Flexible and Scalable Deep Learning with MMLSpark

Mark Hamilton
marhamil@microsoft.com

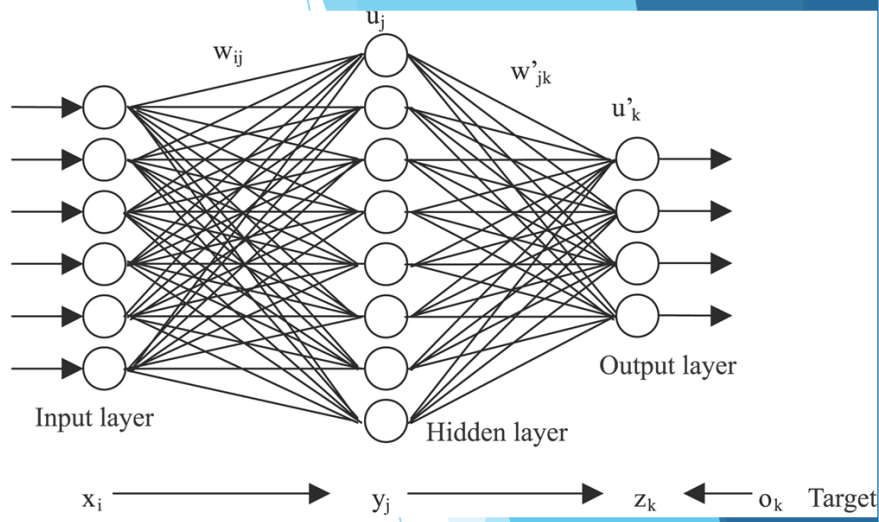
MMLSpark
aka.ms/mmlspark

Agenda

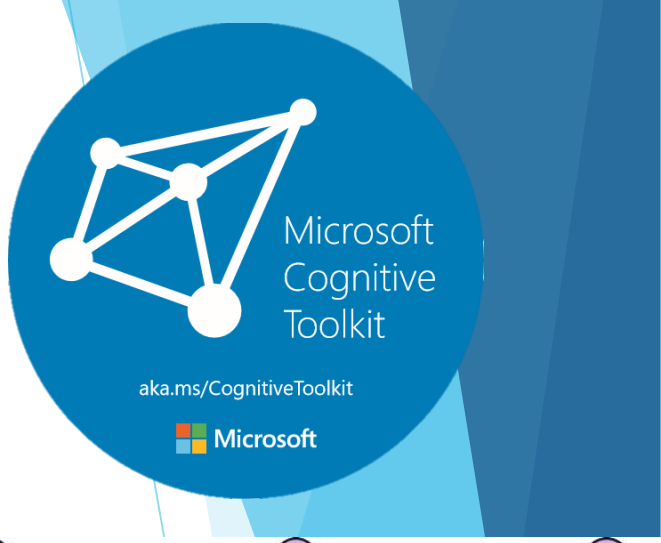
- Background
  - Deep Learning
  - Cognitive Toolkit (CNTK)
  - Apache Spark and SparkML
- MMLSpark
  - Integrating Cognitive Toolkit and Spark
  - OpenCV Spark integration
  - PySpark Wrapper Generation
- Snow Leopard Conservation
  - Transfer Learning
- Future Work
  - CNTK Training
  - Text Analytics

Deep Learning

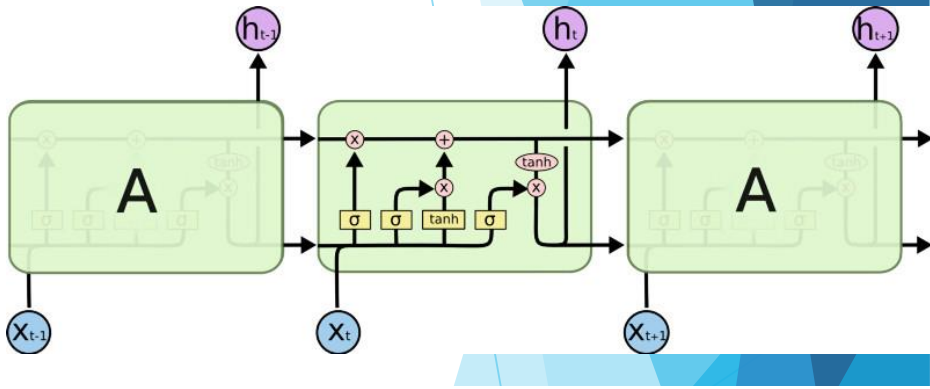
- Originally referred to learning neural networks with many layers
- Now refers to any suitably complicated statistical model trained with gradient descent
- Has become a recent favorite because:
  - Spectacular performance in many domains
  - Quick training w/ gradient descent
  - Low memory footprint w/ Stochastic Gradient Descent (SGD)
  - Large space of possible model architectures
  - Automatic differentiation software and APIs

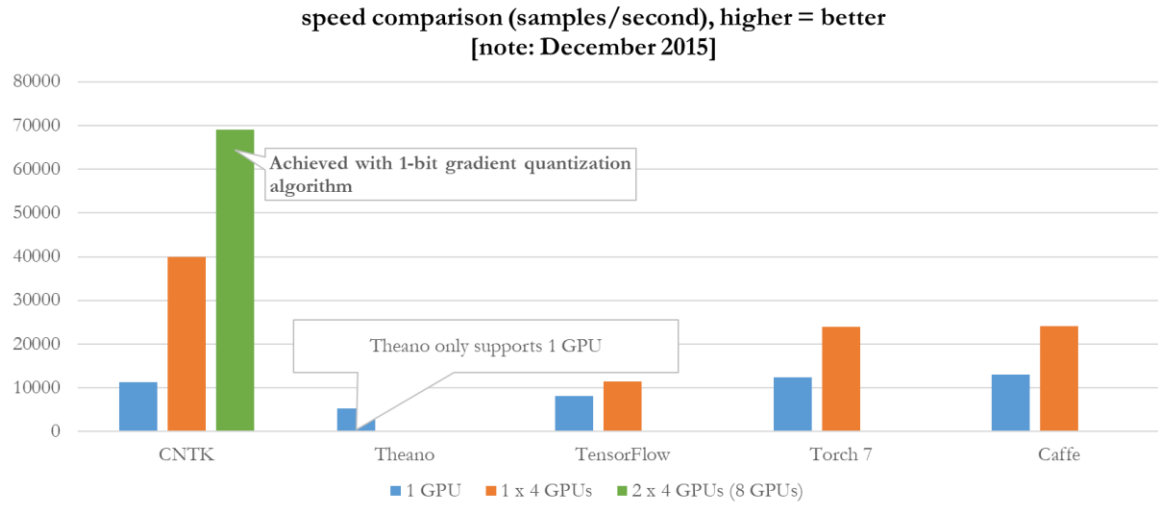


Microsoft Cognitive Toolkit (formerly CNTK) (Dong Yu et al. 2012)

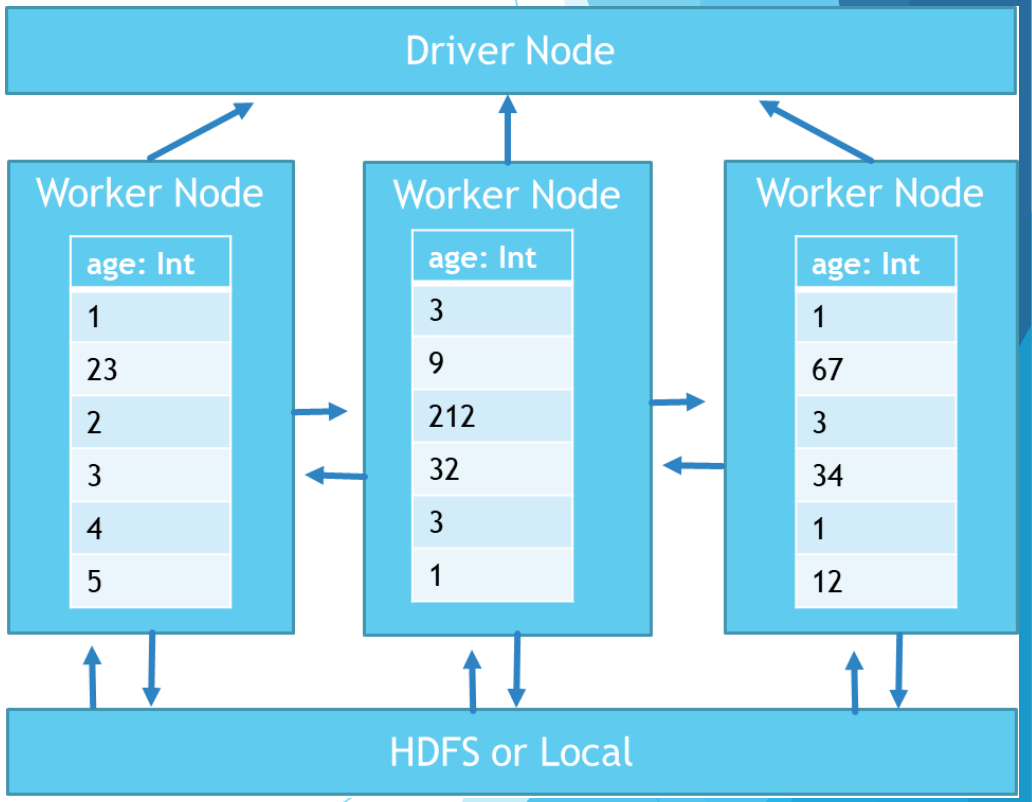


- Microsoft's open-source deep-learning toolkit
- https://github.com/Microsoft/CNTK
- Written in C++, bindings in Python, C#
- Runs over 80% Microsoft internal DL workload
- Can express a huge variety of deep architectures
  - LSTMS, ConvNets, RL, Gans, etc….
- Automatic differentiation
- Compute Agnostic: Compiles down to performant machine code
- Fast!





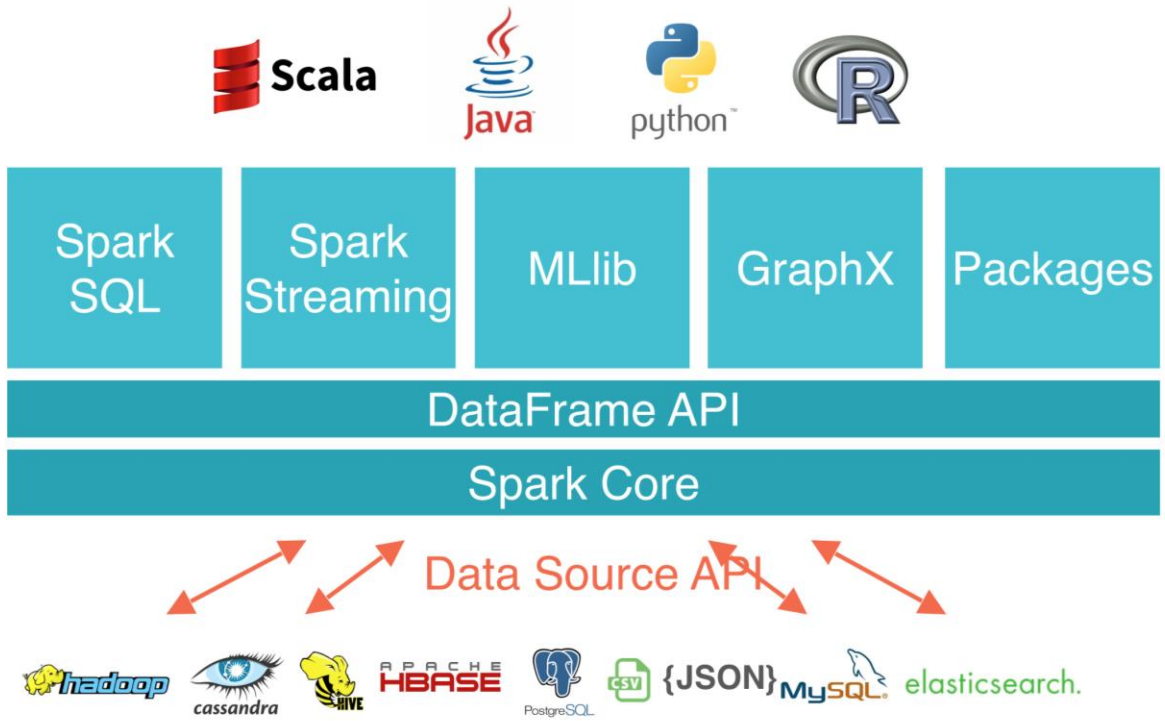
- A **fault-tolerant distributed** computing framework
- Generalizes, optimizes, and combines Map-Reduce and SQL style-operations
- Scales to thousands of machines
- Functional API
- Built in Scala, but has bindings in Python and R
- Has a flourishing community
  - SparkML
  - GraphX
  - Structured Streaming



Spark ML: Machine Learning at Scale

- High level library for machine learning
- Similar abstraction to SciKit Learn (But waaaaay better API)
- Written in Scala but has wrappers in Python
- Models all have a uniform interface: PipelineStage
  - Classification: logistic regression, naive Bayes, ...
  - Regression: generalized linear regression, survival regression, ...
  - Decision trees, random forests, and gradient-boosted trees
  - Recommendation: alternating least squares (ALS)
  - Clustering: K-means, Gaussian mixtures (GMMs), ...
  - Topic modeling: Latent Dirichlet Aallocation (LDA)

```
data = spark.read.csv("hdfs://...")
train, test = data.randomSplit([.5,.5])
model = LogisticRegression().fit(train)
predictions = model.transform(test)
```



Drawbacks:

Spark/SparkML

- Only “conventional” algorithms
  - Difficult to add new models, tons of boilerplate, restricted to certain kids of operations
  - All gradient based algorithms use custom SGD code
- No support for complex datatypes like images
- Need to manually write wrappers to get code into PySpark and SparkR

CNTK

- CNTK could not be scaled with fault tolerance
  - required expensive and dedicated HPC machines
- No support for elasticity or autoscaling
- No support for high throughput parallel streaming
- Mainly support for gradient based models

Microsoft Machine Learning For Apache Spark

Distributed Deep Learning, Image Analysis, Text Analytics, and Much More

https://aka.ms/mmlspark

MMLSpark

aka.ms/mmlspark

- Open Source
  - Contributions welcome!
  - Get started with our docker image + data science examples/course aka.ms/mmlspark
- **Integrates and unifies Spark, CNTK, and OpenCV**
- Tons of other useful abstractions and tools:
  - **Automatic Python/R code generation**, Auto-Featurization, Trained DNN Repository, Multi-Column support, Serialization, Fuzzing etc.

Cognitive Toolkit x Spark Step 1: Java Bindings

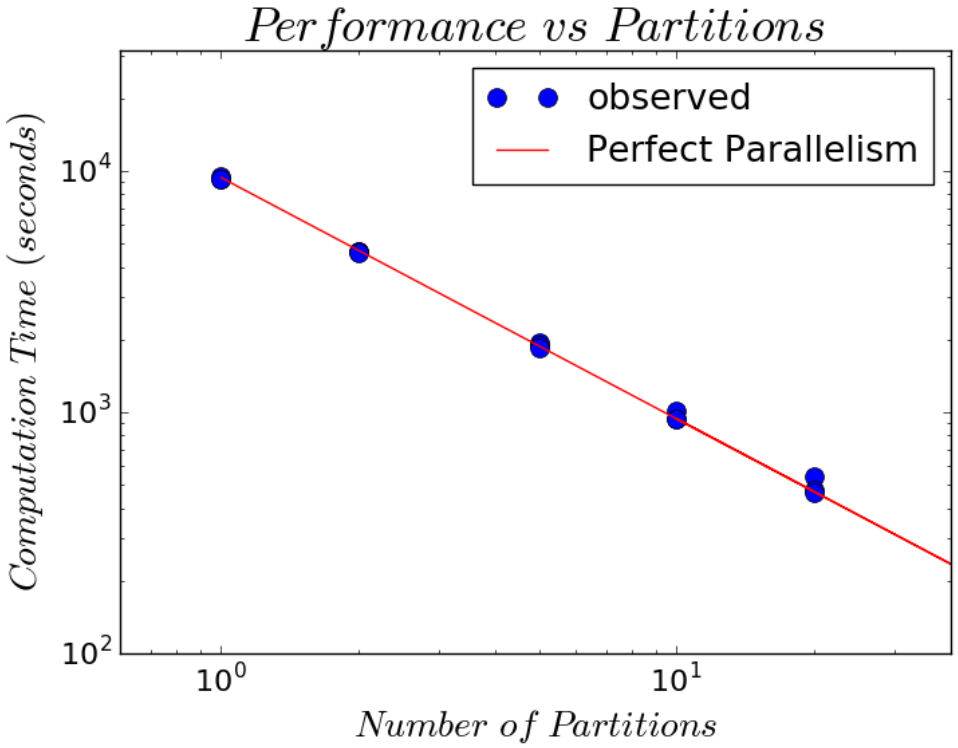
- Cognitive Toolkit is written in C++ but has bindings in Python, Brainscript, and C#
- We use the Simple Wrapper and Interface Generator (SWIG) to expose CNTK's Evaluation library to Java
- All Java bindings are machine generated so they require very little maintenance!
- Try them out in Cognitive Toolkit's > 2.0

```
Function modelFunc = Function.load(new File("resnet20_cifar10_python.dnn"), device);
Variable outputVar = modelFunc.getOutputs().get(0);
Variable inputVar = modelFunc.getArguments().get(0);
```

Cognitive Toolkit x Spark Step 2: Spark Transformers

- Spark is built on Scala (inter-ops with Java), allowing us to use our new CNTK Java Bindings
- Spark allows users to execute custom Scala code on each machine
- We automatically distribute, load, and run the CNTK model on all machines
- Each machine maps a small portion of the total dataset, so performance scales with machines

```
1  val model = new CNTKModel()
2    .setModelLocation(session, modelPath)
3    .setInputCol("images")
4    .setOutputCol("features")
5    .setOutputNodeName("z")
6
7  val result = model.transform(data)
```



Parallel Image Processing with OpenCV

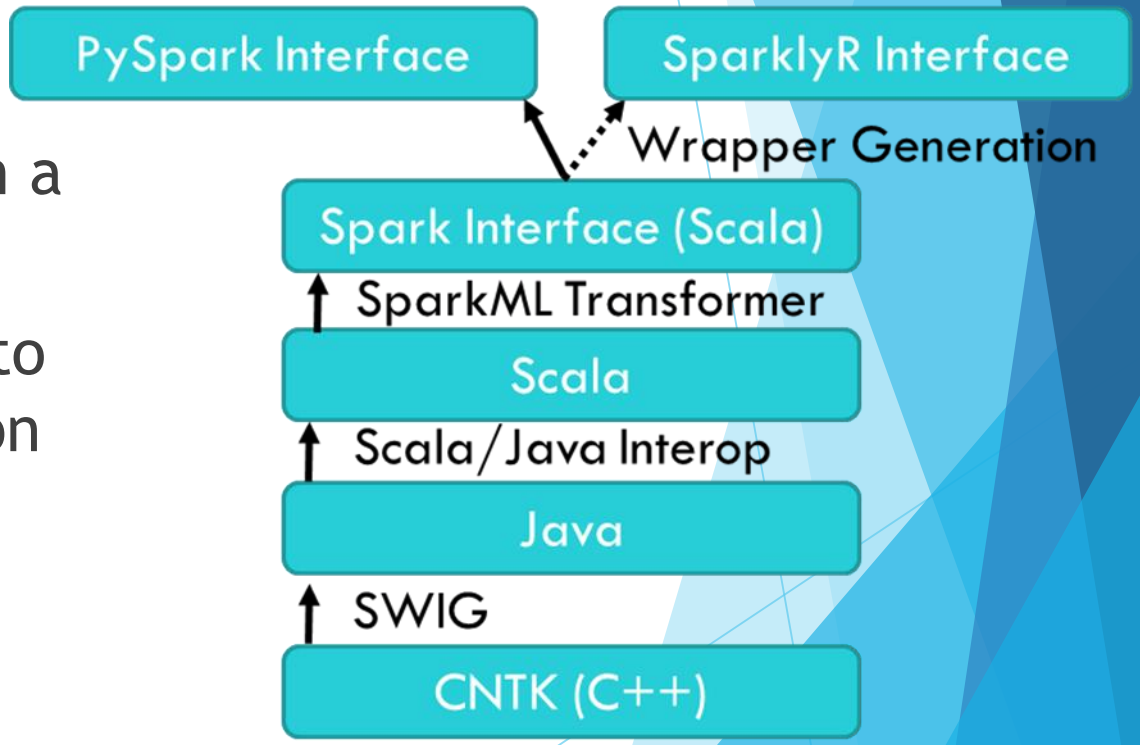
- DNNs are often picky about their input data shape and normalization.
- We provide bindings to OpenCV image processing operations, exposed as SparkML PipelineStages:

```
tr = ImageTransform().setOutputCol("transformed")
    .resize(height = 200, width = 200)
    .crop(0, 0, height = 180, width = 180)

smallImages = tr.transform(images).select("transformed")
```

PySpark Bindings…For Free!

- Our core code is written in Scala
- Python is too hot to forget
- Spark has a exposed bindings to python in a package called PySpark
- We automatically expose all of our work to python through generating SparkML python APIs
- Now also support SparklyR, Spark's R bindings




Snow
Leopard
Trust

Snow Leopard Conservation

- 3,900-6,500 individuals left in the wild
- Little known about their ecology, behavior, movement patterns, survival rates
- More data required to influence survival

Mining

Poaching

Retribution Killing

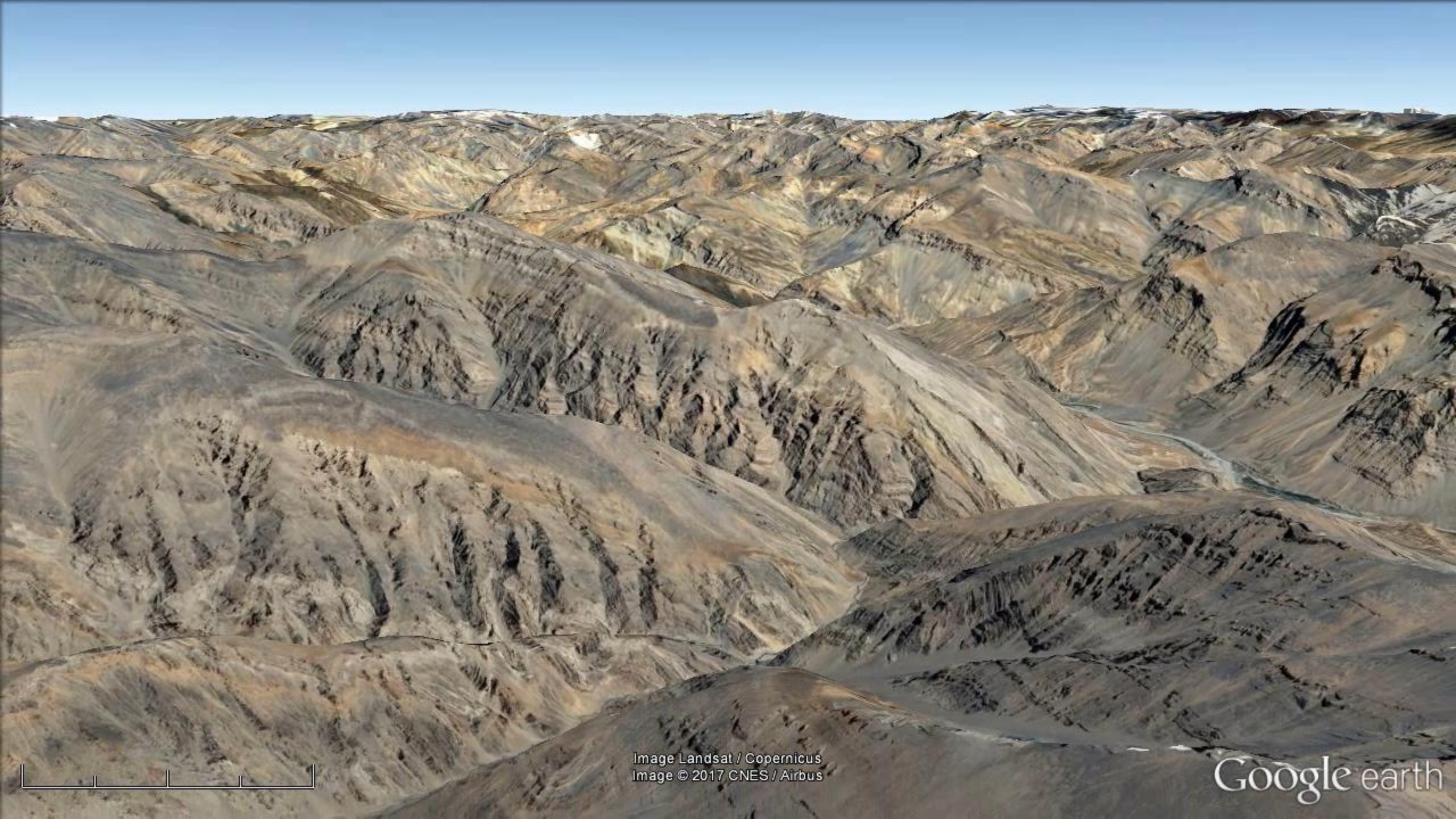
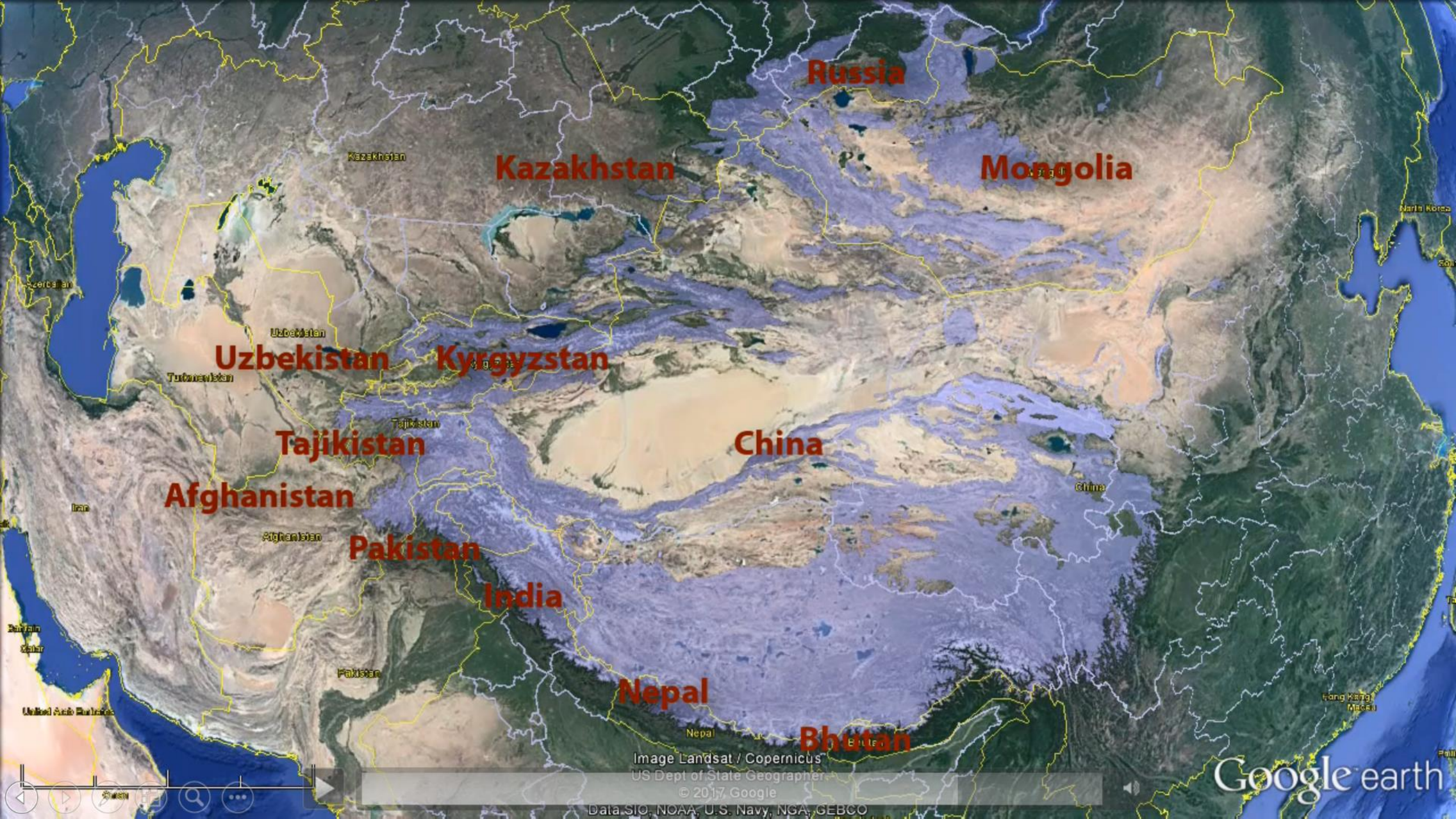
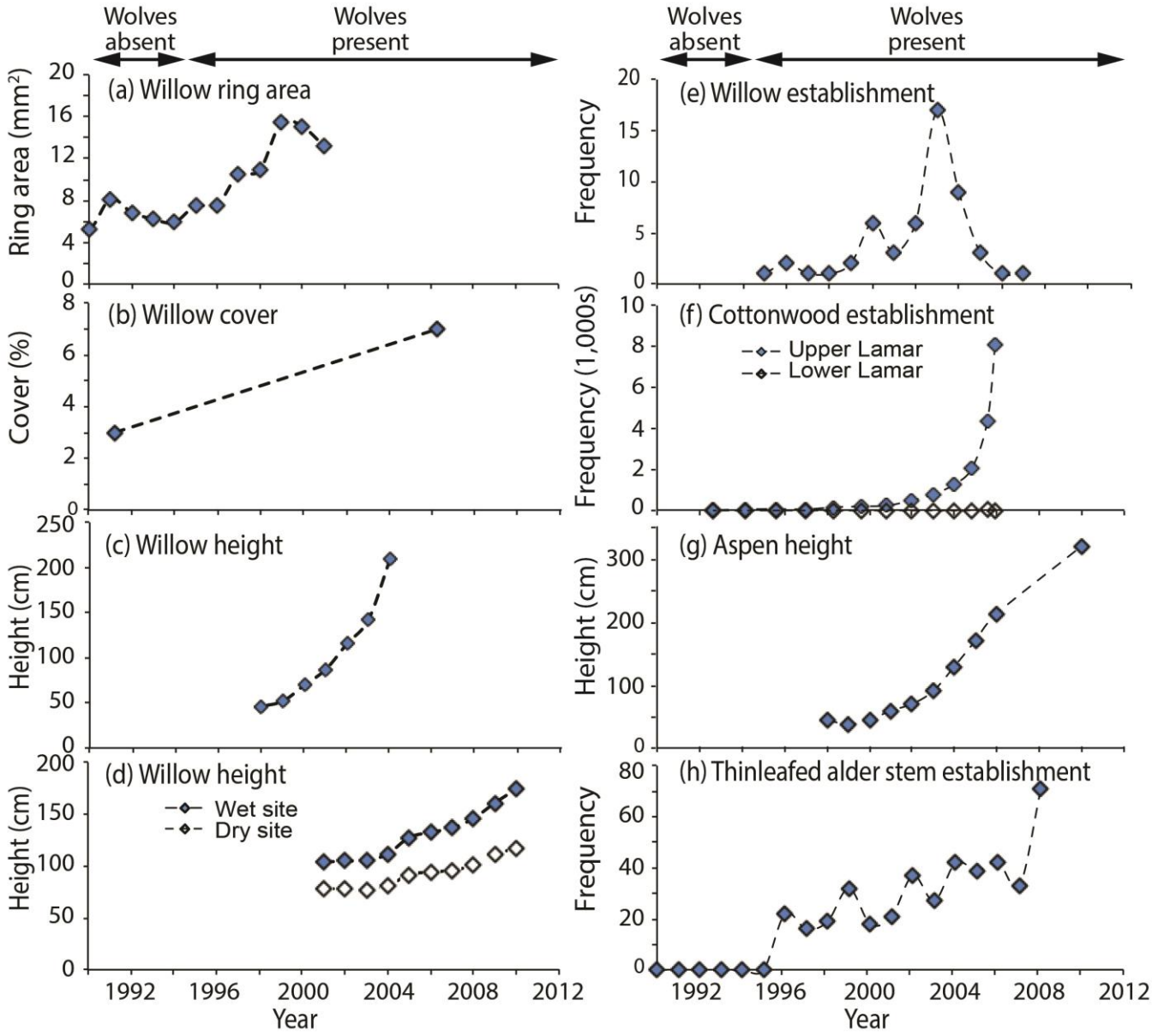


Image Landsat / Copernicus
Image © 2017 CNES / Airbus
Google earth


Russia
Kazakhstan
Mongolia
Uzbekistan
Kyrgyzstan
Tajikistan
China
Afghanistan
Pakistan
India
Nepal
Bhutan
Image Landsat / Copernicus
US Dept of State Geographer
© 2017 Google
Data SIO, NOAA, U.S. Navy, NGA, GEBCO
Google earth

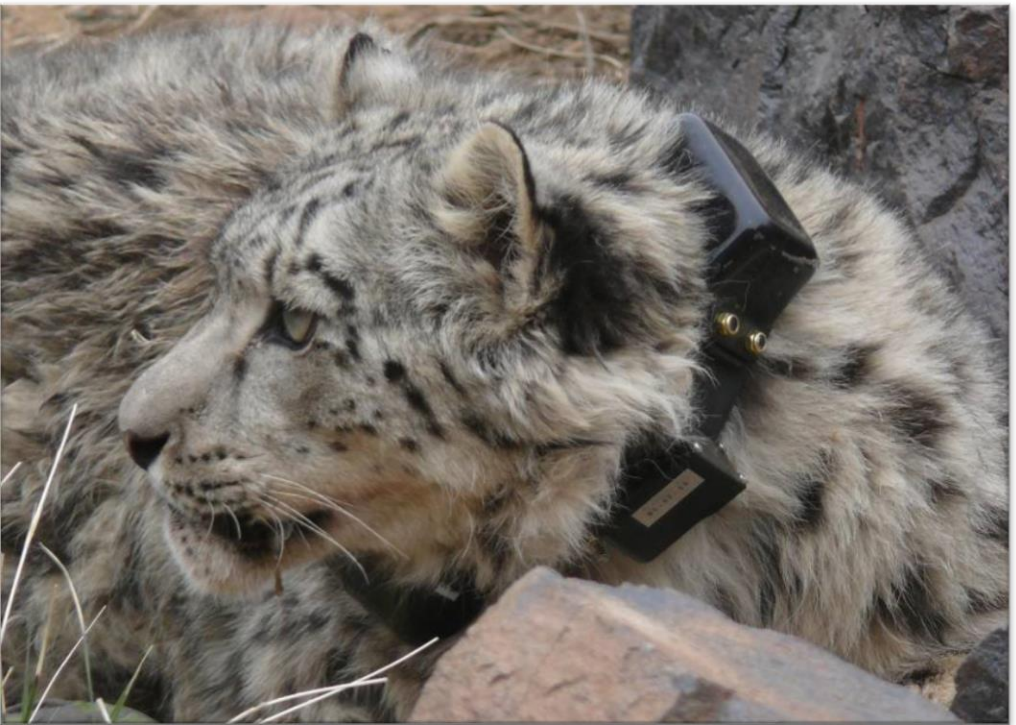
The Impact of Apex Predators

Beschta, Robert L., and William J. Ripple. "Riparian vegetation recovery in Yellowstone: The first two decades after wolf reintroduction." *Biological Conservation* 198 (2016): 93-103.



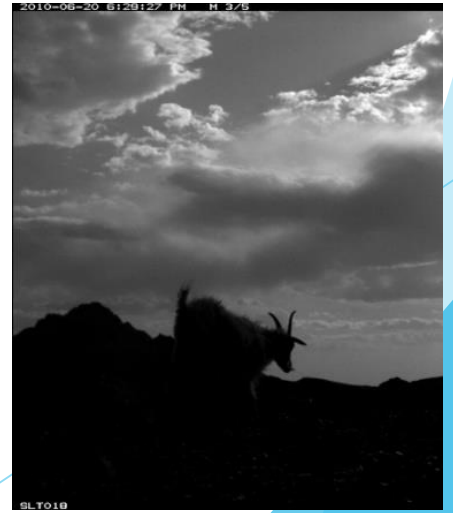
Gathering Leopard Data

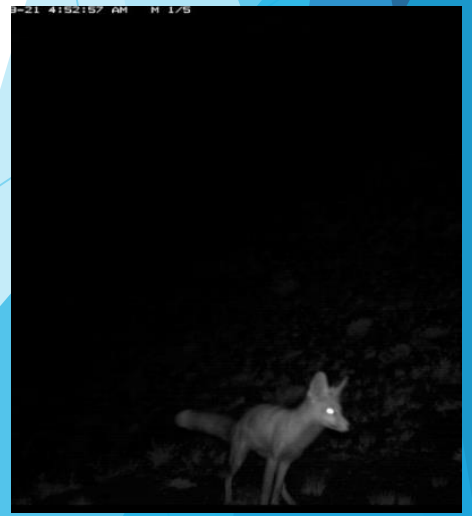
- 42 camera traps over 1,700 sq km
- ~1.3 mil images
- 23 leopards collared in 9 years

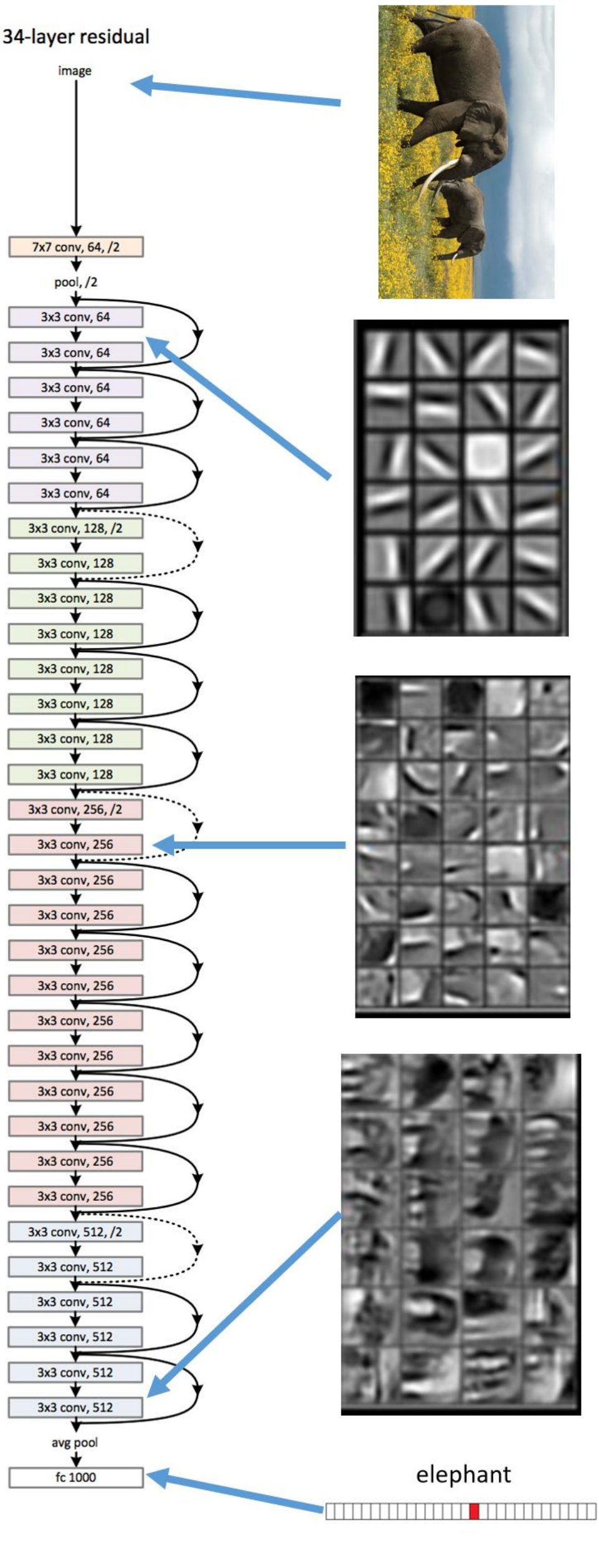


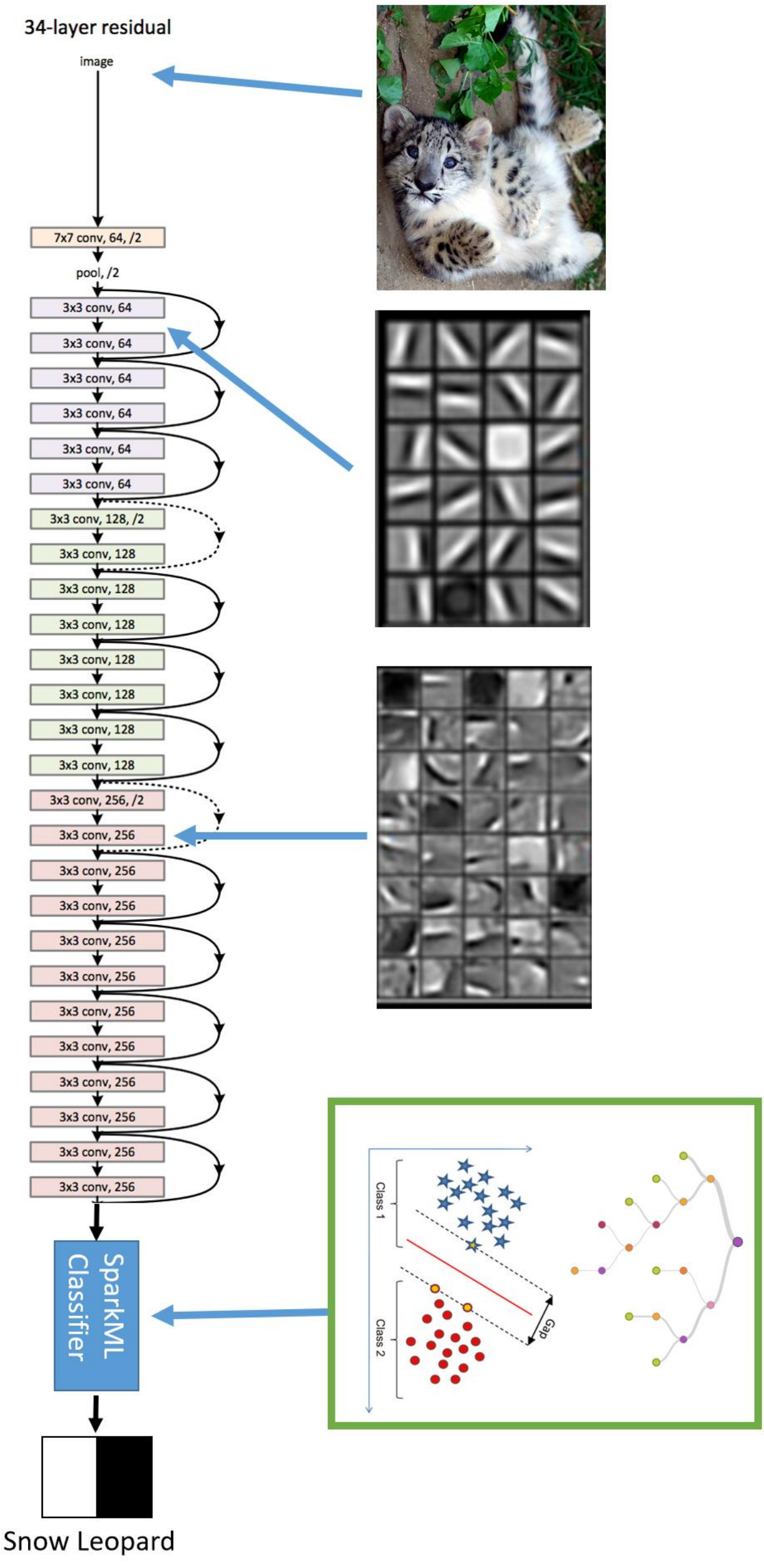
Camera Trap Images

Manually classifying images averages 300 hours per survey

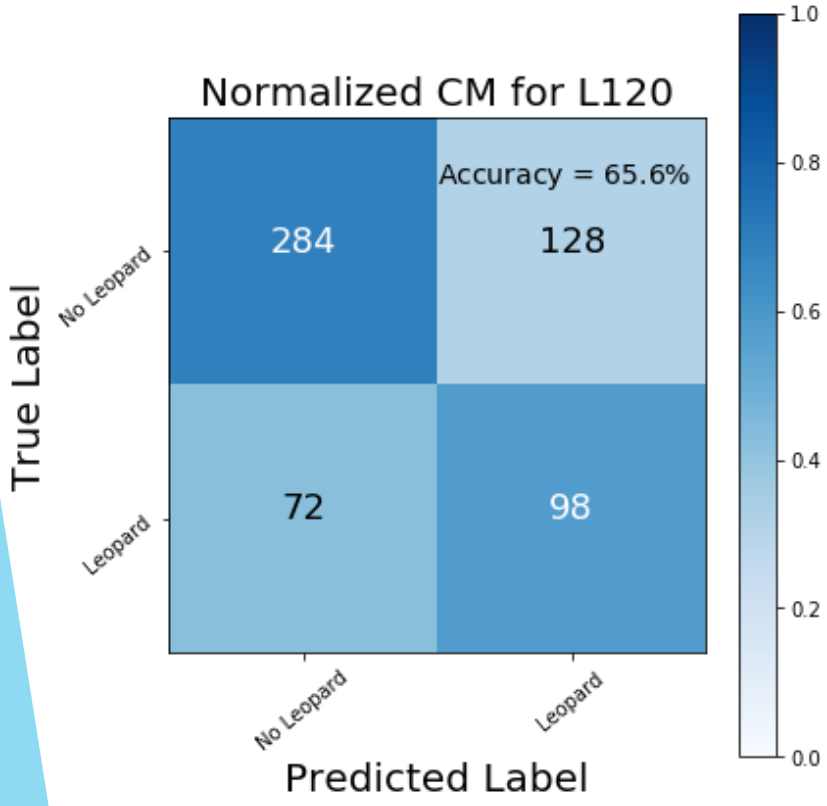


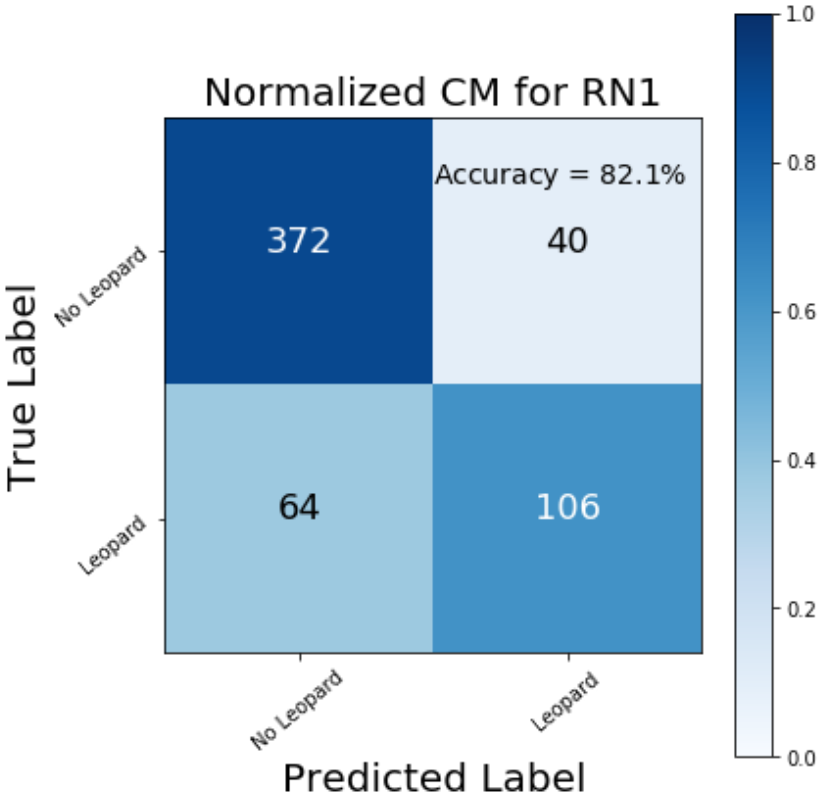


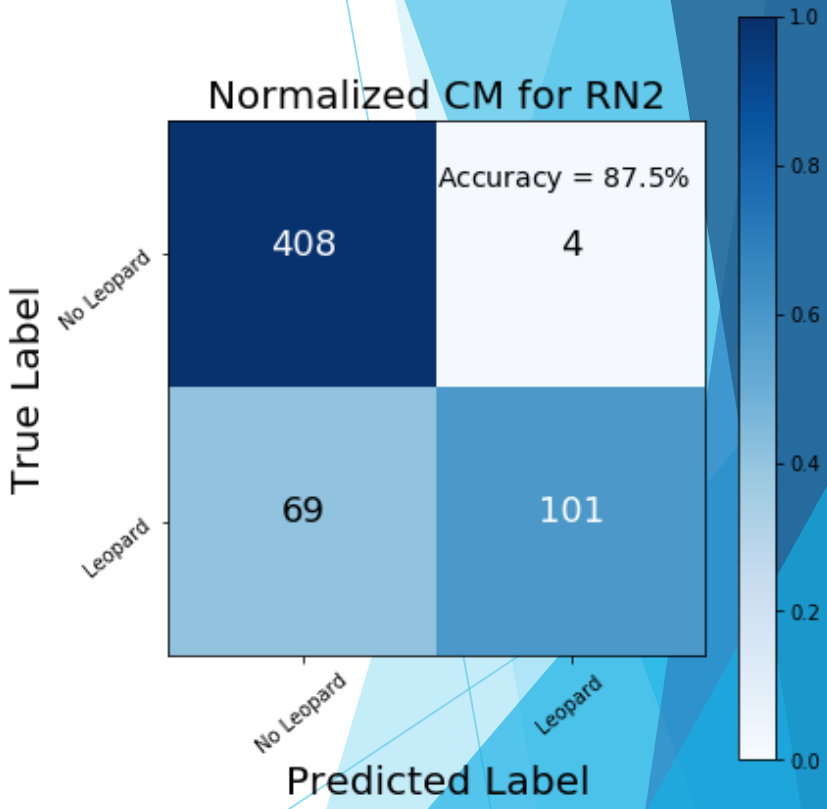

34-layer residual
image
7x7 conv, 64, /2
pool, /2
3x3 conv, 64
3x3 conv, 64
3x3 conv, 64
3x3 conv, 64
3x3 conv, 64
3x3 conv, 64
3x3 conv, 128, /2
3x3 conv, 128
3x3 conv, 128
3x3 conv, 128
3x3 conv, 128
3x3 conv, 128
3x3 conv, 128
3x3 conv, 128
3x3 conv, 256, /2
3x3 conv, 256
3x3 conv, 256
3x3 conv, 256
3x3 conv, 256
3x3 conv, 256
3x3 conv, 256
3x3 conv, 256
3x3 conv, 256
3x3 conv, 256
3x3 conv, 256
3x3 conv, 512, /2
3x3 conv, 512
3x3 conv, 512
3x3 conv, 512
3x3 conv, 512
3x3 conv, 512
avg pool
fc 1000
elephant


34-layer residual
image
7x7 conv, 64, /2
pool, /2
3x3 conv, 64
3x3 conv, 64
3x3 conv, 64
3x3 conv, 64
3x3 conv, 64
3x3 conv, 64
3x3 conv, 128, /2
3x3 conv, 128
3x3 conv, 128
3x3 conv, 128
3x3 conv, 128
3x3 conv, 128
3x3 conv, 128
3x3 conv, 128
3x3 conv, 256, /2
3x3 conv, 256
3x3 conv, 256
3x3 conv, 256
3x3 conv, 256
3x3 conv, 256
3x3 conv, 256
3x3 conv, 256
3x3 conv, 256
3x3 conv, 256
SparkML Classifier
Class 1
Class 2
Gap
Snow Leopard

Performance





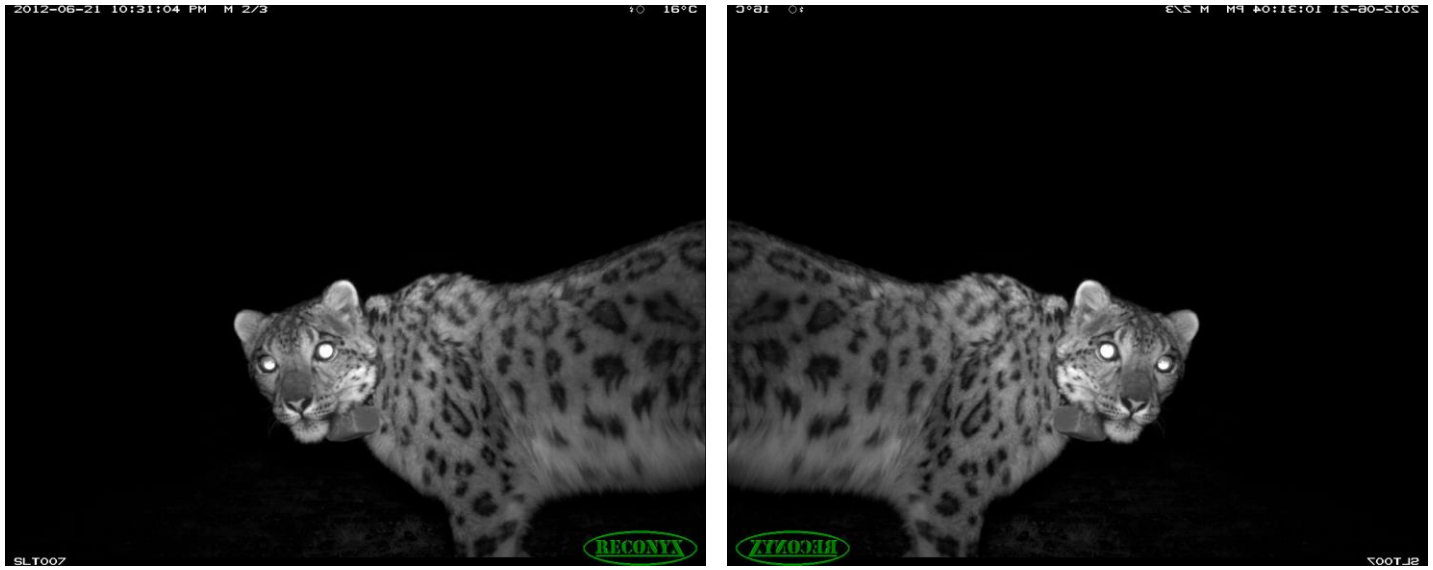


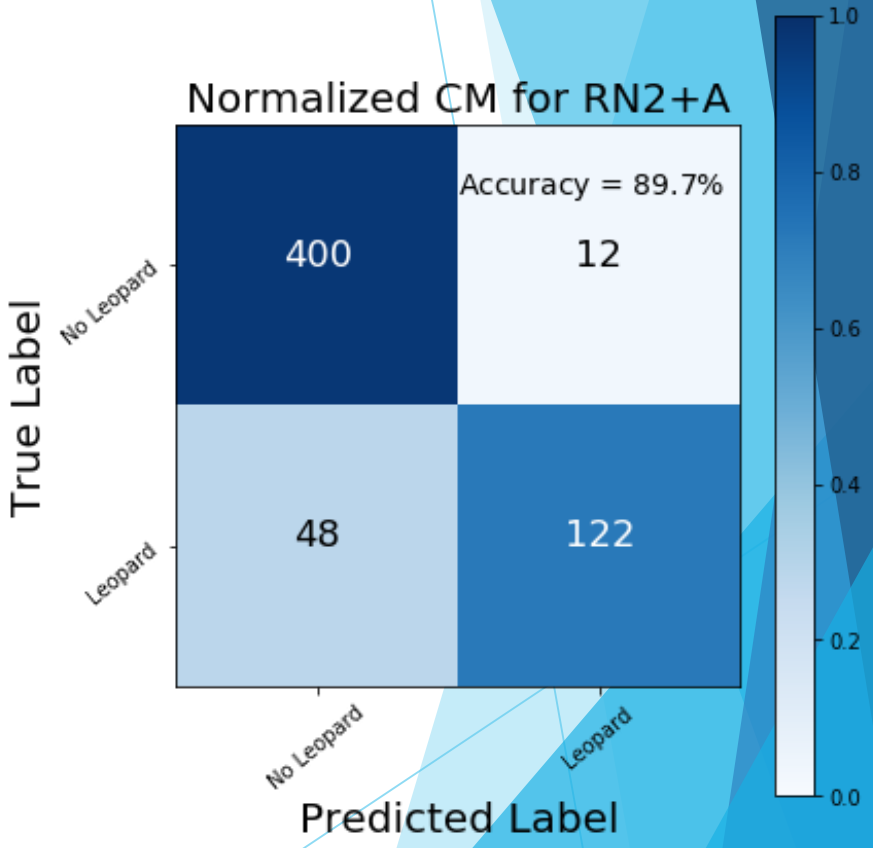
Without Deep Featurization

With Deep Featurization

Further Refinements

- Dataset Augmentation



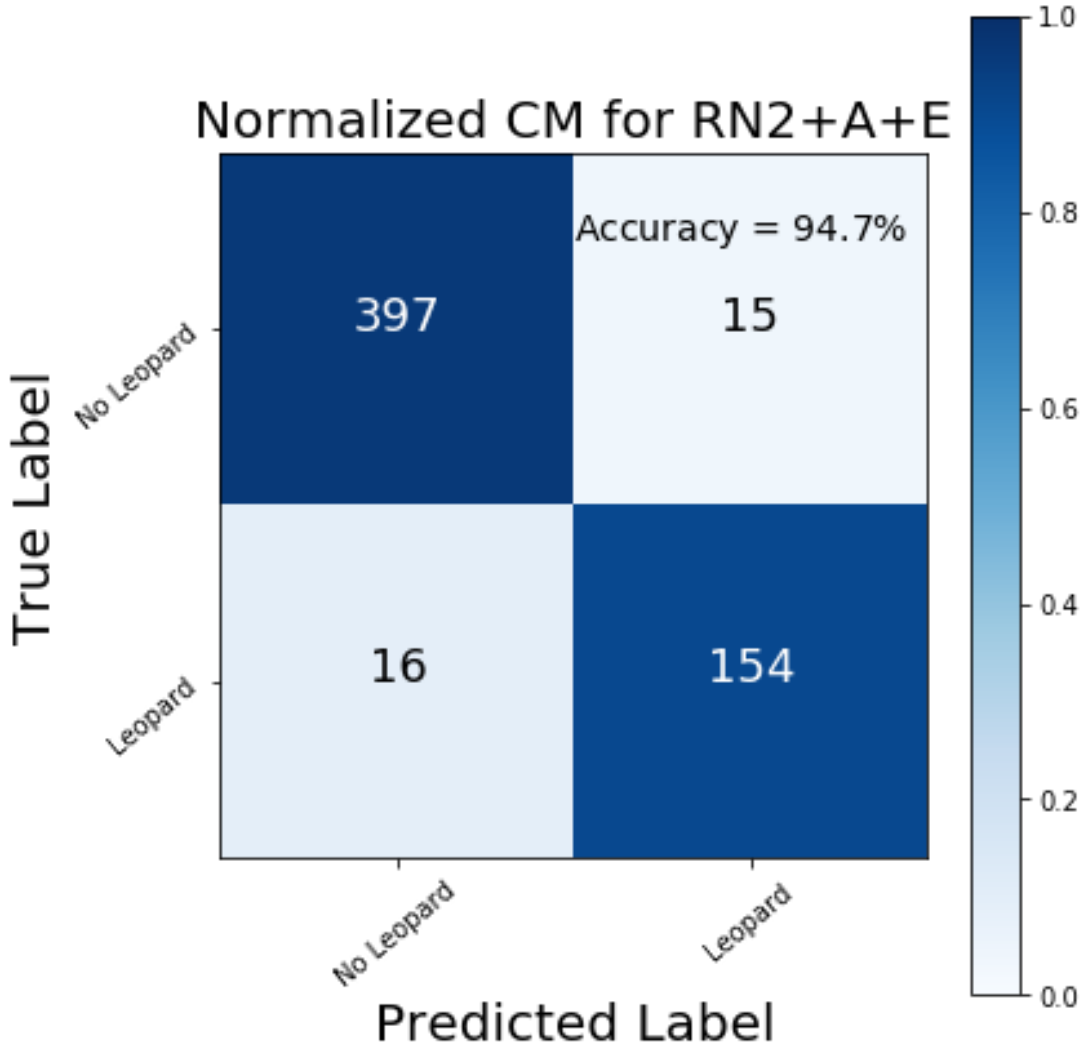


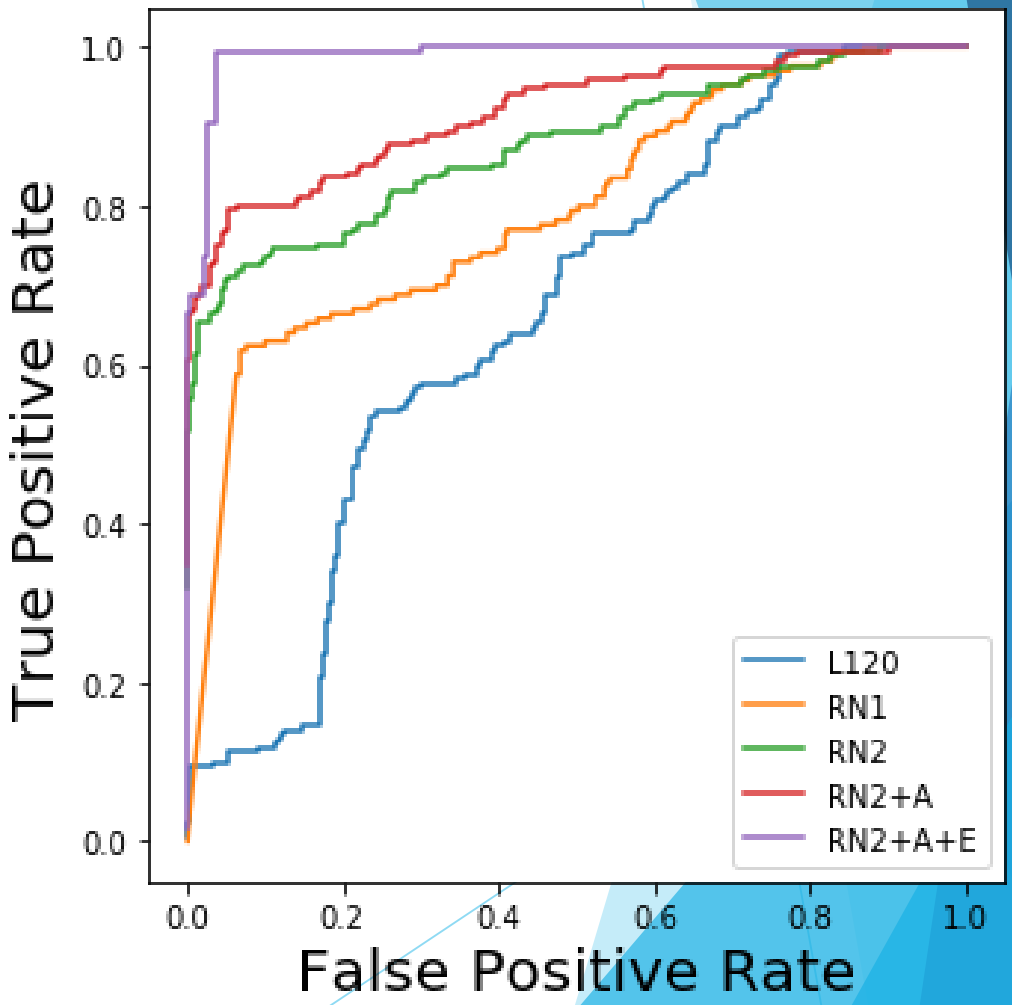

2012-06-30 9:03:10 PM M 1/5
24°C
RECONYX
SLT017


2012-06-30 9:03:11 PM M 2/5
24°C
RECONYX
SLT017


2012-06-30 9:03:12 PM M 3/5
24°C
RECONYX
SLT017

Ensembling over Sequences



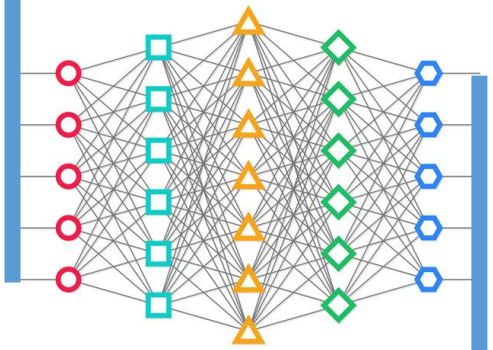
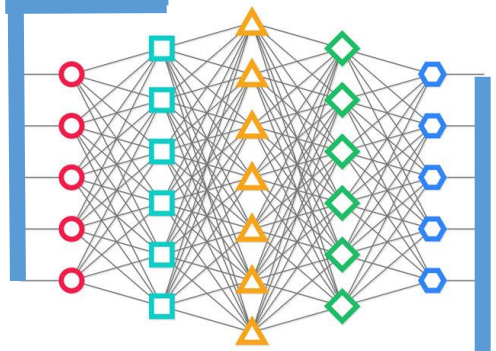





Time

1 2 3 4 5 6

CNTK Pipeline

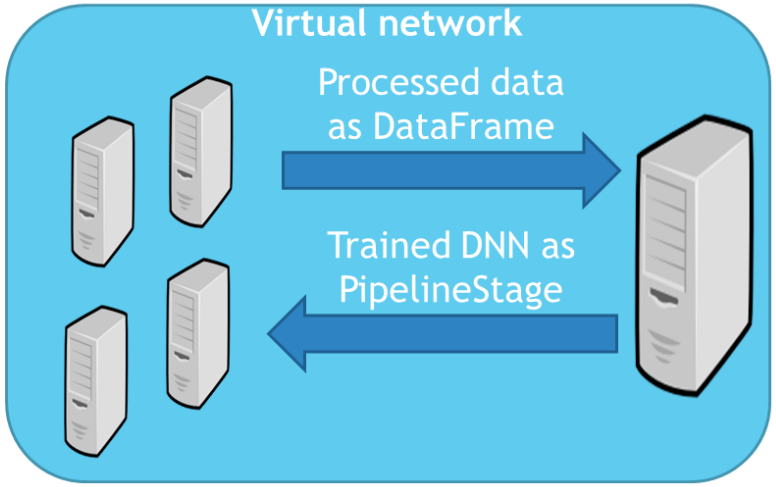



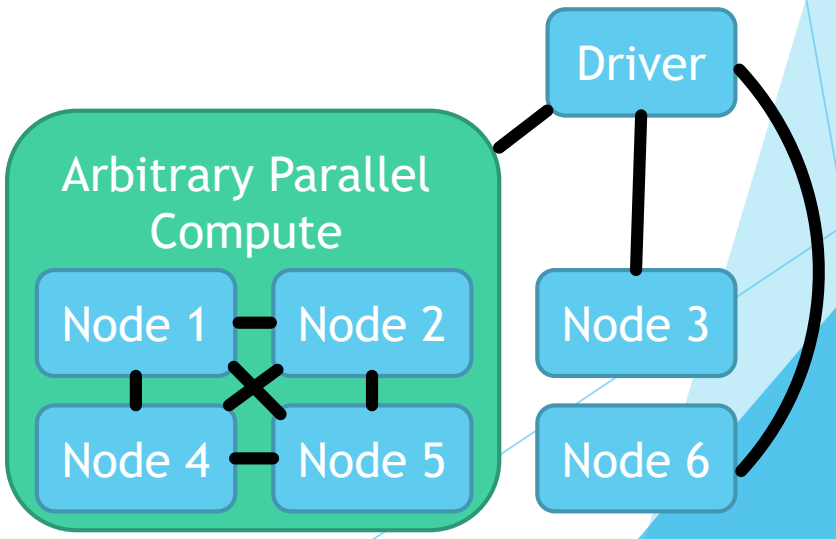
Time

1 2 3 4 5 6

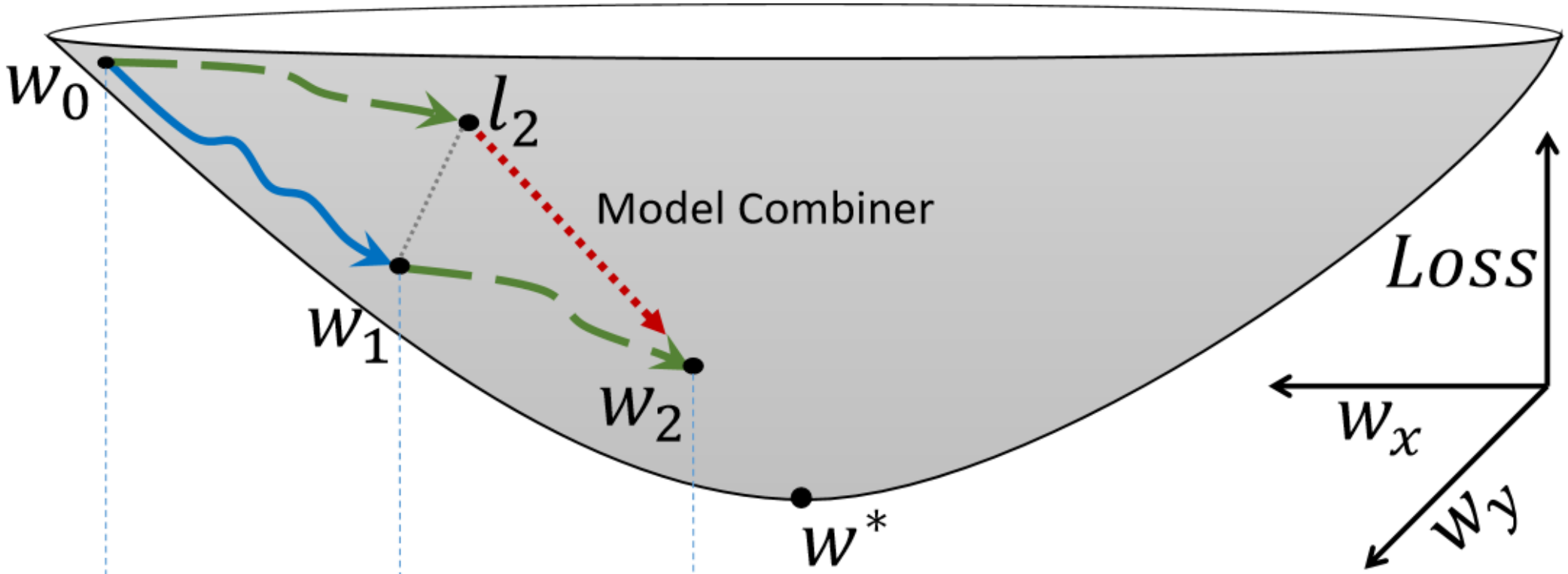
Training

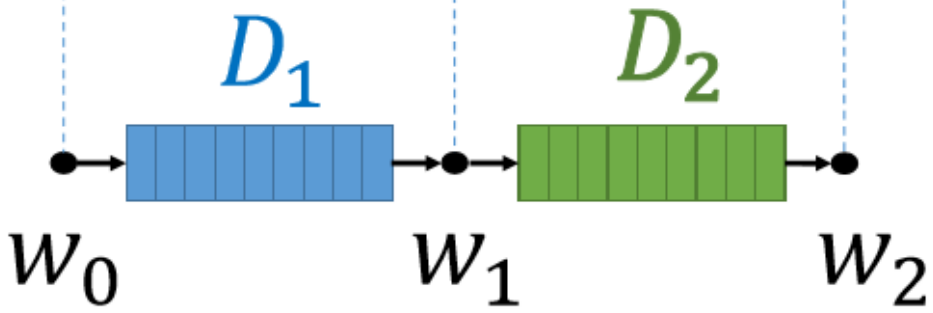
- https://github.com/Azure/mmlspark/tree/gpu
- GPU can give a 10x speedup for network training and eval
- Tether a GPU cluster to spark cluster
- Long Term Goal: Hook into Spark resource manager to request resources for custom jobs: CNTK, LightGBM, etc





Next Steps: Sym SGD and Parallel Training





Maleki, Saeed, Madanlal Musuvathi, and Todd Mytkowicz. "Parallel Stochastic Gradient Descent with Sound Combiners." *arXiv preprint arXiv:1705.08030* (2017).

How you can help

They need more camera surveys!

- 1,700 sq km surveyed of 1,500,000
- $500 will buy an additional camera
- $2,000 will fund a researcher
- Any amount helps!

www.snowleopard.org

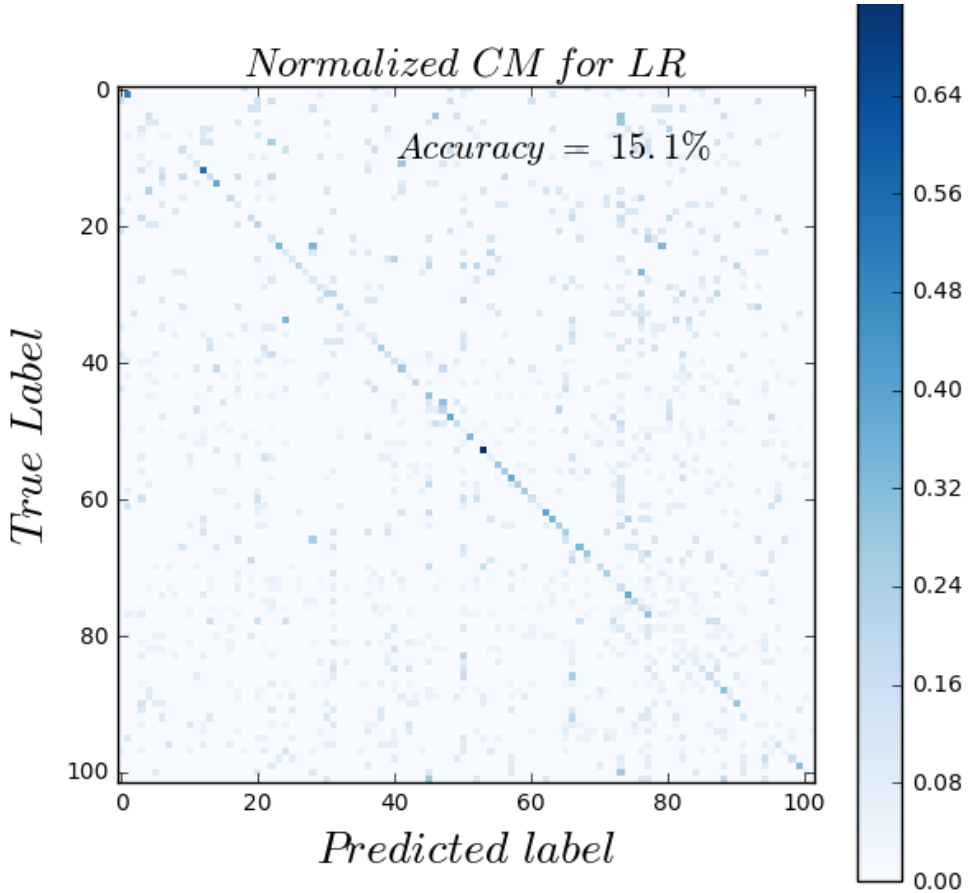
Thanks to

- MMLSpark Team:
  - Akshaya Annavajhala (AK), Danil Kirsanov, Eddie DeLeon, Eli Barzilay, Ilya Matiach, Joe Davison, Maureen Busch, Miruna Oprescu, Ratan Sur, Roope Astala, Sudarshan Raghunathan, Tong Wen, Young Park
- CNTK Team:
  - M. Hillebrand, N. Karampatziakis, W. Manousek, Z. Wang, C. Zhang, Liqun Fun

Where to Learn More

- Snow Leopard Blog Post:
  - https://blogs.technet.microsoft.com/machinelearning/2017/06/27/saving-snow-leopards-with-deep-learning-and-computer-vision-on-spark/
- MSJAR Article: “Massively Parallel Neural Networks with CNTK on Spark”
  - **http://aka.ms/msjar**
- Submitting to PMLR 2017: Flexible and Scalable Deep Learning with MMLSpark
- Sample Notebooks
  - https://github.com/mhamilton723/notebooks/blob/master/SnowLeopard.ipynb
  - https://github.com/Azure/mmlspark/blob/master/notebooks/samples/305%20-%20Flowers%20ImageFeaturizer.ipynb

Without Deep Featurization



With Deep Featurization