Geological Similarity Pipeline

Pipeline Description

I’m using Kubeflow on a Kubernetes cluster to orchestrate my pipeline

The Pipeline consists of three components:

* Unzip-and-upload-images
* Train-model
* Deploy app

Graphical user interface, text, application

Description automatically generated

Unzip-and-upload-images:

The unzip-and-upload-images takes as input the location of the zipped folder in GCP and asks for a target bucket where the extracted images will be stored. This step also divides our images into a Train Set and Test set. This component then passes over the bucker name where the images were extracted. The code for extracting the images can be found at <https://github.com/varunvohra94/geological_similarity/blob/main/pipelines/zip-to-gcs/unzip_to_gcs.py>

Train-Model:

The train-model component takes as input the output of the last component (unzip-and-upload-images) which is the bucket where the train and test sets have been stored. The train-model component then submits a training job to GCP AI-Platform notebooks with the corresponding train and test sets. After training, the model and the embeddings for all the images are stored in the same bucket where the train and test images reside. The output of this component is the bucket where the model and the embeddings files are stored.

Deploy-App:

The deploy-app component takes as input the output of the train-model component (train-model) which is the bucket where the model and embeddings are stored. The deploy-app then builds an app on GCP App Engine

Prediction Results:

The prediction results are available at

<https://opportune-baton-267215.uc.r.appspot.com/>

This URL requires you to upload an image and select a value for K. Once done, the app should show you K most similar images to model.

CI/CD:

I’m using GCP Cloud Build to enable CI/CD in my pipeline. I have linked my GitHub repository to cloud build and also have a github trigger in my repository. I also have cloudbuild.yaml files for all the containers in my application which automatically builds and pushes the containers to GCP container registry to always use the code from the latest Push to the ‘main’ branch.