



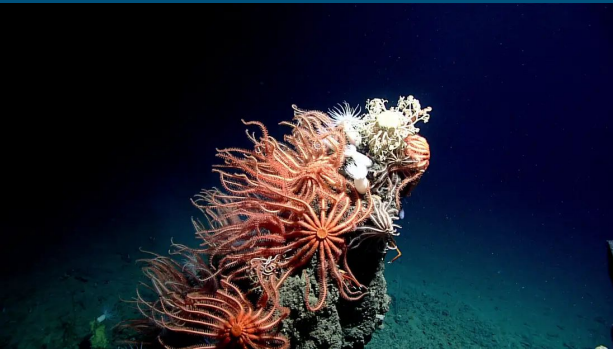
Deep Sea Image Classification

Cloud Computing

Jennifer Crosby

Benjamin Lee

Polly McKim



Overall Project



NOAA

- Acquired deep sea research expedition images from NOAA
- Built and deployed a machine learning model to classify images as interesting or not interesting based on creature presence
- Used AWS architecture to greatly increase speed and scalability of the model
- Delivered a model of close to 70% accuracy
- Used a CNN (convolutional neural network) algorithm as our model



Sources Used

- NOAA repository of deep sea images
- CVision AI (<http://cvisionai.com/>) houses NOAA images on an online server
- 1500 images + additional created through image augmentation



Features Implemented



Python libraries to install:

Pandas, numpy , matplotlib, sklearn, pytorch, pytorchvision,
warnings, PIL, image_folder, random

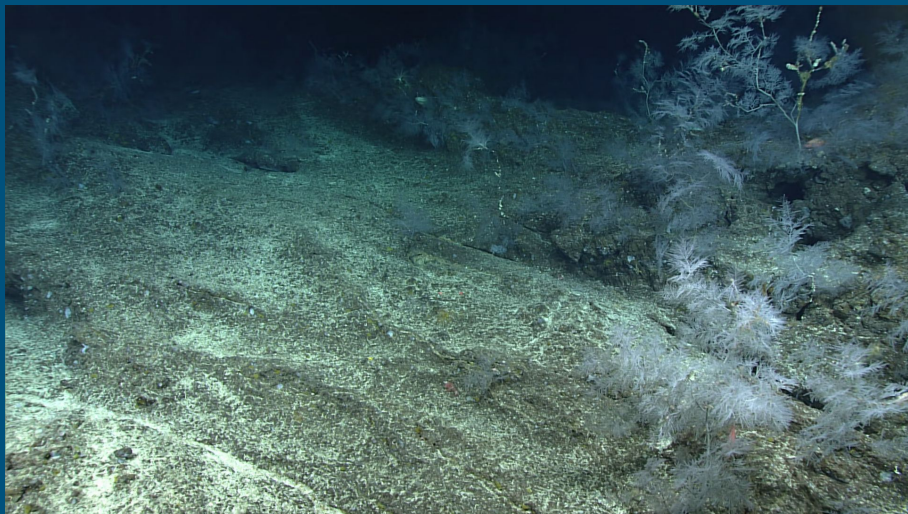
To use in model:

CNN, Adam optimizer, back propagation

The PyTorch logo, which consists of the word "PYTORCH" in a bold, black, sans-serif font. The letter "O" is replaced by a stylized orange flame icon with a small purple dot above it.

Examples of Inputs

Not Interesting



Interesting



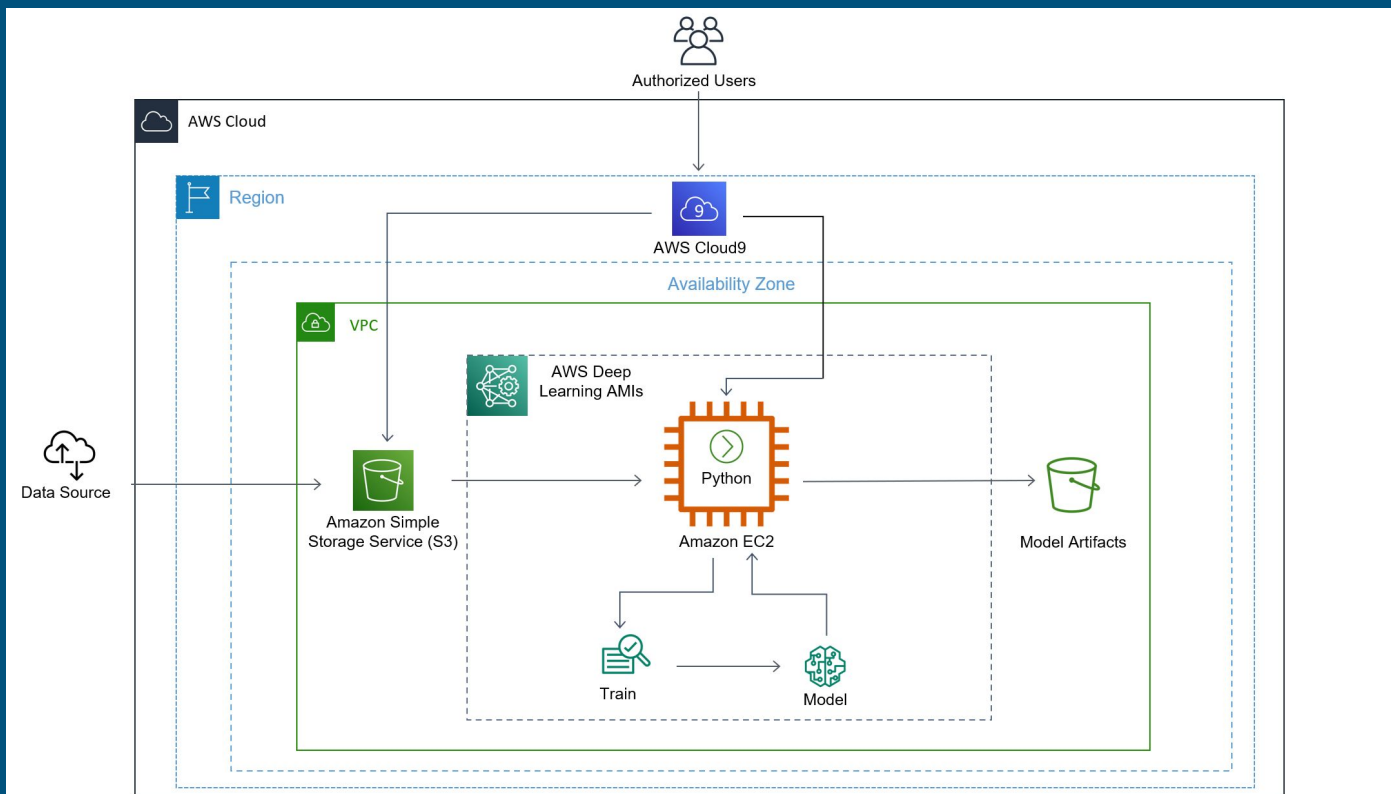
AWS Architecture Used for Project



- EC2 (AMI Name Deep Learning Base AMI (Ubuntu 18.04) Version 31.0);
Type: g3.4xlarge
- VPC
- AWS Cloud9- Cloud IDE

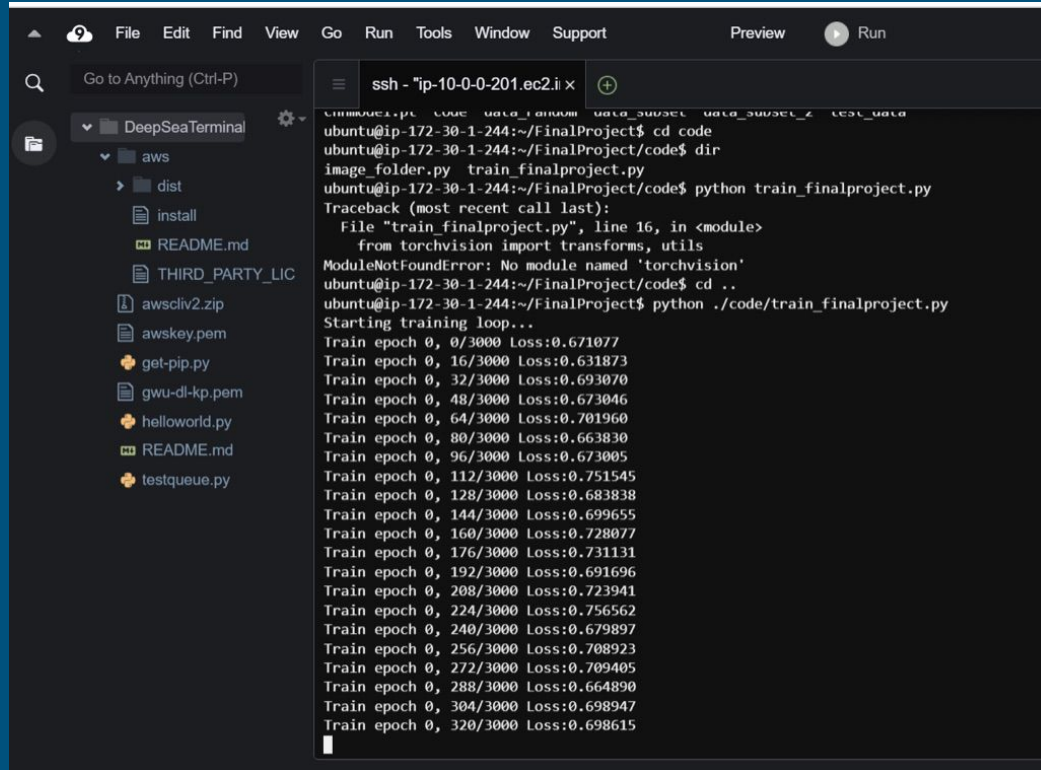


Architecture for Deep Learning EC2 Deployment



Cloud Architecture Demo

Python Running the Model on the Deep Learning EC2



```
File Edit Find View Go Run Tools Window Support Preview Run
Go to Anything (Ctrl-P)
DeepSeaTerminal
aws
dist
install
README.md
THIRD_PARTY_LIC
awscliv2.zip
awskey.pem
get-pip.py
gwu-dl-kp.pem
helloworld.py
README.md
testqueue.py
ssh - "ip-10-0-0-201.ec2.i x
ubuntu@ip-172-30-1-244:~/FinalProject$ cd code
ubuntu@ip-172-30-1-244:~/FinalProject/code$ dir
image_folder.py  train_finalproject.py
ubuntu@ip-172-30-1-244:~/FinalProject/code$ python train_finalproject.py
Traceback (most recent call last):
  File "train_finalproject.py", line 16, in <module>
    from torchvision import transforms, utils
ModuleNotFoundError: No module named 'torchvision'
ubuntu@ip-172-30-1-244:~/FinalProject/code$ cd ..
ubuntu@ip-172-30-1-244:~/FinalProject$ python ./code/train_finalproject.py
Starting training loop...
Train epoch 0, 0/3000 Loss:0.671077
Train epoch 0, 16/3000 Loss:0.631873
Train epoch 0, 32/3000 Loss:0.693070
Train epoch 0, 48/3000 Loss:0.673046
Train epoch 0, 64/3000 Loss:0.701960
Train epoch 0, 80/3000 Loss:0.663830
Train epoch 0, 96/3000 Loss:0.673005
Train epoch 0, 112/3000 Loss:0.751545
Train epoch 0, 128/3000 Loss:0.683838
Train epoch 0, 144/3000 Loss:0.699655
Train epoch 0, 160/3000 Loss:0.728077
Train epoch 0, 176/3000 Loss:0.731131
Train epoch 0, 192/3000 Loss:0.691696
Train epoch 0, 208/3000 Loss:0.723941
Train epoch 0, 224/3000 Loss:0.756562
Train epoch 0, 240/3000 Loss:0.679897
Train epoch 0, 256/3000 Loss:0.708923
Train epoch 0, 272/3000 Loss:0.709405
Train epoch 0, 288/3000 Loss:0.664890
Train epoch 0, 304/3000 Loss:0.698947
Train epoch 0, 320/3000 Loss:0.698615
```

68%

Overall Accuracy Achieved

Findings & Cautionary Tales

- Ran much faster than on a typical personal computer
- Auto Scalability or using more GPUs would likely have improved the run time even more but would have added to the cost of running the model
- Understanding of EC2 Ubuntu choices for deep learning is important (ie make sure the family you select has the capacity and features you need)
- GPUs do not exist in all availability zones or on every option of deep learning Ubuntu (Don't select a T when you really need a G or a P!)
- The future the process would be simplified by figuring out feed the data directly from the S3 to the running python program on the EC2

Questions?

