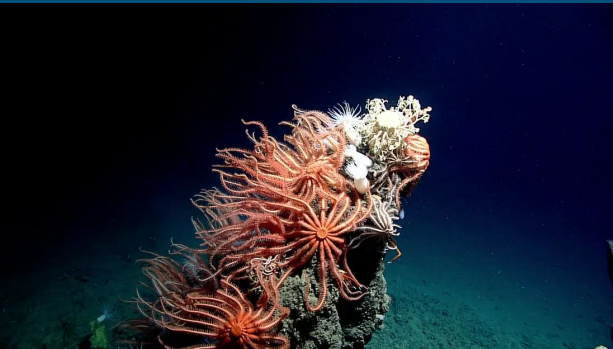





Deep Sea Image Classification

Cloud Computing

Jennifer Crosby
Benjamin Lee
Polly McKim



Problem Statement

Researchers currently spend thousands of hours sifting through countless photos to determine which ones have items of interest for their research. Can deep learning be used to solve this problem?

Can we find cute photos like this?



While filtering out photos like this?



Project Overview



- Acquired deep sea research expedition images from NOAA
- Manually classified images as interesting or not based on creature presence
- Built and deployed a machine learning model to learn to classify images based on their content labels
- Used AWS architecture to greatly increase speed and scalability of the model
- Delivered a model of over 75% 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



Libraries & Features Implemented

Python libraries to install:

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

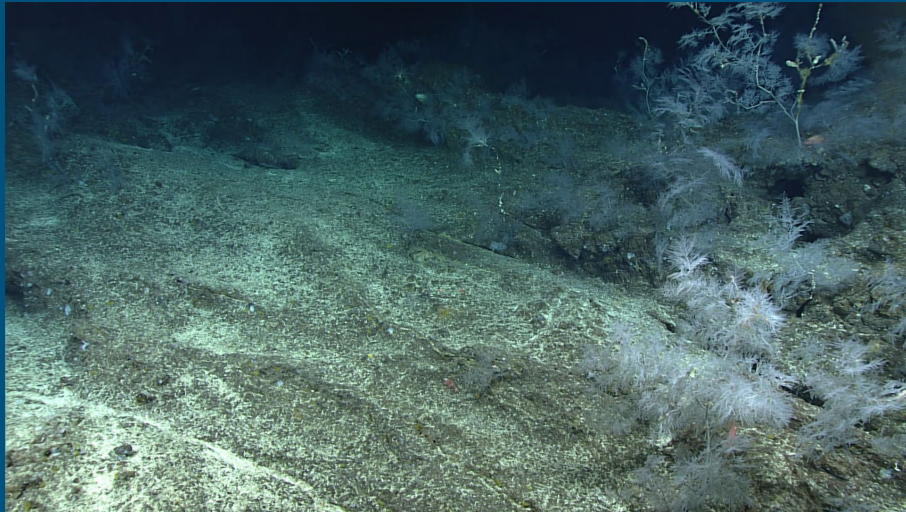
Features used in modeling:

CNN, Adam optimizer, back propagation

The PyTorch logo is displayed in a white rectangular box. It features 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



Target file

18	56321614_EX1708_VID_20170926T212000Z_ROVHD.mp4_1378.png	interesting
19	56321614_EX1708_VID_20170926T212000Z_ROVHD.mp4_2247.png	interesting
20	56321614_EX1708_VID_20170926T212000Z_ROVHD.mp4_2277.png	interesting
21	56321614_EX1708_VID_20170926T212000Z_ROVHD.mp4_3866.png	interesting
22	56321614_EX1708_VID_20170926T212000Z_ROVHD.mp4_4555.png	interesting
23	56321614_EX1708_VID_20170926T212000Z_ROVHD.mp4_4675.png	interesting
24	56321614_EX1708_VID_20170926T212000Z_ROVHD.mp4_4885.png	interesting
25	56321614_EX1708_VID_20170926T212000Z_ROVHD.mp4_5274.png	interesting
26	56321614_EX1708_VID_20170926T212000Z_ROVHD.mp4_5604.png	interesting
27	56321614_EX1708_VID_20170926T212000Z_ROVHD.mp4_6113.png	interesting
28	56321614_EX1708_VID_20170926T212000Z_ROVHD.mp4_6323.png	interesting
29	56321614_EX1708_VID_20170926T212000Z_ROVHD.mp4_7222.png	not_interesting
30	56321614_EX1708_VID_20170926T212000Z_ROVHD.mp4_7462.png	not_interesting
31	56321614_EX1708_VID_20170926T212000Z_ROVHD.mp4_7852.png	not_interesting
32	56321614_EX1708_VID_20170926T212000Z_ROVHD.mp4_7882.png	not_interesting
33	56321614_EX1708_VID_20170926T212000Z_ROVHD.mp4_7972.png	not_interesting
34	56321615_EX1708_VID_20170926T200000Z_ROVHD.mp4_539.png	interesting
35	56321615_EX1708_VID_20170926T200000Z_ROVHD.mp4_629.png	interesting

The 1500 images were manually classified

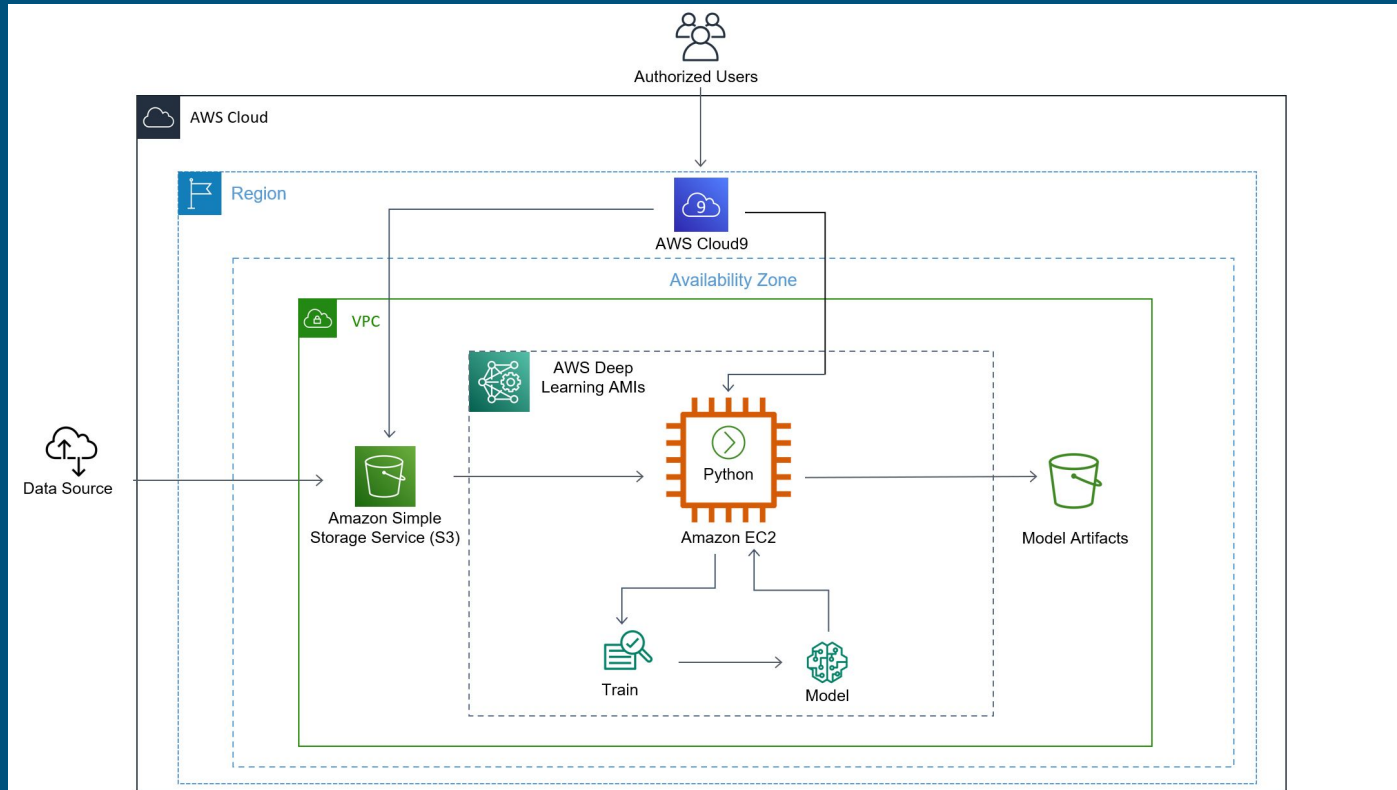
AWS Architecture Used for Project



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

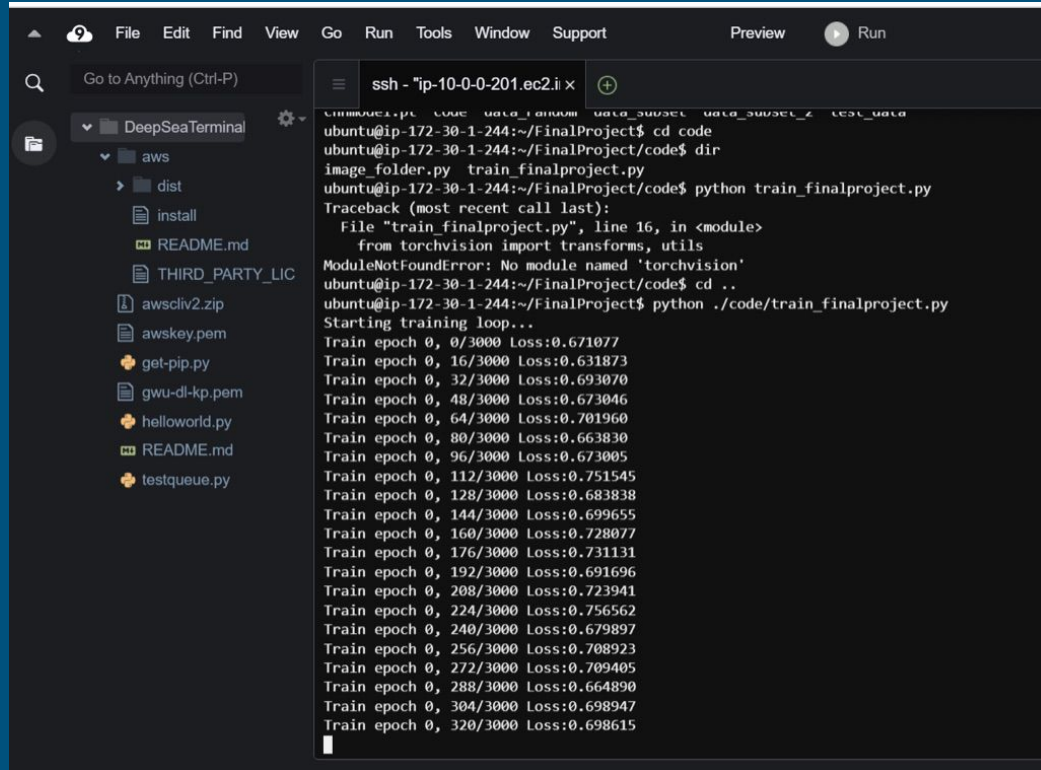


Architecture for Deep Learning EC2 Deployment



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
```

76%

Overall Accuracy Achieved

```
Validation Loss 0.67914  
Validation Accuracy: 76.0  
ubuntu@ip-172-30-1-244:~/FinalProject$
```

EC2 Deep Learning Findings & Cautionary Tales

- G8, Ran much faster than on a typical personal computer. Ideal P3.
- 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. Need to make sure have limits available to implement.
- 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!).
- In the future, the process would be simplified by figuring out how to feed the data directly from the S3 to the running python program on the EC2

Sagemaker



Amazon SageMaker

Sagemaker Marketplace


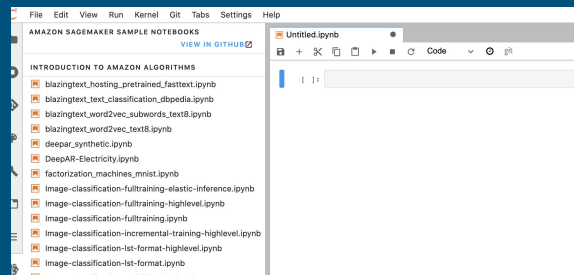


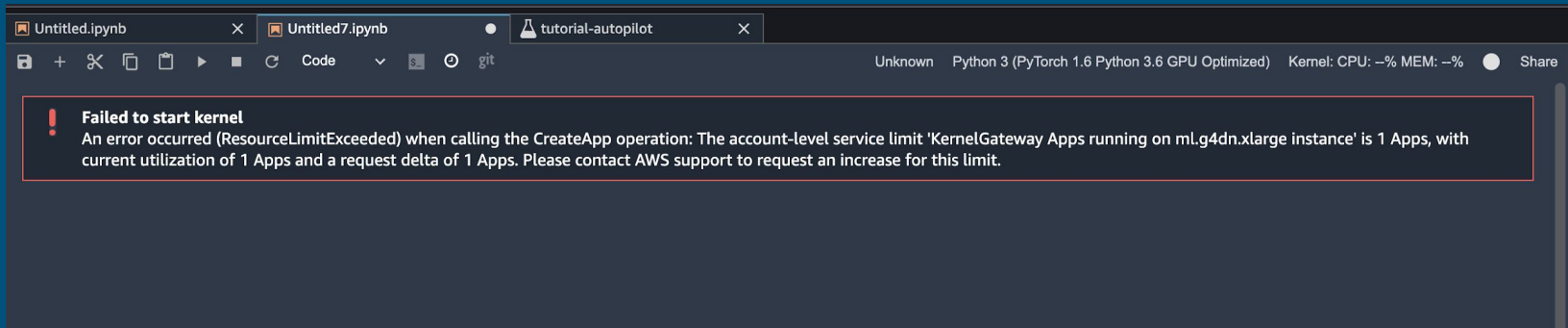
Image Recognition (Trainable Algorithm) [↗](#)
By [Sensifai](#) [↗](#) | Ver v1
Algorithm - Fulfilled on Amazon SageMaker
[Free Trial](#)

Sensifai offers automatic image recognition and tagging. For example, our basic software recognizes thousands and thousands of objects/scenes and concepts in images such as flowers, landmarks, objects and animals. In Sagemaker platform, you can easily fine-tune this software to recognize a new set...

Prefab Notebooks



Instance Limit Issues



Sagemaker Findings & Cautionary Tales

- Sagemaker notebooks/instances, like much of AWS, have predetermined blocks on certain kinds of notebooks so people don't inadvertently run up a large bill
- AWS is very slow to respond to requests to raise these limits
- While Sagemaker worked well for smaller tutorial type models that we did to learn about the platform, we were unable to get adequate access to run a deep learning model within the sagemaker ecosystem



Amazon SageMaker

Questions?

