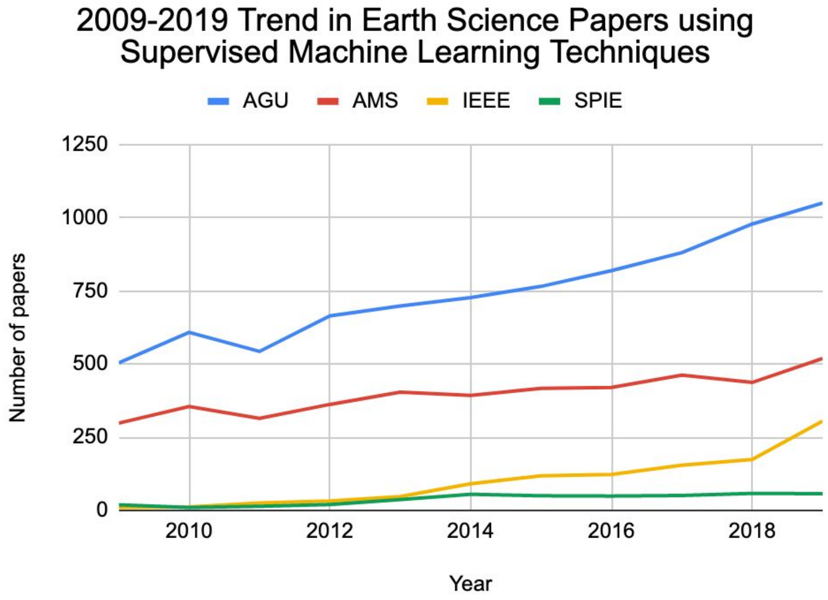
Machine Learning Pipeline for Earth Science Using Sagemaker

Iksha Gurung [1], Muthukumaran Ramasubramanian [1], Drew Bollinger [2], Manil Maskey [3], Shubhankar Ghalot [1], Rahul Ramachandran [3]

1. University of Alabama in Huntsville
2. Development Seed
3. NASA

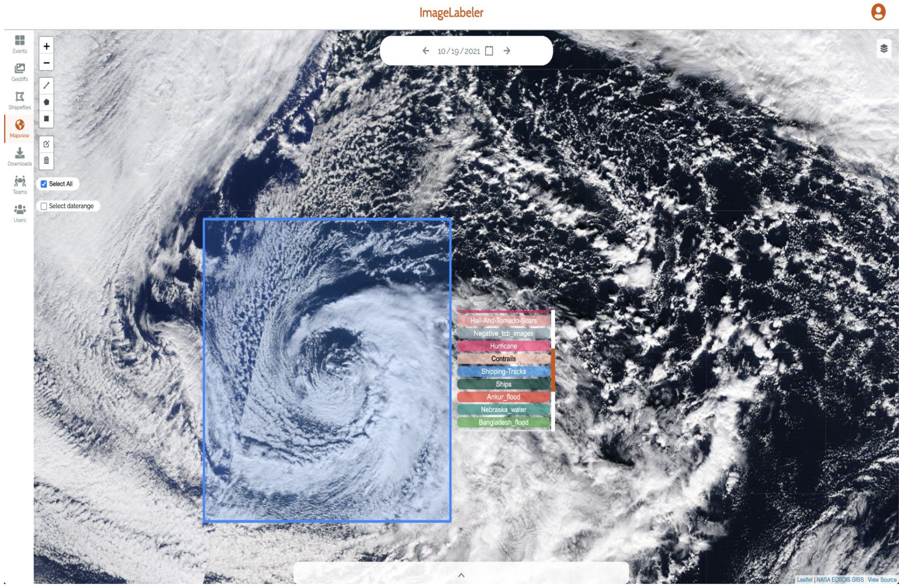
Introduction

- Machine learning (ML) is gaining popularity in the Earth science domain
- Higher the amount of quality data, the better the model.
- CPU training of such ML models is slow; GPU is used for training.
- Maintaining GPU servers is an additional responsibility.
- Multiple iterations of experiments needed before a better performing model is trained.
- Dataset creation, versioning of datasets, models, and experiments is hard.



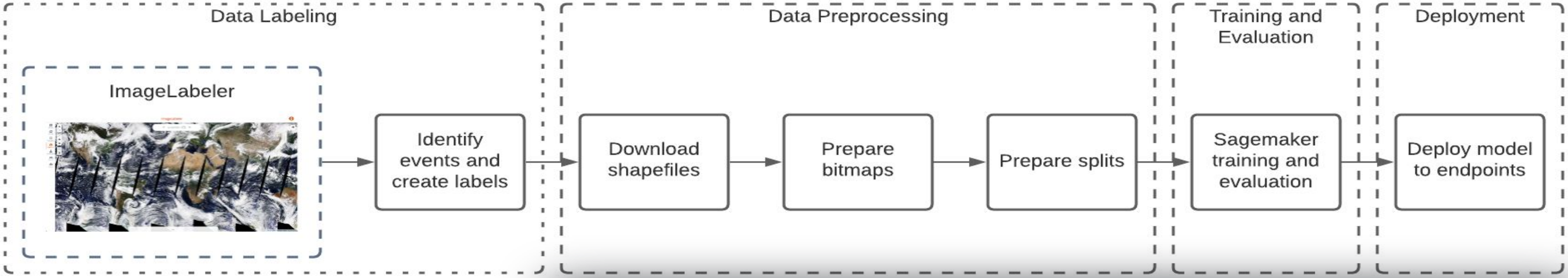
ImageLabeler

- Create labeled datasets
- Subject matter experts (SMEs) validate labels
- Export labeled data in ML-ready format



ImageLabeler
https://impact.earthdata.nasa.gov/labeler

Pipeline



- Data labeling done in ImageLabeler
- Data preparation, model build, train, and deployment handled by Sagemaker

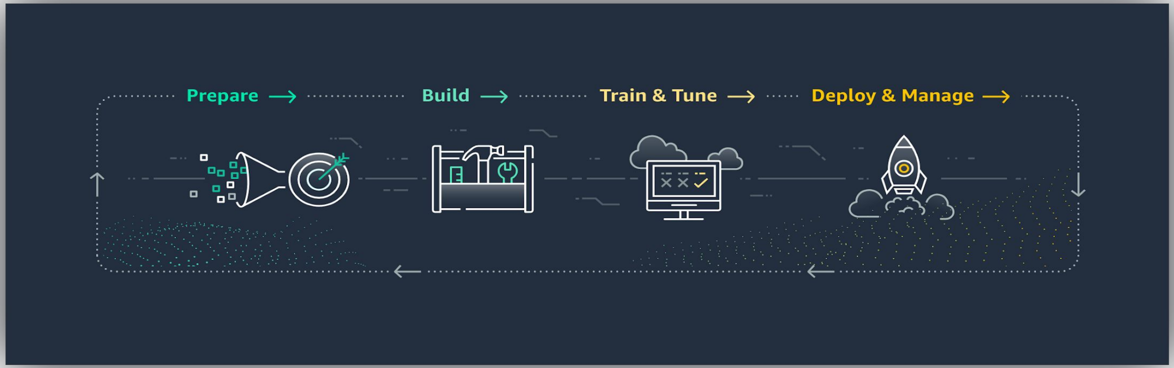
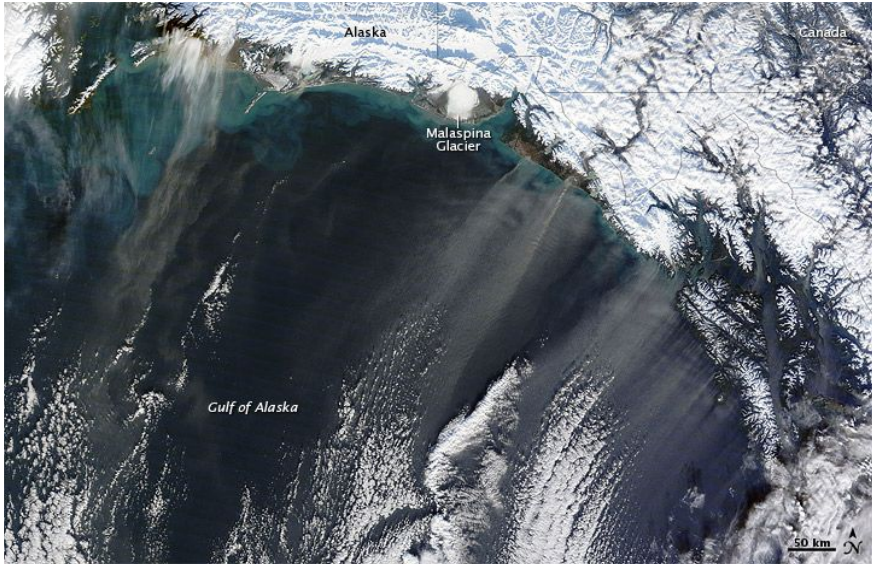
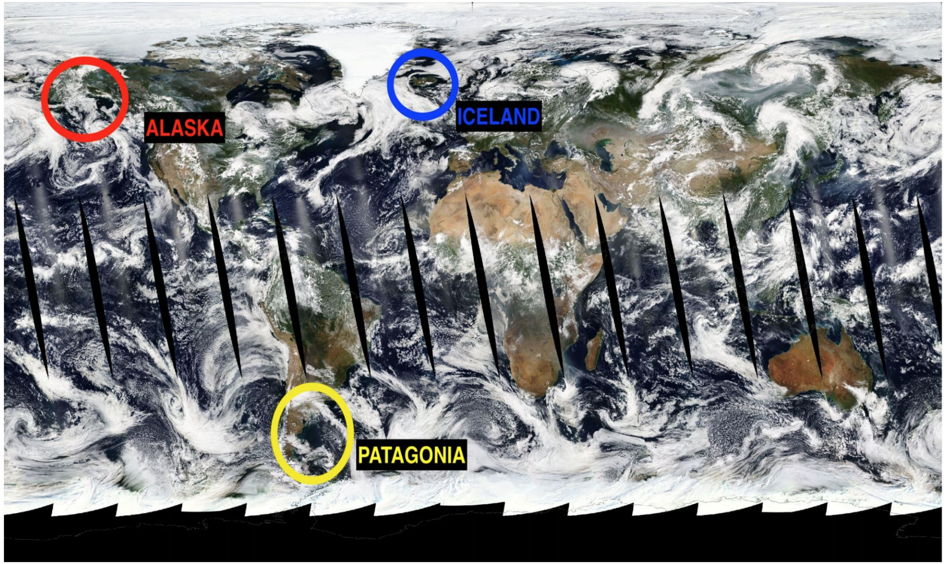


Fig: Sagemaker pipeline
Credit: https://aws.amazon.com/sagemaker/

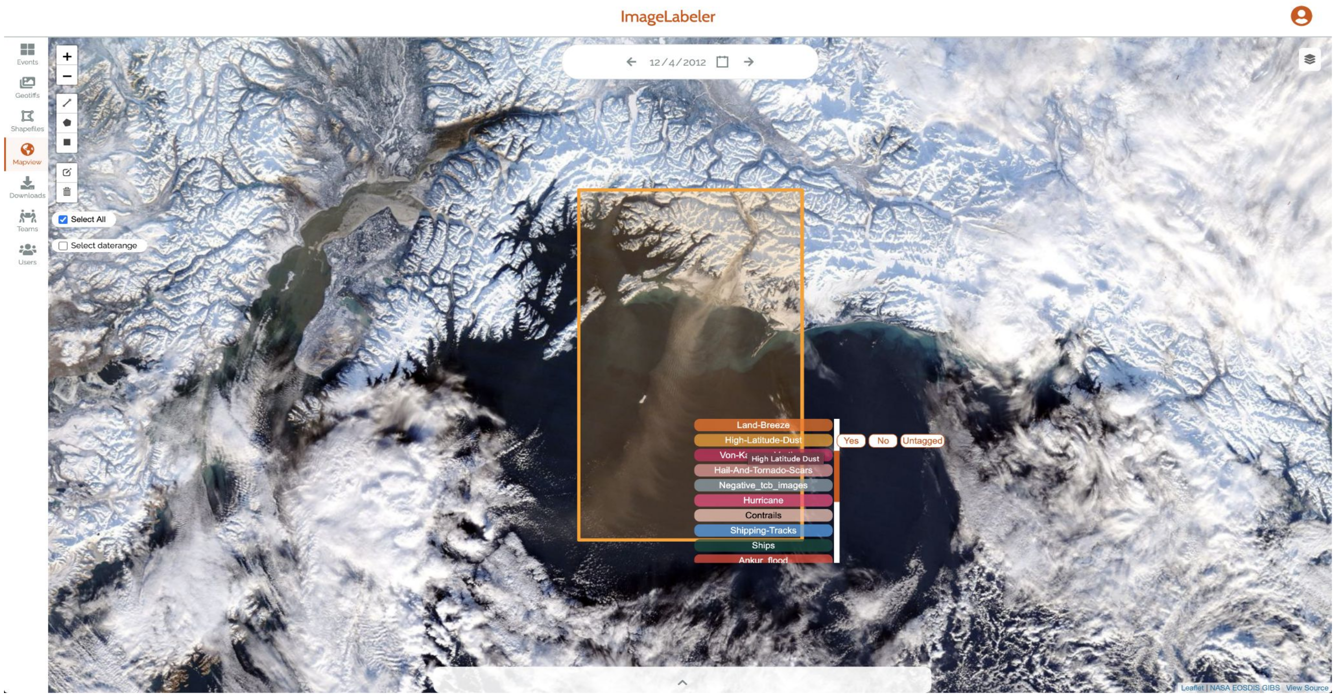
Usecase: High Latitude Dust

- Dust events confined to latitudes > 40N and < 40S
- Sources of polar atmospheric aerosol concentrations and surface deposition



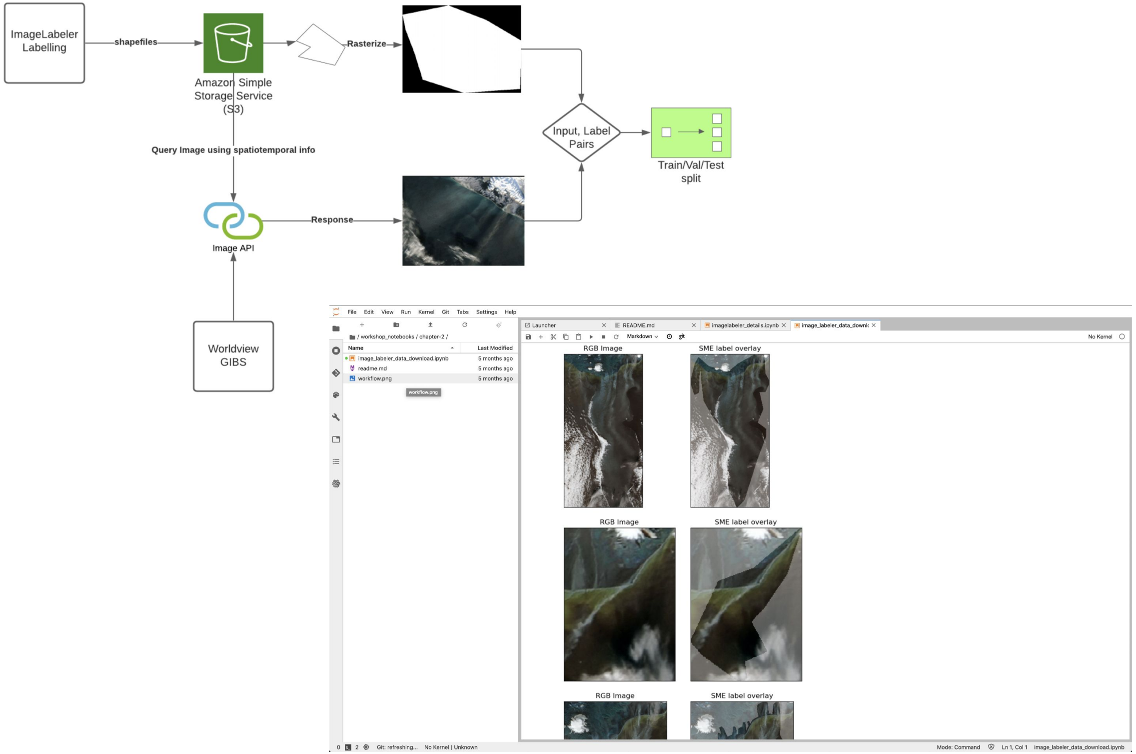


Data Labeling



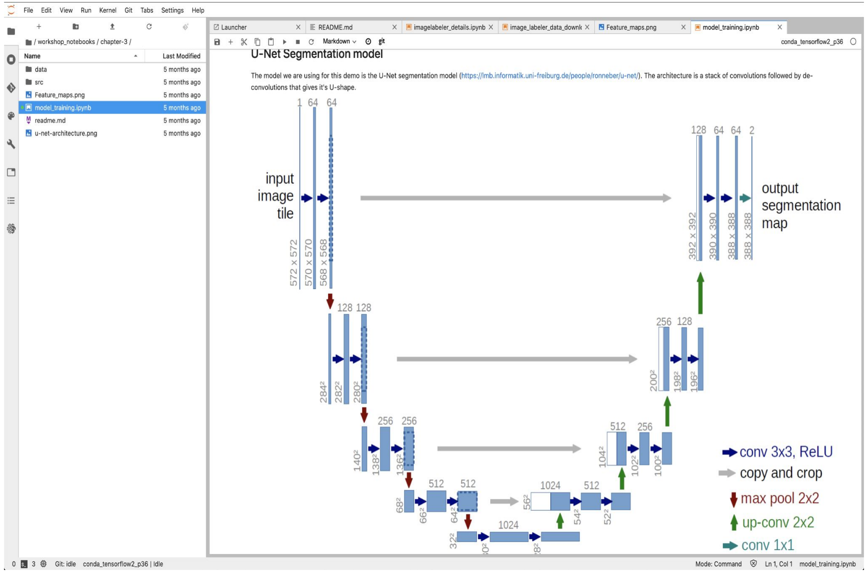
- Use ImageLabeler to identify and label HLD events in ImageLabeler
- SMEs review the labels

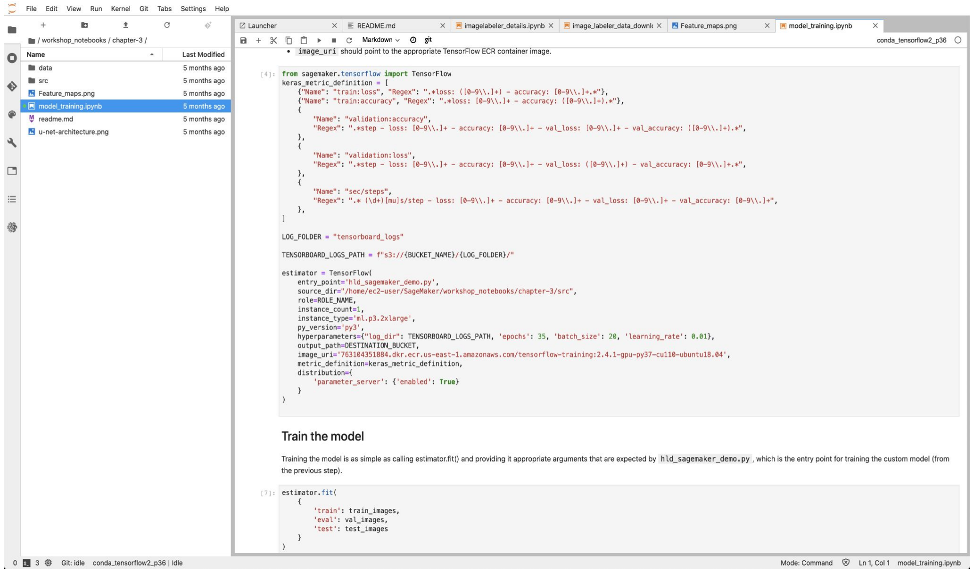
Data Preprocessing



- Download shapefiles from S3 bucket into Sagemaker notebook instance
- Convert shapefiles into bitmaps
- Prepare image and label pairs
- Prepare train/val/test splits

Training and Evaluation





- Train a custom U-Net model for segmentation
- Sagemaker package launches a training instance (just for the duration of training)
- Trained model is saved in a S3 bucket.

Deployment

Deploy the trained model from within the SageMaker instance

```
[6]: # Refer to Chapter-3 checkpoints or select from your S3 bucket.
model_location = f"{BUCKET_NAME}/tensorflow-training-2021-06-03-15-56-11-370/output/model.tar.gz"
framework_version = '2.4.1'

model = TensorFlowModel(
    framework_version=framework_version,
    role='notebookAccessRole',
    model_data=model_location
)
```

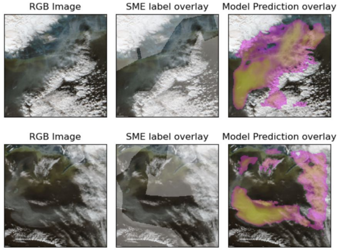
```
[7]: estimator = model.deploy(initial_instance_count=1, instance_type='ml.t2.large')
```

```
update_endpoint is a no-op in sagemaker>=2.
See: https://sagemaker.readthedocs.io/en/stable/v2.html for details.
---------------!
```

- Trained model can be loaded using Sagemaker.
- Model can be deployed using "model.deploy".

```
[9]: modis_batch, bmp_batch = get_test_data()
bmp_predict_batch = np.asarray(estimator.predict(modis_batch)['predictions'])
for j in range(len(modis_batch)):
    bmp_data = bmp_batch[j]
    f, ax = plt.subplots(1, 3, constrained_layout=True, dpi=100)
    ax[0].imshow(modis_batch[j].astype('uint8'))
    ax[0].set_title('RGB Image')
    ax[0].xaxis.set_ticks([])
    ax[0].yaxis.set_ticks([])
    ax[1].imshow(modis_batch[j].astype('uint8'))
    ax[1].xaxis.set_ticks([])
    ax[1].yaxis.set_ticks([])
    ax[1].set_title('SME label overlay')
    ax[2].imshow(modis_batch[j].astype('uint8'))
    ax[2].set_title('Model Prediction overlay')
    ax[2].xaxis.set_ticks([])
    ax[2].yaxis.set_ticks([])
    bmp_data = bmp_batch[j].astype('uint8')
    ax[1].imshow(ma.masked_where(bmp_batch[j] != 0, bmp_batch[j])[:,:,0],alpha=0.35,cmap='Purples')
    ax[2].imshow(ma.masked_where(bmp_predict_batch[j] < 0.5, bmp_predict_batch[j])[:,:,0],alpha=0.45,cmap='spring')
```

```
['../chapter-3/data/test/high-latitude-dust_2003-10-30_113.tiff', '../chapter-3/data/test/high-latitude-dust_2009-10-29_240.tiff', '../chapter-3/data/test/high-latitude-dust_2018-01-28_197.tiff', '../chapter-3/data/test/high-latitude-dust_2018-05-08_192.tiff', '../chapter-3/data/test/high-latitude-dust_2015-04-01_admin_117.tiff']
(256, 256, 3)
(256, 256, 3)
(256, 256, 3)
(256, 256, 3)
(256, 256, 3)
```



- Deployed model can then be used for inference.
- Deployed models can also be deployed to endpoints and accessed using URLs.

Conclusion

- ML is being adopted for scientific discoveries.
- Maintaining a GPU server is an overhead.
- Sagemaker takes care of operations to prepare a scalable, reproducible environment for experiments to run.

Resources

- High Latitude Dust: https://ams.confex.com/ams/2020Annual/meetingapp.cgi/Paper/369216
- Sagemaker examples: https://github.com/aws/amazon-sagemaker-examples
- Earth Science Example with HLD: https://github.com/NASA-IMPACT/workshop_notebooks
- ImageLabeler: https://impact.earthdata.nasa.gov/labeler/

Contact

Iksha Gurung
Email: ig0004@uah.edu