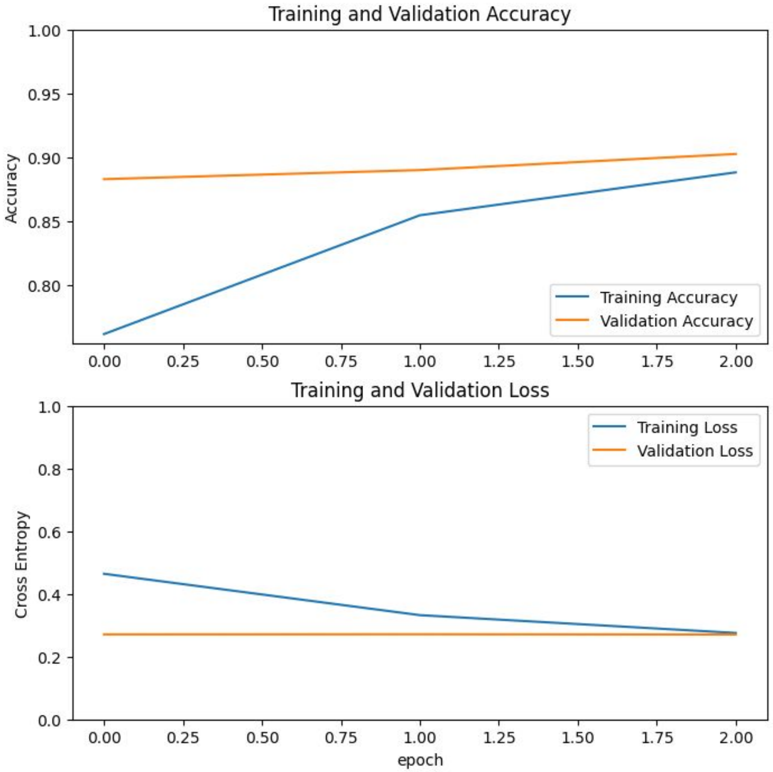
Shown images: 1-80. Total amount of images: 23906.

Select All on the Page for Download

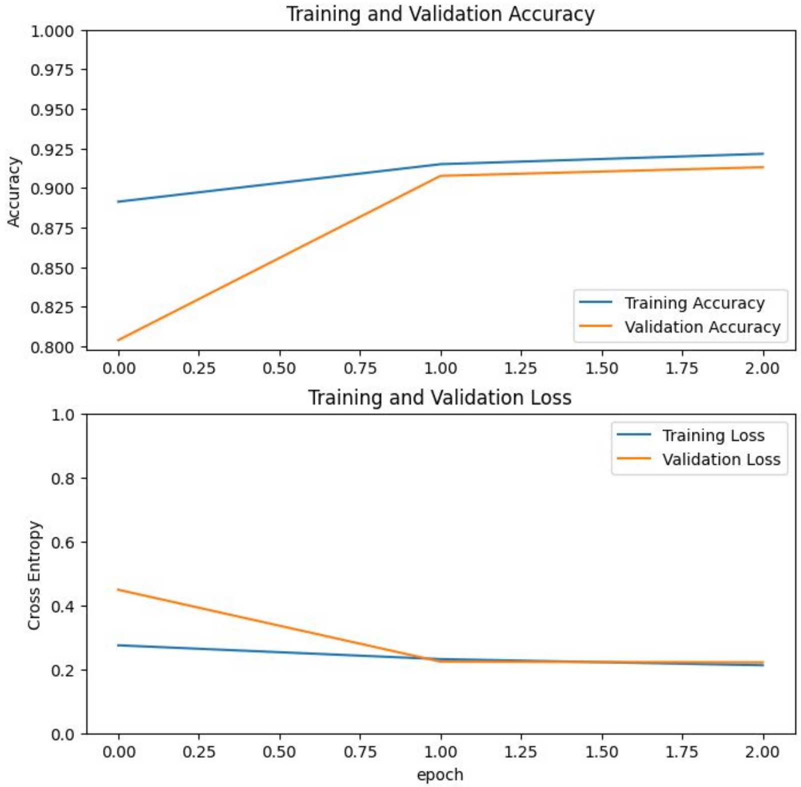
Transfer Learning

- InceptionV3 - MobileNetV2
  - MobileNet Specializes in small low-latency/power-usage with the focus for embedded systems
- Freezing layers
  - Keep layers with desired pattern recognition
  - Train on the new dataset (ISIC - Skin Cancer Images)
- Fine tuning
  - Adjusting Parameters (Output layers)
  - Reduce Overfitting/Underfitting
  - Great for specialized use-cases

Accuracy Metrics



AWS

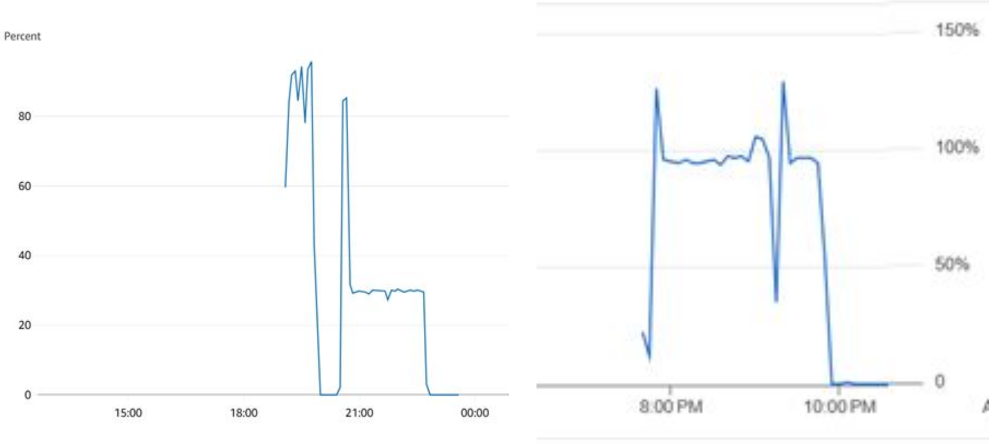


GCP

Metrics on the Cloud- AWS/GCP

- Results from Training on AWS
  - Total training time: 126.5 minutes
  - Evaluation time: 2 minutes
  - Test accuracy: 88.9%

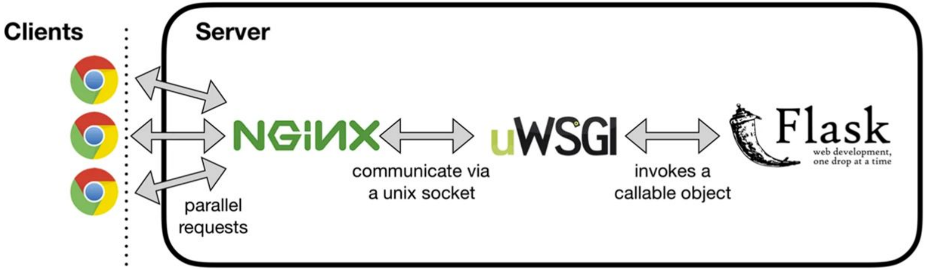
- Results from Training on GCP
  - Total training time: 121.1 minutes
  - Evaluation time: 0.6 minutes
  - Test accuracy: 90.6%



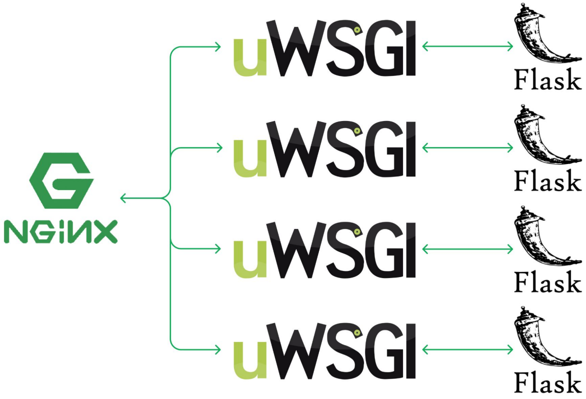
CPU Utilization (%) in AWS (Left) and GCP (Right)

AWS - Structure of Server

- Web Server : **NginX**
- WSGI (web server gateway interface): **uWSGI**
- Web Application Server : **Flask**


Clients
Server
NGINX
uWSGI
Flask
web development, one drop at a time
parallel requests
communicate via a unix socket
invokes a callable object

- Load Test Results (500 user load)
  - with a single process
    - Missing some results
    - Server fails and freezes
  - with 4 processes
    - Saved the results of all requested images
    - Continuously maintain and handle load

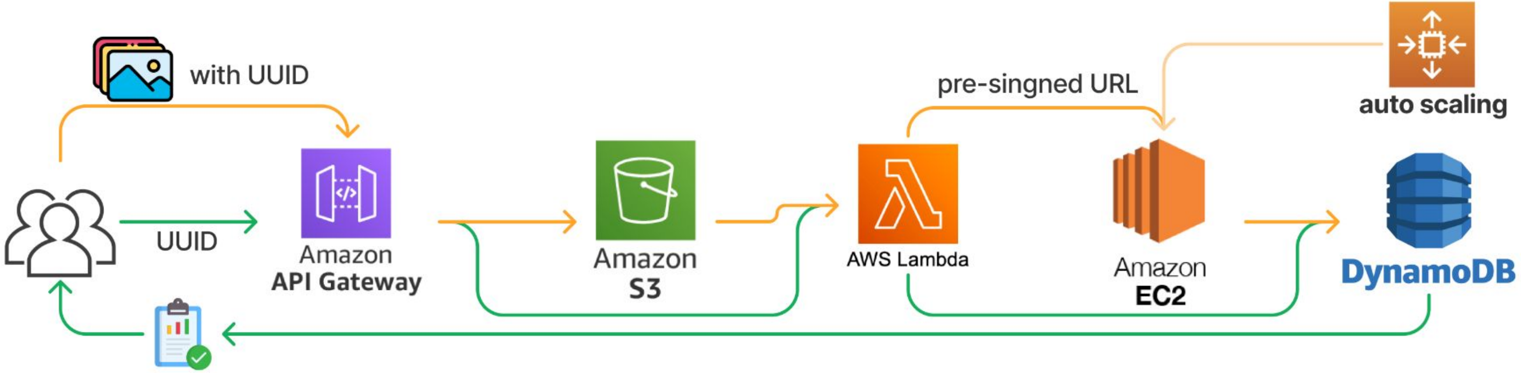


AWS Architecture

- Serverless Architecture
  - Designed for easy maintenance and high traffic
  - Image Processing with AWS Lambda

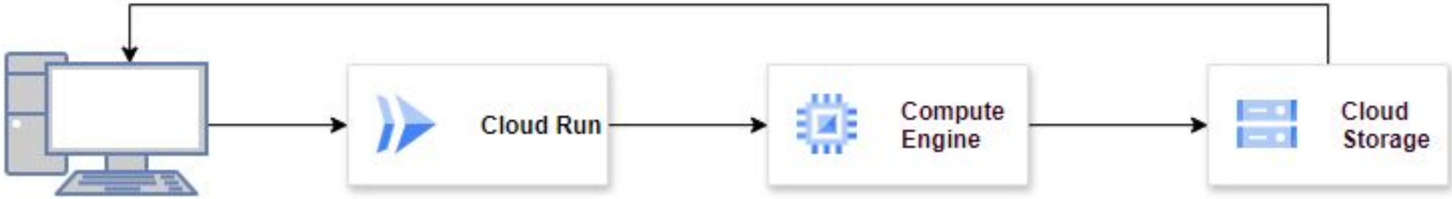
- Security and accuracy
  - UUID for Data Uniqueness
  - Pre-Signed URLs for Secure Data Transfer

- Scalability (auto scaling)
  - Maintain performance even as high traffic
  - Capable of providing non-stop services.



GCP Architecture

- Using Google Cloud services such as Cloud Run, Compute Engine, and Cloud Storage.
- Cloud Run uses the default type of Compute Engine service with broad IAM permissions.
- Cloud Storage stores the app resources, such as images, prediction percentages, and software packages.



Learning Adaptation & Challenges

- AI
    - Transfer learning
    - Training the model on Cloud

- Cloud
    - Using Flask instead of Flutter

- Challenges
    - Lack of dataset with darker skin color
    - Model creating

Thank you!