Earth Science Deep Learning: Applications and Lessons Learned

Manil Maskey, Rahul Ramachandran, J.J. Miller, Jia Zhang, and Iksha Gurung
IMPACT Project - Advanced Concepts Lead, NASA Marshall Space Flight Center
University of Alabama in Huntsville
Carnegie Mellon University

IGARSS 2018
July 24, 2018. Valencia, Spain

Overview

- Deep learning
- Earth science applications
- Lessons Learned
- Outlook and conclusion

Deep Learning

- A subfield of machine learning
- Algorithms inspired by function of the brain
- Scales with amount of training data
- Powerful tool without the need for feature engineering
- Suitable for Earth Science applications

Recent Deep Learning Successes

- Facebook
  - Translates about 2 billion user posts per day in more than 40 languages
  - Photo search and photo organization
- Microsoft
  - Speech-recognition products: Bing voice search, X-Box voice commands
  - Search rankings, photo search, translation systems
- Google:
  - Almost all services
- Medical Science
  - Diagnosis Language translation
- Playing strategy games
- Self driving cars

Deep Learning for Earth science at MSFC

- Phenomena identification
- Hurricane intensity (wind speed) estimation
- Severe storm (hailstorm) detection
- Transverse bands detection
- Entity extraction for knowledge graph creation
- Ephemeral water detection

Tropical Cyclone Intensity Estimation

- The Dvorak technique
  - Vernon Dvorak (1970s)
  - Satellite-based method
  - Cloud system measurements
  - Development patterns corresponds to T-number
- Deviation-angle variation technique (DAVT)
  - Piñeros et al. (2008)
  - Variance for quantification of cyclones
  - Calculates using center (eye) pixel
  - Directional gradient statistical analysis of the brightness of images

Issues

- Subjective/Uncertainty
- Lack of generalizability
- Inconsistency
- Complexity

15 UTC 10 Oct 17 NHC advisory on Tropical Storm Ophelia

"Dvorak intensity estimates range from T2.3/33 kt from UW-CIMSS to T3.0/45 kt from TAFB toT4.0/65 kt from SAB. For now, the initial intensity will remain at 45 kt, which is an average of the scatterometer winds and all of the other available intensity estimates."

Observation:
Two human experts at TAFB and SAB differed by 20 knots in their Dvorak analyses, and the automated version at the University of Wisconsin was 12 kt lower than either of them!

Can we objectively predict wind speed from images?

Data

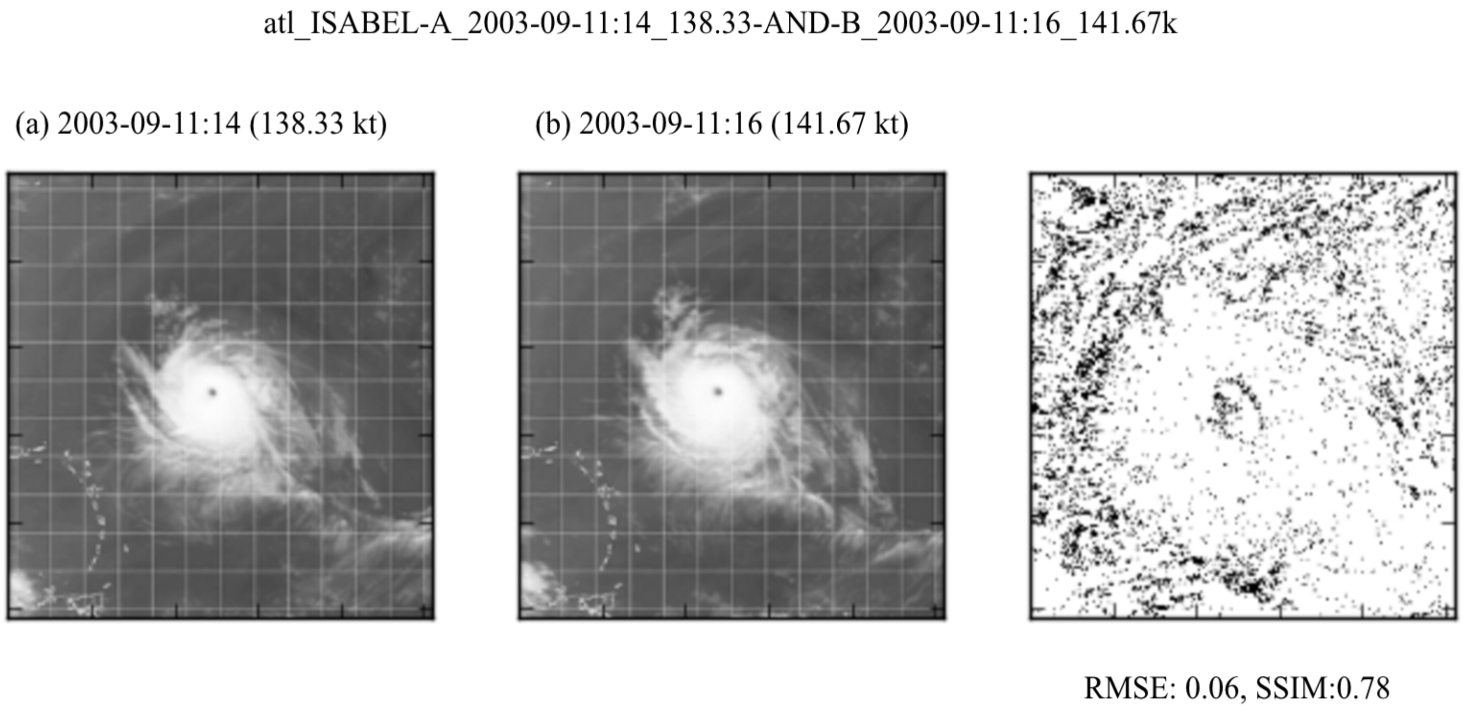
- Images
  - GOES-IR
  - From 2000 to 2017
  - East Pacific and Atlantic
- Cyclone data
  - National Hurricane Center (http://www.nhc.noaa.gov) (HURDAT and HURDAT2)
  - Hurricane Research Division (http://www.aoml.noaa.gov/hrd/hurdat/Data_Storm.html)
- Preprocessing
  - Subset GOES +/-5 deg. around eye
  - Nearest 1kt wind speed interval
  - Removed >70% missing data

Data Distribution

- Unbiased data splitting
  - Year 2000 – 2016 Training
    - 97152 images
  - Year 2017 Testing
    - 4840 images

Data augmentation

- Interpolate to increase even more
- 2 hours interpolated image differences



2 hour interpolated image differences

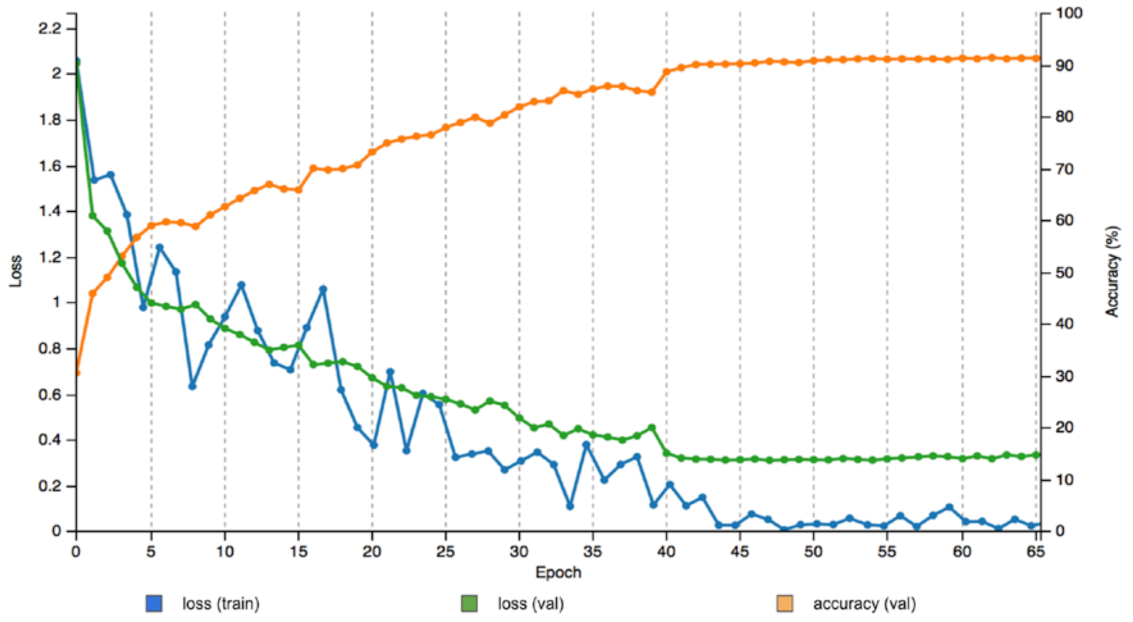
Training, test, and validation

- (Training + Validation) 70% - 30% (Test)
- (Training) 75% - 25% (Validation)

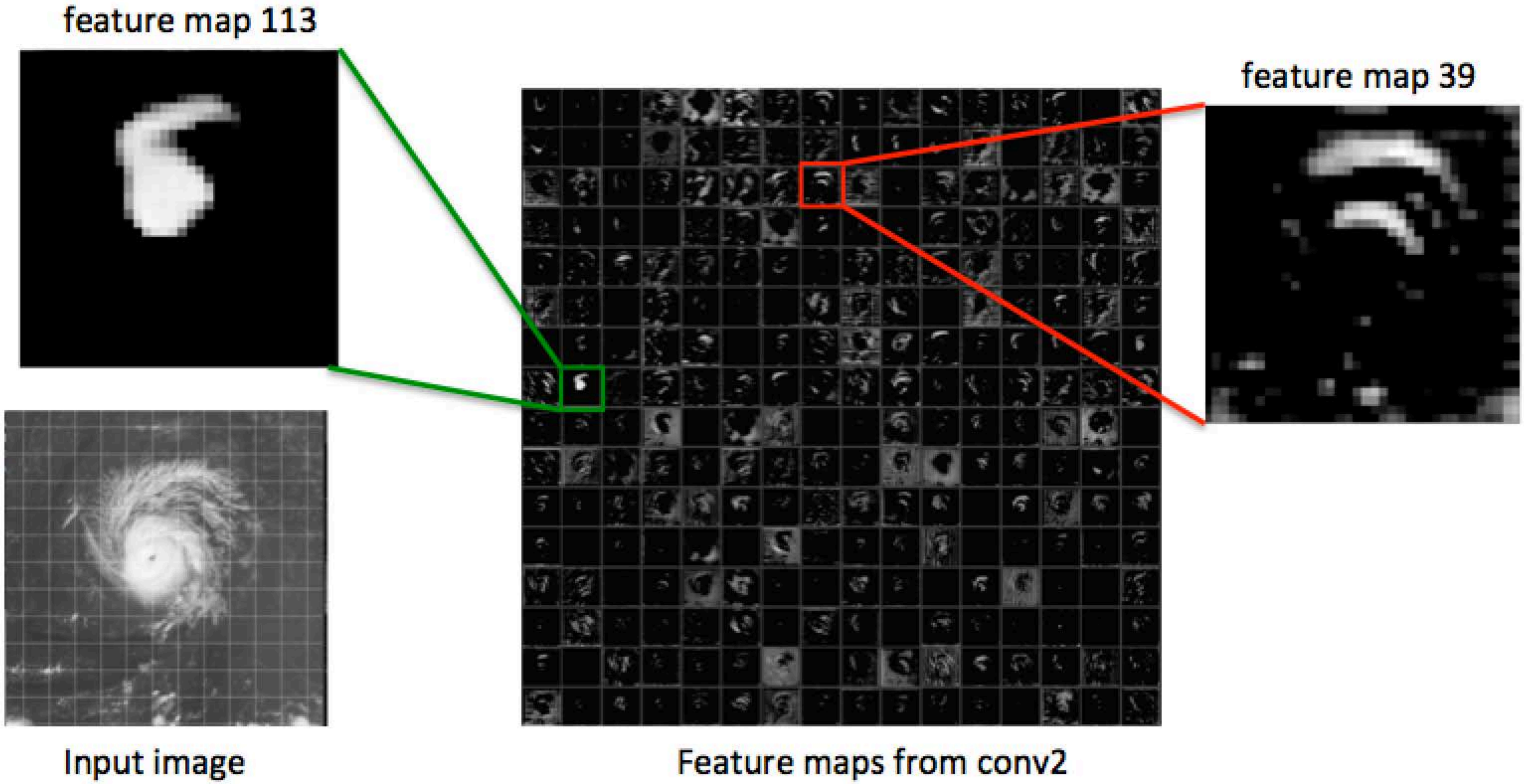
Hurricane Category	Train	Validation	Test	Total
H1	3314	1104	1816	6234
H2	1860	620	994	3474
H3	1848	616	992	3456
H4	1886	628	1032	3546
H5	603	201	306	1110
NC	126	42	54	222
TD	6363	2121	3576	12060
TS	9863	3288	5575	18726
Total	**25863**	**8620**	**14345**	**48828**

Training

- Preprocessing
  - Resize to 232 x 232 for input
  - Subtract image mean from training images
- GRID K520 4GB GPU
- Stopped at 90% validation accuracy
- 65 epochs in 8 hours



Visualization



Feature maps from second convolution

Initial performance

- Model with around 90% of validation accuracy
- Tested against 14,345 test images (Atlantic + Pacific)
  - Confusion Matrix
  - Classification Report
  - Accuracy
  - RMS Intensity Error

Accuracy

- Top-1: exact-hits
- Top-2: exact-hits + 2^{nd}-hits

	Total Counts	Accuracy
Top-1	11571	80.66%
Top-2	13695	95.47%

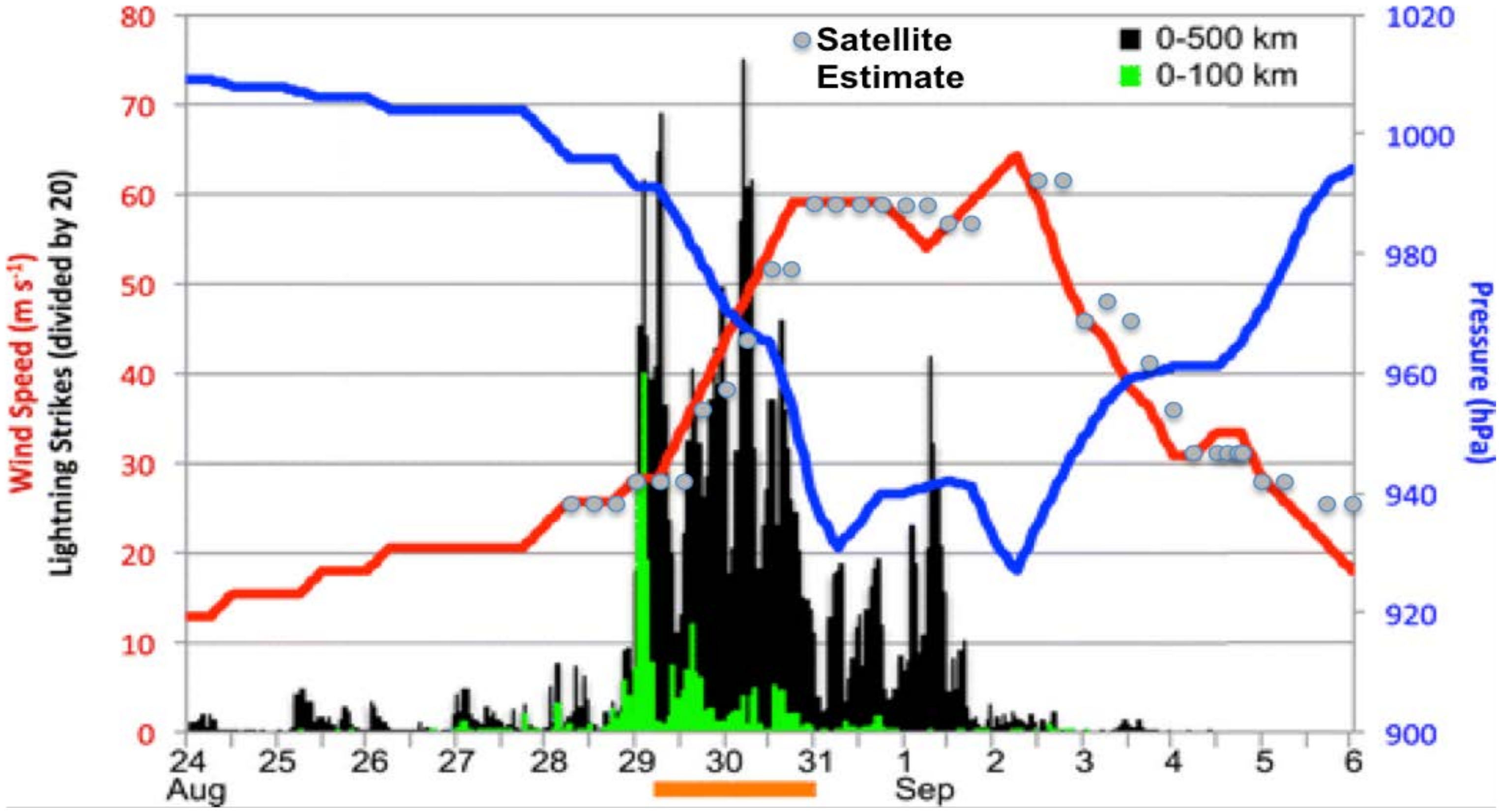
Category	Total	Top-1	2^{nd} hit	Top-2
NC	54	32	15	47
TD	3576	3174	364	3538
TS	5575	4838	665	5503
H1	1816	1235	432	1667
H2	994	614	215	829
H3	992	657	212	869
H4	1032	816	148	964
H5	306	205	73	278
Total	**14345**	**11571**	**2124**	**13695**

Error Metrics

- Our model
  - Across Atlantic and Pacific
  - Achieved RMSE of 9.19 kt
- North Atlantic
  - Piñeros et al. (2011): 14.7 kt
  - Ritchie et al. (2012): 12.9 kt
- North Pacific
  - Ritchie et al. (2014): 14.3 kt

Category	RMSE	MAE
NC	10.14	6.19
TD	6.59	2.18
TS	7.68	2.71
H1	12.17	6.59
H2	12.43	6.82
H3	12.44	6.31
H4	10.50	4.09
H5	10.08	5.32
Total Average	**9.19**	**3.77**

Detailed look: Hurricane Earl, 2010



Adapted from Stevenson et al. (2014). Time series of satellite-derived intensity estimates (circles) for Hurricane Earl (2010), added to best track intensities and lightning flash rate time series.

Hurricane Wind Speed Estimation Portal

- Deploying model in production

Challenges

- Deep Learning Black Box
- Training Data
- Deploying the model in production
- Boundary between data and code
- Consistent training data

Deep Learning Black Box

- How it works?
- “Self-learning machines”
- Uncertainty
- Building trust

Large scaled labeled training data

- Algorithms can be fine tuned for customized applications
- Successful applications have one thing in common
  - Large number of data points needed to learn large number of parameters
- Barrier for using deep learning
- ~~Data~~ Training Data is the NEW oil
- Manually creating labeled training data is bottleneck

Examples

	VGGNET	DeepVideo	GNMT
Task	Identify image	Identify video	Translate
Input Data	Image	Video	English Text
Output	1000 Classes	47 Classes	French Text
# of Parameters	~140 million	~100 million	~380 million
Labeled Data Size	1.2 million images	1.1 million videos	6 million sentence pairs 340 million words

Labeled training data

Application	Training Data Size ~	Methodology
Hurricane intensity (wind speed) estimation	97,000+	Combining imagery with storm database
Severe storm (hailstorm) detection	93,000+	Storm reports
Transverse bands detection	9,000+	Manual
Dust climatology	8,000+	Manual
Ephemeral water detection	650,000+	Combining shapefiles and time series analysis

Existing strategies to increase training size

- Data Augmentation
- Transfer Learning
- Permutation Invariance
- Data Programming

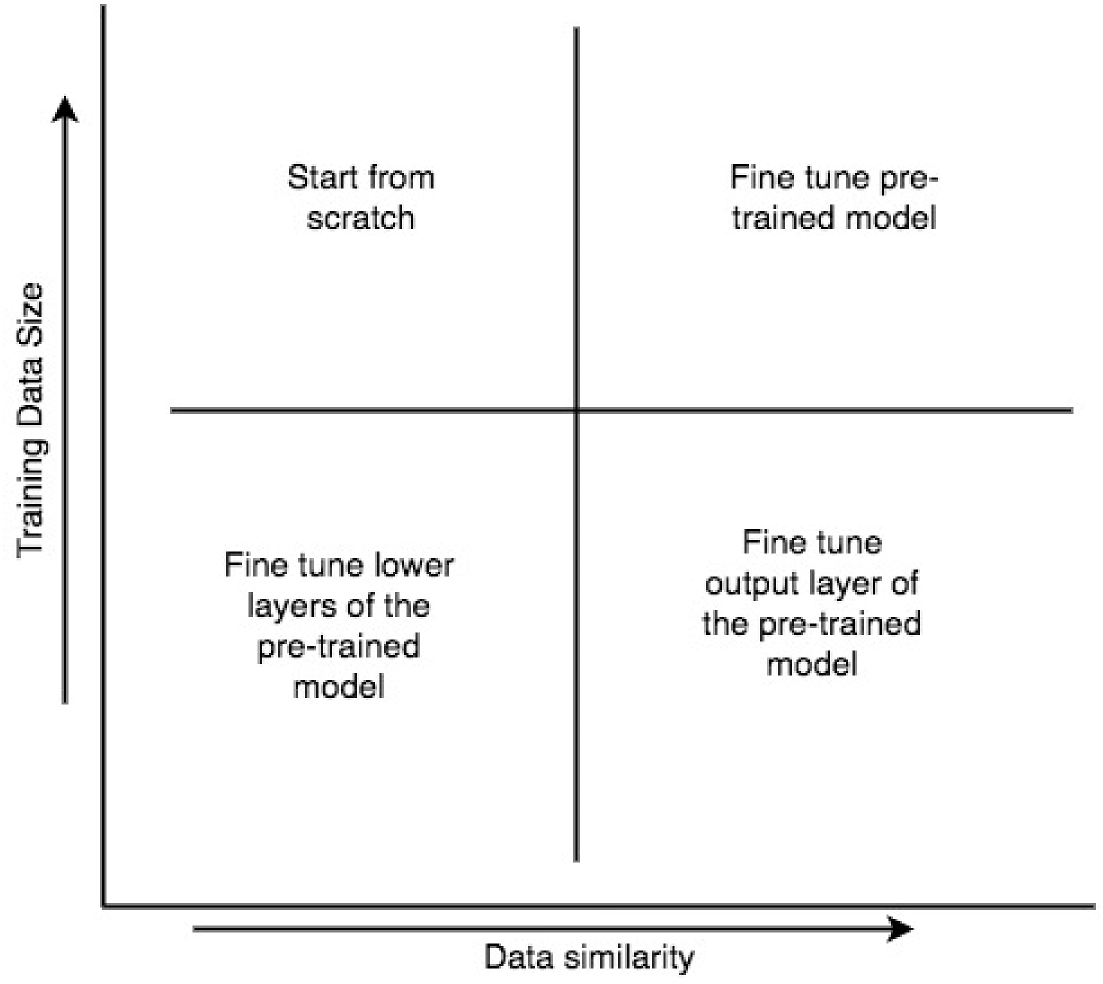
Data augmentation

- For computer vision tasks
- Mirroring
- Random cropping
- Color shifting
- PCA

Transfer Learning

- Network gains knowledge from training data
- Compiled as “weights” of the network
- Weights can be extracted and then transferred to another network
- Instead of training network from scratch, “transfer” the learned features
- Pre-trained model
  - Created by someone else to solve similar problem
- Ways to fine tune the model
  - Feature extraction
  - Architecture
  - Train some – freeze some

Using pre-trained models



Data programming

- Programmatic creation of training dataset
- User
  - Provides unlabeled data
  - Writes labeling functions (LFs)
    - expresses supervision strategies (domain heuristics)
  - Chooses a discriminative model

Weak supervision

- Distant supervision
- Crowdsourcing
- Weak classifiers
- Domain rules/heuristics

Example

- Information Extraction from Earth Science Literature
- Unstructured text
- Extract information: dataset usage, hypothesis validation, etc.
- No large labeled training dataset
- Various ontologies, vocabularies, and glossaries?
- Custom heuristics?
- Regular expressions
- Rule-of-thumb
- Negative label generation

Studying dust events

Sample text:

"Pronounced changes in the aerosol optical parameters, derived from AERONET, have been observed during dust storms."

```
def labelingFunction1(input):
    concept = (input.phenomenon,input.property)
    return 1 if concept in DOMAIN_KB else 0


def labelingFunction2(input):
    found = re.search(r'.*derived.*',input.text.between)
    return 1 if found else 0
```

Sample Labeling Functions to extract mentions of dust events and properties

- labelingFunction1: Leverage existing Earth Science knowledgebase (e.g., SWEET)
- labelingFunction2: Domain heuristics

Snorkel

- Data programming framework
- Creates a noisy training set – by applying LFs to the data
- Learns a model of the noise (learns accuracy of LFs)
- Trains a noise-aware discriminative model

Process of training data creation

- Model noise in training set creation process
- Use low-quality sources to train high-quality models
- Traditional “distant supervision” rule: external knowledgebase
- Learn accuracy and correlations for a handful of rules

Our Approach: creating labeled datasets

- Existing strategies
- External database
- Unstructured data
- Expert labeling
- Labeling tool
- Citizen science
- Validation data

Application	Training Data Size ~	Methodology
Hurricane intensity (wind speed) estimation	49,000	Combining imagery with storm database
Severe storm (hailstorm) detection	93,000	Storm reports
Transverse bands detection	9,000	Manual
Dust climatology	8,000	Manual
Ephemeral water detection	650,000	Combining shapefiles and time series analysis

Use case – Creating Training Dataset

Detect hail in NASA GPM GMI measurement

- Hail not only results in damages but also contaminates passive microwave-based rainfall retrievals, which are the primary means for global precipitation measurement
- Current methods for hail detection use radar or rely on single passive microwave (PMW) frequencies (e.g., 37 GHz)

Use case – Creating Training Dataset

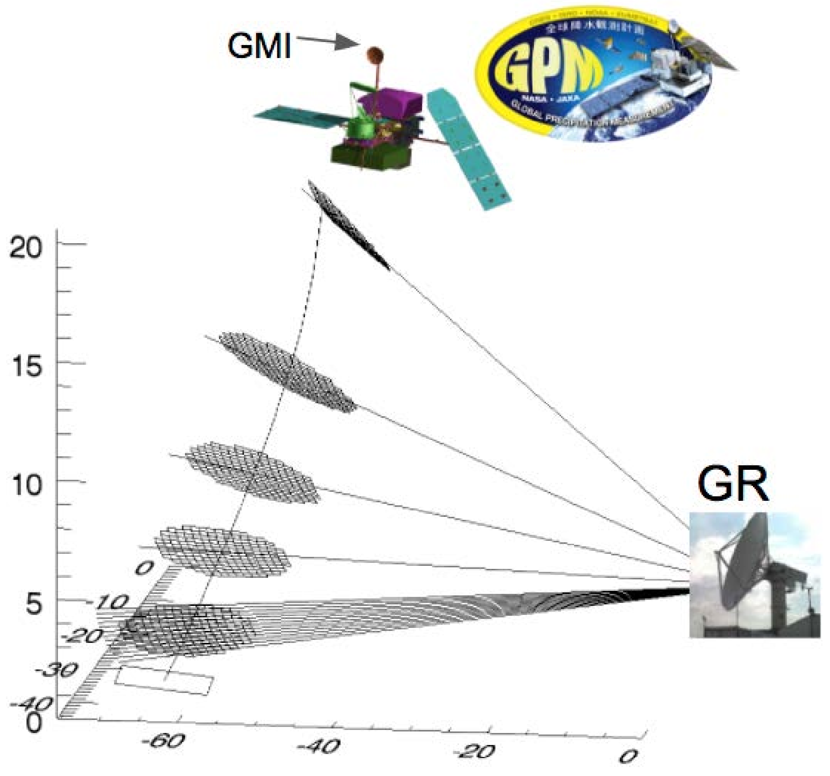
Detect hail in NASA GPM GMI measurement

- Hail not only results in damages but also contaminates passive microwave-based rainfall retrievals, which are the primary means for global precipitation measurement

- Current methods for hail detection use radar or rely on single passive microwave (PMW) frequencies (e.g., 37 GHz)

Approach

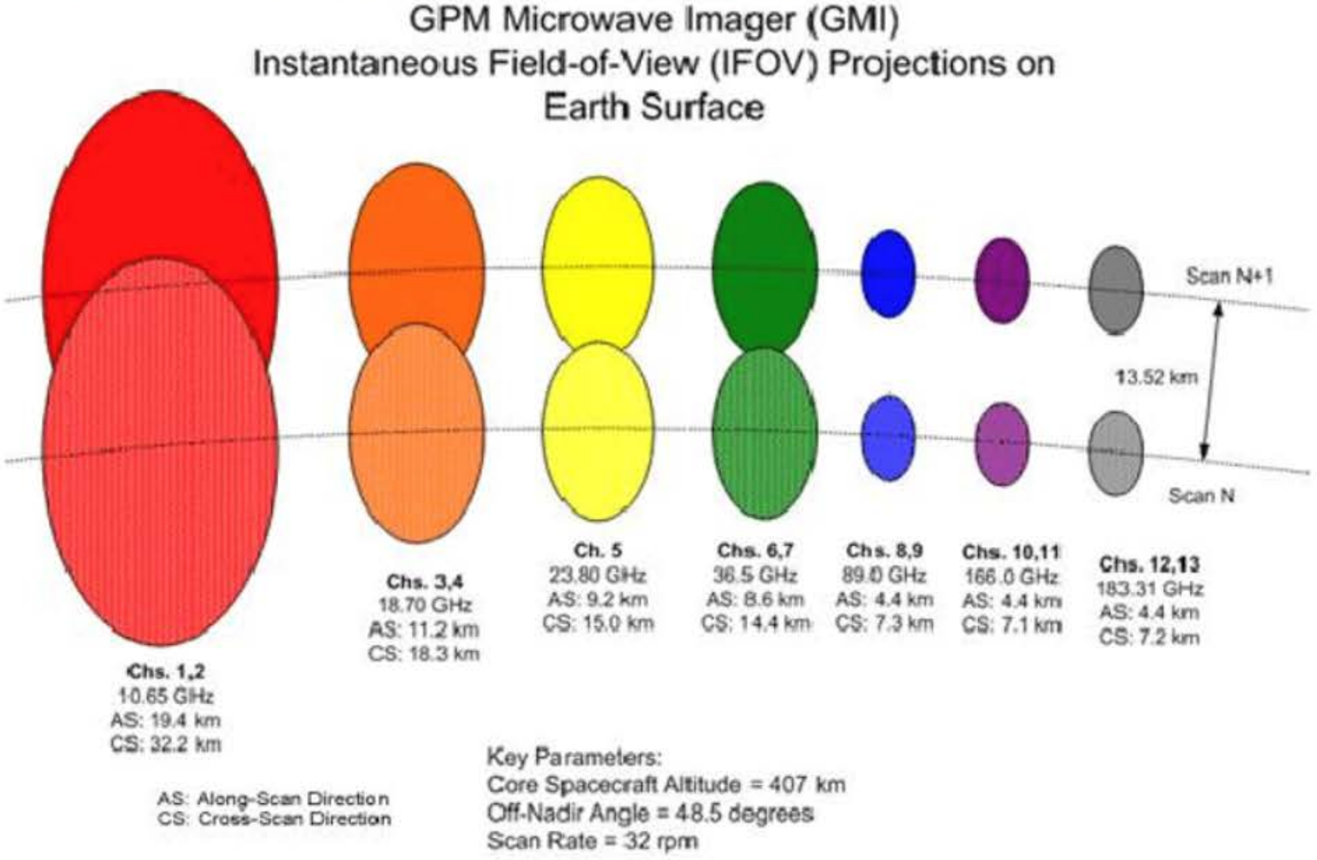
- Use GPM GV data (GHRC) to provide large dataset of ground "truth"
- Constrain GMI-GR matchups to where GR indicates hail
- Combine GMI channels into a common coordinate system
- Create plots to train a CNN



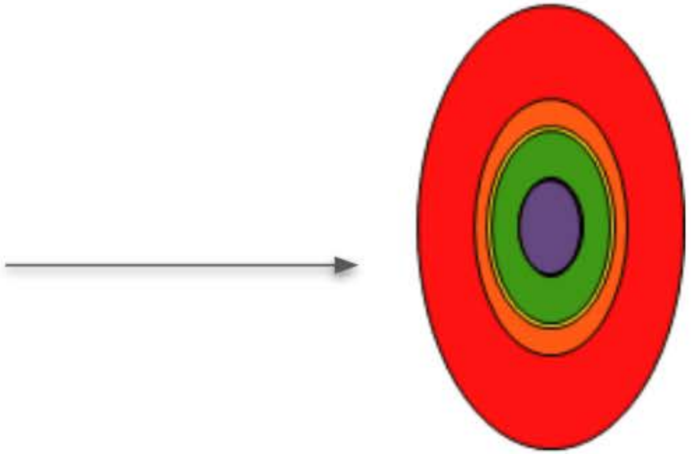
*GPM GV consists of **geometrically matched** GMI (rectangle) and Ground Radar (GR) measurements (waffles) like that illustrated here. The GMI views the ground along a slanted path and provides a 2-D measurement of precipitation. The GR views the atmosphere along cones and provides a 3-D measurement of precipitation. The information contained at the intersection of the GR cones with the GMI slant path provides a bulk characterization of precipitation within affecting the GMI measurement and subseqent precipitation retrievals.*

Generating GMI image

Surface footprints of GMI channels



"Bullseye" of GMI channels



This is actually how the instrument views the earth at each channel.

Labeled training data

Application	Training Data Size ~	Methodology	Strategy
Hurricane intensity (wind speed) estimation	97,000+	Combining imagery with storm database	Data Augmentation
Severe storm (hailstorm) detection	163,000	Storm reports	None
Transverse bands detection	9,000	Manual	Data Augmentation and Transfer Learning
Dust climatology	8,000	Manual	Data Augmentation and Transfer Learning
Ephemeral water detection	650,000	Combining shapefiles and timeseries analysis	None

Publishing dataset

- Should Earth science training dataset be published as other datasets?
- Catalog – NASA CMR?

Available Public Datasets on AWS

Geospatial and Environmental Datasets

Learn more about working with geospatial data on AWS at Earth on AWS.

- Landsat on AWS: An ongoing collection of satellite imagery of all land on Earth produced by the Landsat 8 satellite.
- Sentinel-2 on AWS: An ongoing collection of satellite imagery of all land on Earth produced by the Sentinel-2 satellite.
- GOES on AWS: GOES provides continuous weather imagery and monitoring of meteorological and space environment data across North America.
- SpaceNet on AWS: A corpus of commercial satellite imagery and labeled training data to foster innovation in the development of computer vision algorithms.

Deploying Model in Production

- Research
  - Acceptable accuracy
  - Nice charts > publish paper
- In production:
  - Load your model with its weights
  - Preprocess your data
  - Perform the actual prediction
  - Handle the prediction response data
- Issues:
  - Does the model confidence remain the same over time?
  - How do you maintain?
  - Complete the loop with new training data

Deploying Model in Production

- Performance requirements
  - Metrics and baselines with initial model
  - Monitor over time

- Back-testing
  - Model, Data and Software will change
  - Automate the evaluation of production model
    - Back-testing model changes on historical data
    - More than hyperparameter tuning
    - Needs clear demarcation
    - Run the current operational model to baseline performance
    - Run new models, competing for a place to enter operations
    - Run periodically and generate automatic reports
- Now-testing
  - Test of production model on latest data
  - Idea is to get early warning that the model may be faltering
    - Content drift
    - Training data exploited by your model are subtly changing with time

Key takeaways

- Deep learning ideal for "Supervised" learning
- Algorithms can be fine tuned for customized applications
- **Large labeled datasets** fuel impressive classification accuracy
- **Challenge:**
  - *Creating/Identifying/Accumulating large labeled datasets*

Thank you.

Manil Maskey
manil.maskey@nasa.gov