Intel Image Classification

Erika Pham, Katherine Sejas, Ei Tanaka

Content

1. **Scope of the Project**

2. **Image Classification Model**

- Data Preprocessing

- ResNET50 Implementation

3. Project Architecture

- Architecture

- Implementation

Scope of the Project

- Using the Intel Image Classification (https://www.kaggle.com/datasets/puneet6060/intel-image-classification) dataset.
- Originally published on https://datahack.analyticsvidhya.com by Intel for an Image classification challenge; retrieved on Kaggle.
- Dataset has approx. 25,000 images with six categories: buildings, forests, glaciers, mountains, sea, and streets.
- Goal:

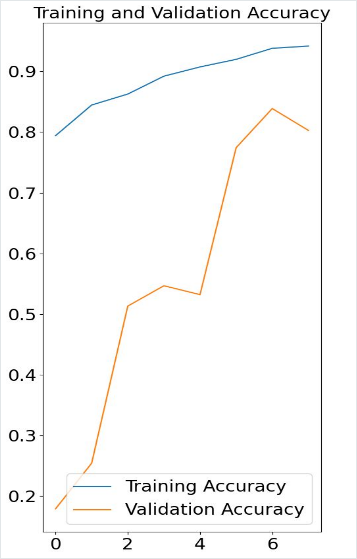
1. Implement machine learning techniques to build a model to accurately classify images.
2. Utilizing Google Colab and AWS to implement the architecture for front and back-end processes.

Image Classification Model – Data Preprocessing

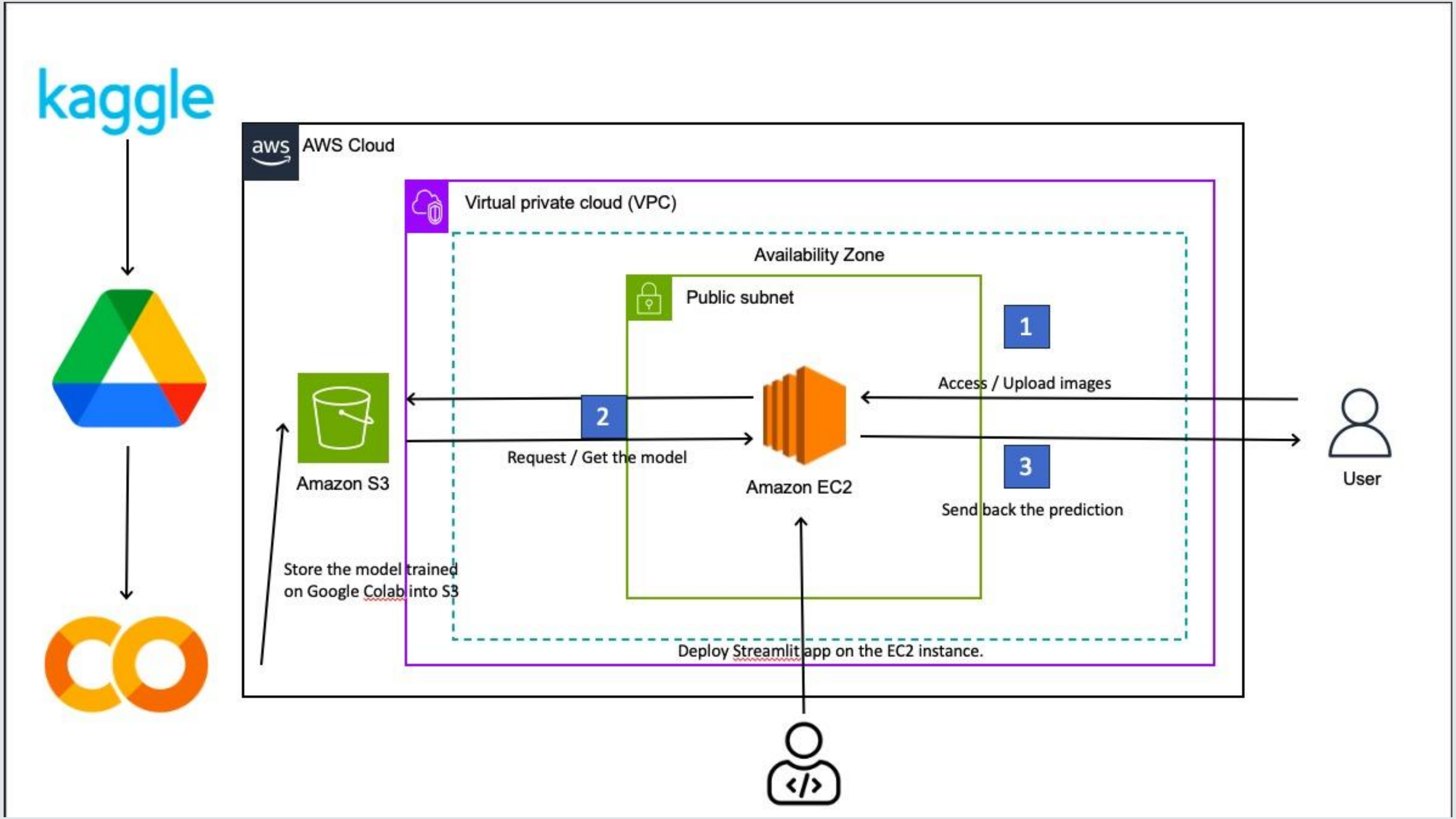
- Data consists ~14k images in the training set, 3k in the test set, and 7k in the prediction set.
- Input image size is 150 x 150.
- Using tensorflow’s ImageDataGenerator to generate image data that allows for real-time image augmentation – which marginally alters the image each time they’re read, feeding the model more data without needing to increase data size.

Image Classification Model – ResNet-50

- Using Keras' s ResNet-50, a variant of the ResNet (Residual Network) architecture, which is widely used in the field of deep learning, particularly in computer vision. It is a convolutional neural network.
- Pre-loaded the weights from training done on ImageNet (a large image dataset).
- We replaced the final classification layer of the original ResNet-50 model with a global average pooling layer followed by a dense layer with 6 classes and utilized softmax activation.
- Compiled with Adam optimizer and categorical cross entropy loss.
- Each epoch during training took about 6 minutes to complete.
- A validation accuracy of 80.28% was achieved at epoch 8.



Project Architecture



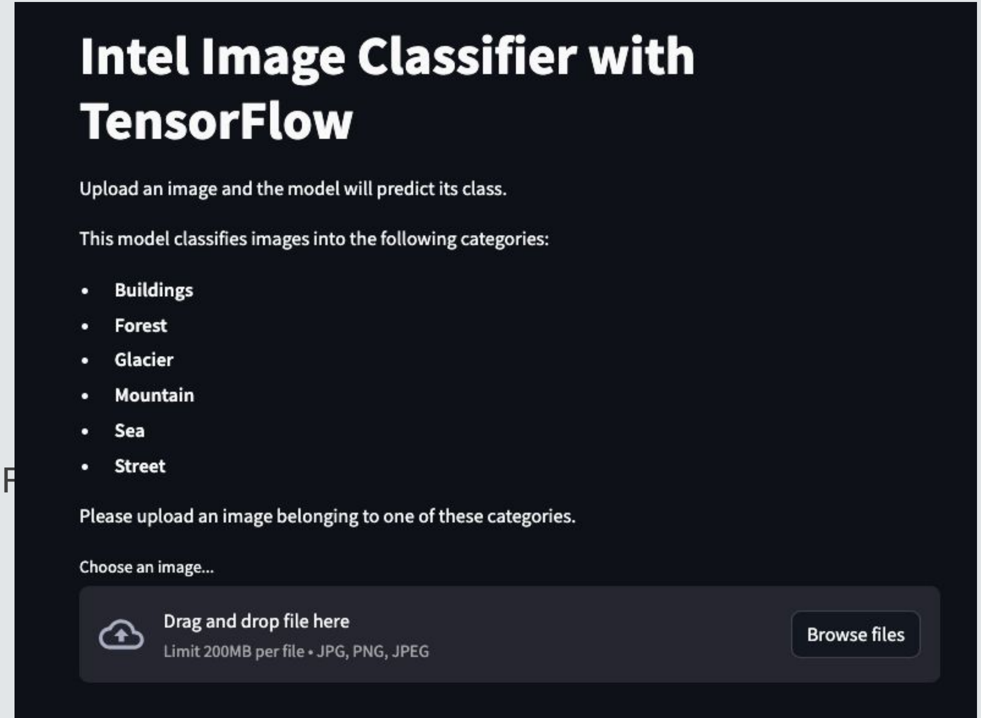
Implementation of application

<Python>

- Web Application: **Stremlit**
- Access to the S3 bucket: **boto3**
- Image Preprocessing: **PIL**

<Implementation>

- Activate Streamlit on EC2 instance
- Users can access this app through an external UF ,which Streamlit generates. (port: 8501)



Intel Image Classifier with TensorFlow

Upload an image and the model will predict its class.

This model classifies images into the following categories:

- **Buildings**
- **Forest**
- **Glacier**
- **Mountain**
- **Sea**
- **Street**

Please upload an image belonging to one of these categories.

Choose an image...

Thank you