Big Data Systems

Image Classification

Sophia Williams

Wilmer Maldonado

Victor Teelucksingh

Introduction

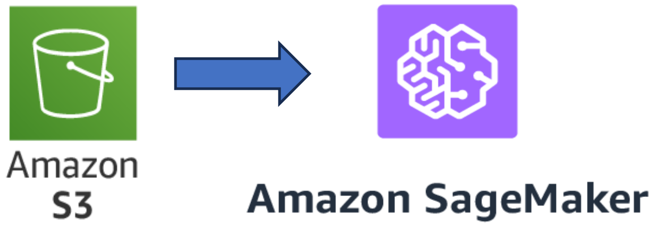
- Processing big data poses several significant challenges
  - Storage
  - Data pipeline management
  - Computing power and training time
  - Analysis and interpretation
- **Goal:** Combine the use of AWS services for large data processing and deep learning models to assist the elderly and people who are visually impaired to better navigate their homes.
  - Deploy model by developing an app (mobile or web) to classify user images

IMAGENET

Meta

Model Architecture

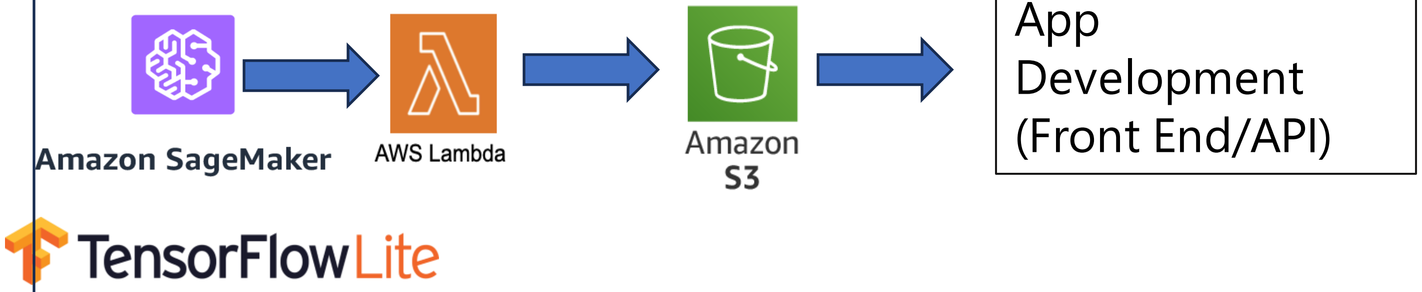
Model Training Process Flow



Amazon S3

Amazon SageMaker

Inference on new images



Amazon SageMaker

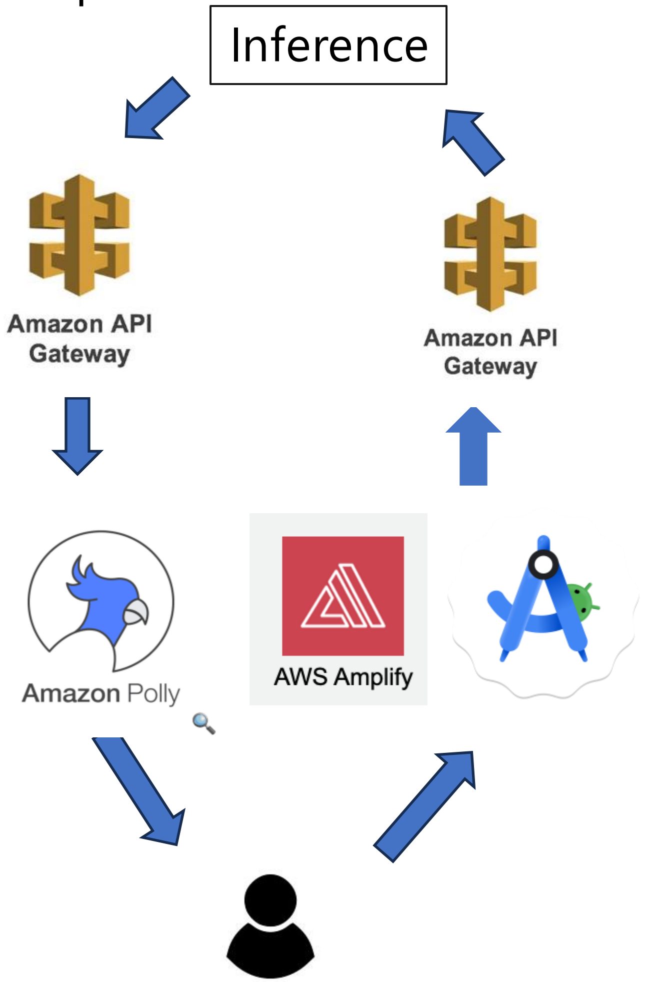
AWS Lambda

Amazon S3

App Development (Front End/API)

TensorFlowLite

App development



Inference

Amazon API Gateway

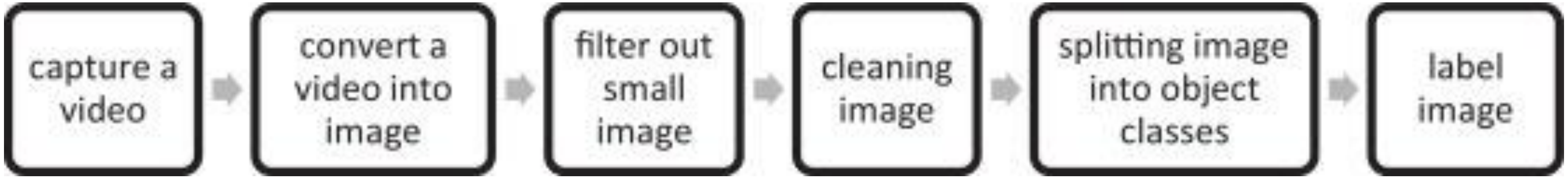
Amazon API Gateway

Amazon Polly

AWS Amplify

Data Description

- The dataset contains 37,500 fully labelled digital images collected in several elderly home cares in Malaysia in 25 different indoor object categories (i.e., bed, sofa, table, etc.)
  - Sourced from Mendeley Data
  - 1500 .images per class (containing some duplicates from the same class randomly modified using augmentation—rotations following a simple geometric transformation)
  - ~3GB total size (all .jpg)
- Captured using iPhone XS Max main camera and digitalized per process below



capture a video

convert a video into image

filter out small image

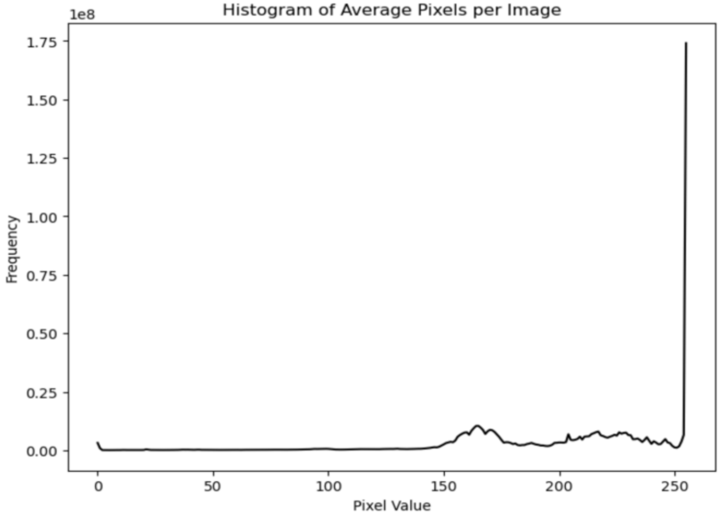
cleaning image

splitting image into object classes

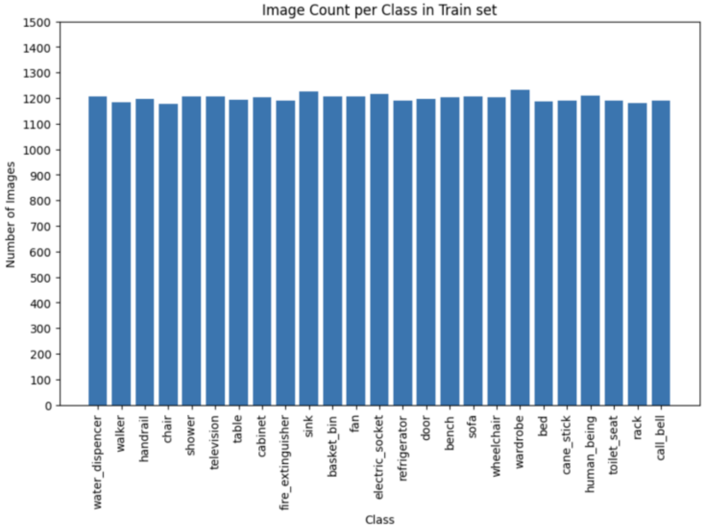
label image

Data Analysis / Data Exploration

Image Resolution

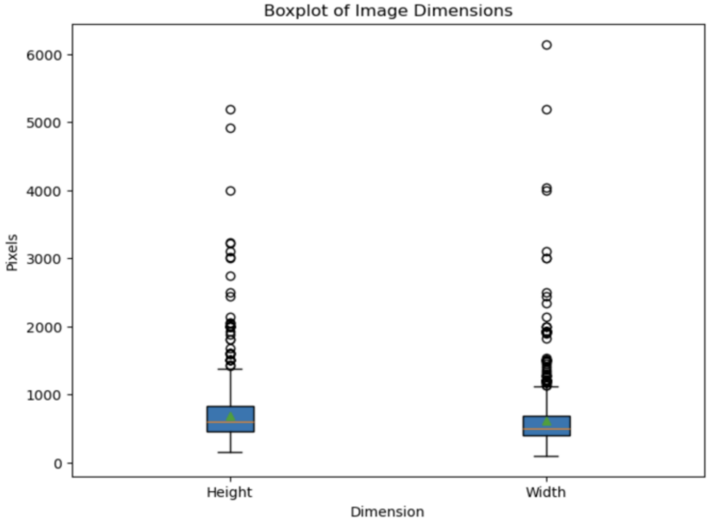


Frequency of images class/category



Images Displayed with their Labels from Dataset

Range of sizes per image



Data Preprocessing

- Load data using from S3 bucket using keras
  - Define parameters (batch size, image dimensions)
  - 80/10/10 train/validation/test split

- Additional data augmentation
  - Random horizontal flip
  - Random rotation
  - Random zoom

- Buffer prefetching and shuffle to optimize data retrieval performance

- preprocess_input() method from Keras specifying ResNet50 model

Test-driven development

- Troubleshoot upload image data to S3 bucket
  - AWS CLI vs AWS S3 UI

- AWS Sagemaker Notebook Instance
  - Memory issues during training
    - ml.m5.2xlarge vs. default ml.t3.medium

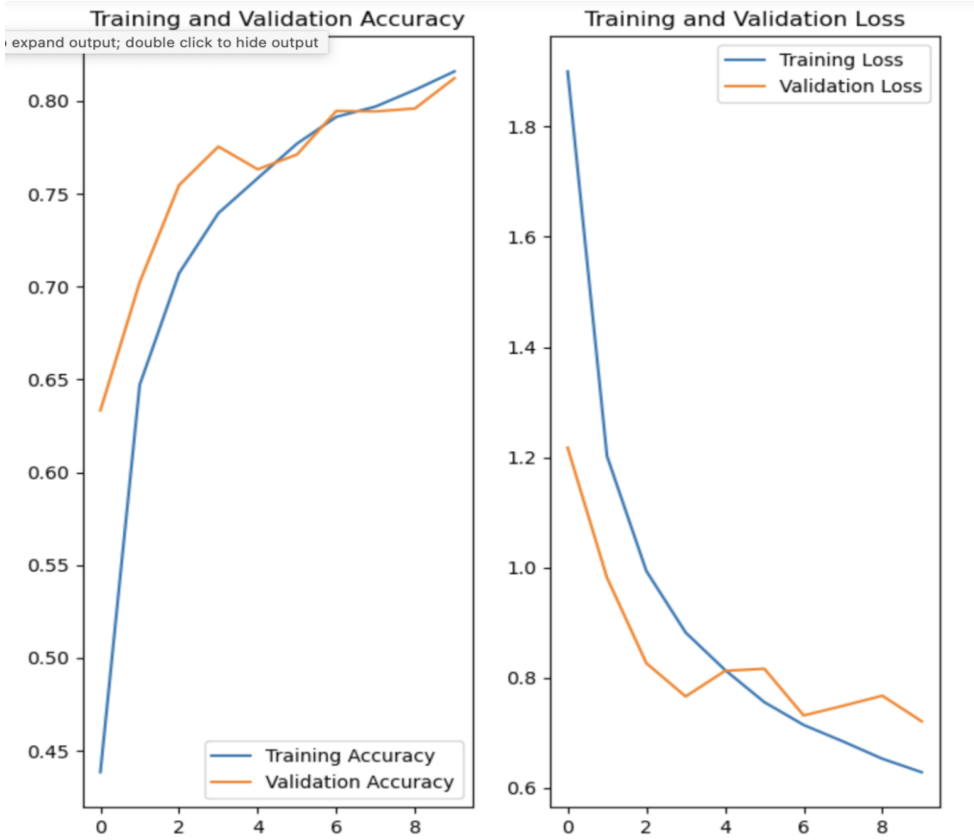
- Parameter selection (optimized for validation accuracy)
  - Optimizer Adam
  - Learning Rate 0.001
  - Epochs

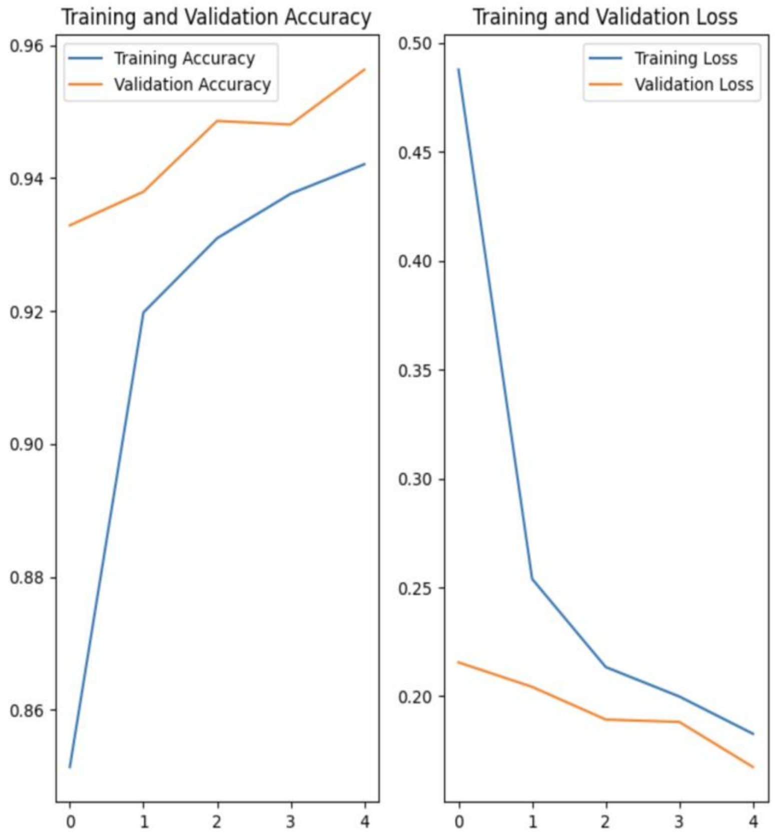
- Compare baseline CNN model and ResNet50 transfer learning model

Results

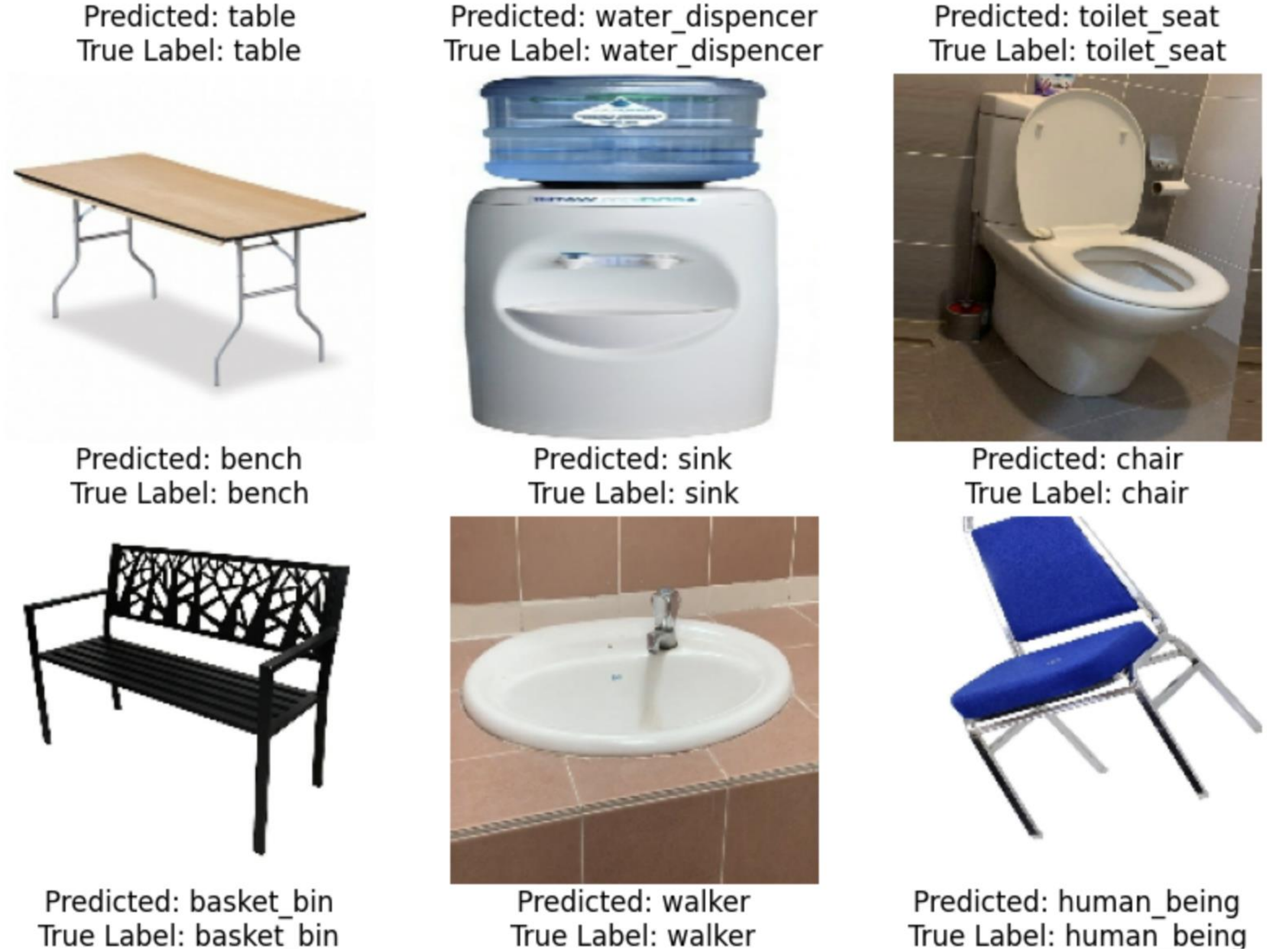
- **Baseline CNN Model:**
- 81% validation accuracy
- 81% test accuracy

- **ResNet-50 Transfer Learning Model:**
- 95% validation accuracy,
- 96% test accuracy





Visualize Test Results



Predicted: table
True Label: table

Predicted: water_dispencer
True Label: water_dispencer

Predicted: toilet_seat
True Label: toilet_seat

Predicted: bench
True Label: bench

Predicted: sink
True Label: sink

Predicted: chair
True Label: chair

Predicted: basket_bin
True Label: basket_bin

Predicted: walker
True Label: walker

Predicted: human_being
True Label: human_being

Conclusions & Next Steps

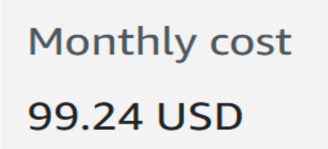
- **High Accuracy Metrics**
  - Accuracy metrics indicate strong model performance on test set
  - Developed models can support development of devices for elderlies with vision constraint and disabilities in healthcare
  - Can also benefit general public (esp. disabled personnel)

- **Potential Improvements**
  - Test add'l transfer learning options
  - Advanced testing / finetune hyperparameters
  - App development
    - TFLite => FireBaseML or Android Studio

Project's Budget Estimate

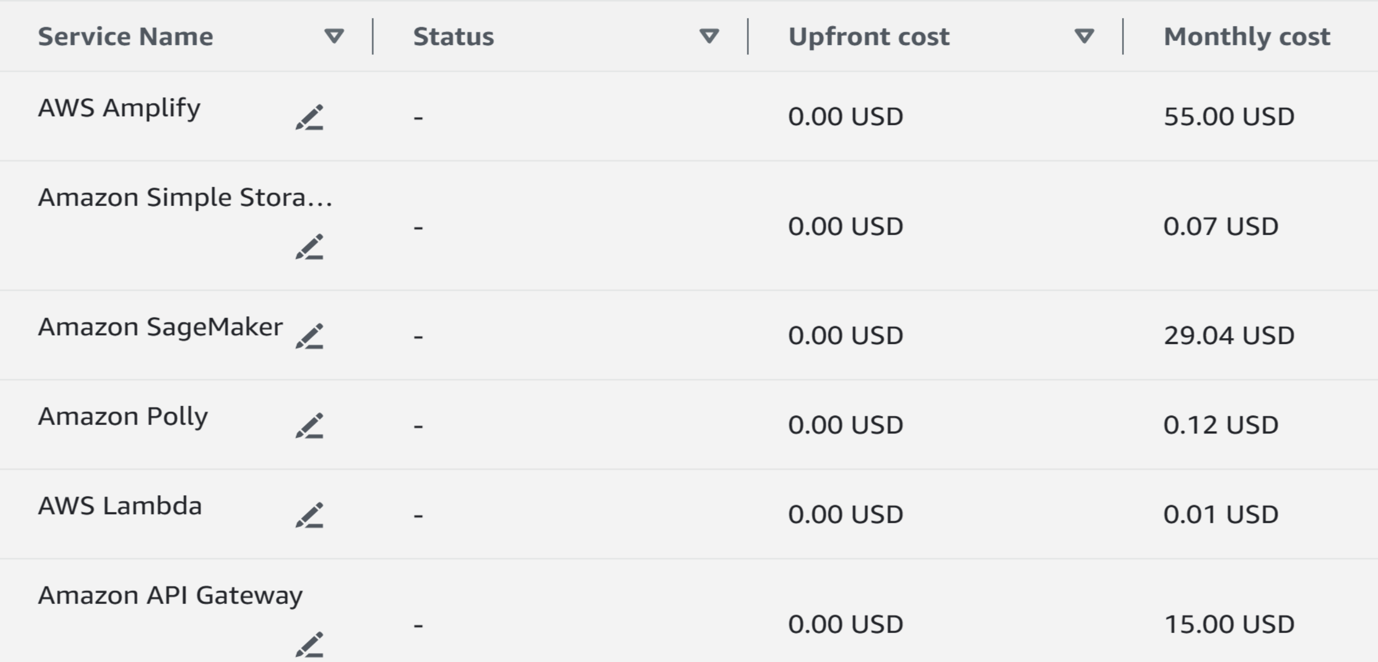
- **Amazon Amplify:**
  - Install and configure the AWS Amplify CLI.
  - Initialize a new Amplify project in your mobile app directory.
- **Amazon S3**:
  - S3 bucket to store images uploaded by users.
- **Amazon SageMaker**:
  - Set up Amazon Image Classification Model to train and analyze images.
- **Amazon Polly**:
  - Amazon Polly to converts text to speech.
- **AWS Lambda**:

  Lambda function:
    - Accepts an image from the S3 bucket trigger.
    - Sends the image to Amazon SageMaker for object detection.
    - Converts the object name to speech using Amazon Polly.
    - Sends the audio response back to the user.
- **Amazon API Gateway**:
  - Create a RESTful API using Amazon API Gateway to trigger the Lambda function.



Monthly cost

99.24 USD



Service Name	Status	Upfront cost	Monthly cost
AWS Amplify	-	0.00 USD	55.00 USD
Amazon Simple Stora...	-	0.00 USD	0.07 USD
Amazon SageMaker	-	0.00 USD	29.04 USD
Amazon Polly	-	0.00 USD	0.12 USD
AWS Lambda	-	0.00 USD	0.01 USD
Amazon API Gateway	-	0.00 USD	15.00 USD

Resources

- https://about.fb.com/news/2023/09/new-ray-ban-meta-smart-glasses/
- https://www.tensorflow.org/tutorials/images/classification
- https://aws.amazon.com/getting-started/hands-on/build-android-app-amplify/
- https://data.mendeley.com/datasets/fpctx3svzd/1