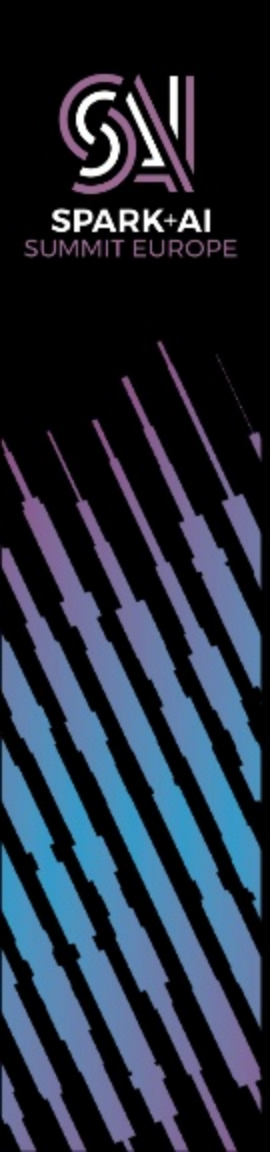


Unsupervised Object Detection using the Azure Cognitive Services on Spark

Mark Hamilton, Microsoft
Anand Raman, Microsoft

#SAISExp4


Snow
Leopard
Trust

BBC | Sign in | News | Sport | Weather | Shop | Earth | Travel

NEWS

Home | Video | World | US & Canada | UK | Business | Tech | Science | Stories

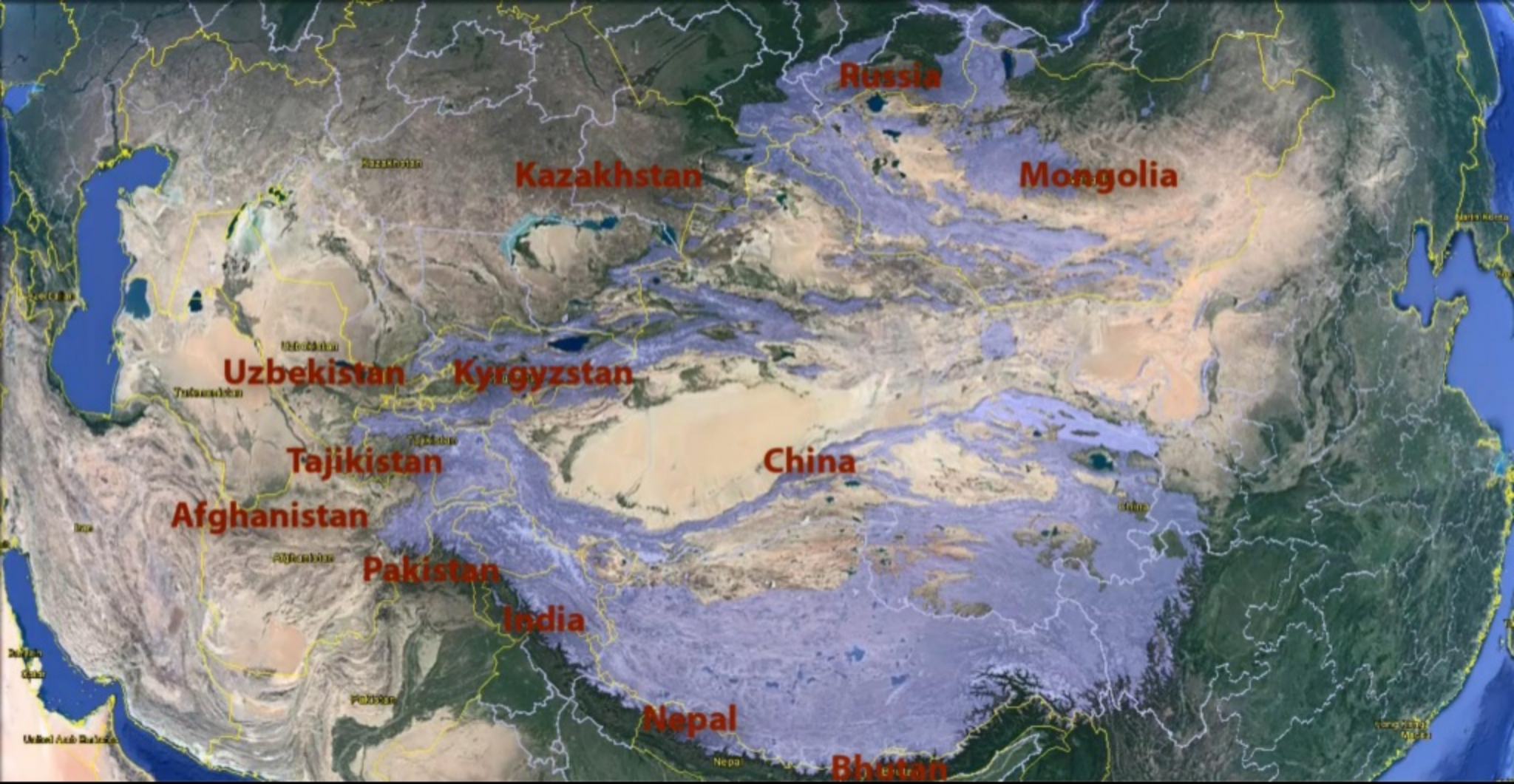
Asia | China | India

Snow leopard no longer 'endangered'

14 September 2017 Share

Statement on IUCN Red List Status Change of the Snow Leopard

The Snow Leopard Trust, one the leading conservation organizations working to protect this cat, opposes the IUCN's decision to change the snow leopard's Red List status from 'Endangered' to 'Vulnerable'.


Russia
Kazakhstan
Mongolia
Uzbekistan
Kyrgyzstan
Tajikistan
China
Afghanistan
Pakistan
India
Nepal
Bhutan

Snow Leopard Trust

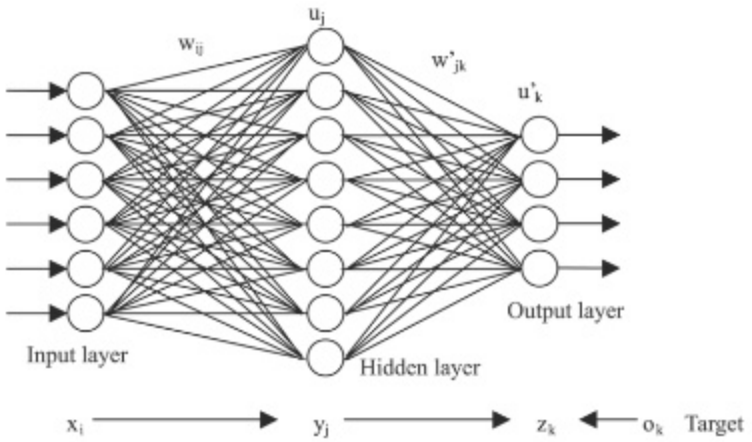
Classifying all 1.3 million images will take ***20k hours***

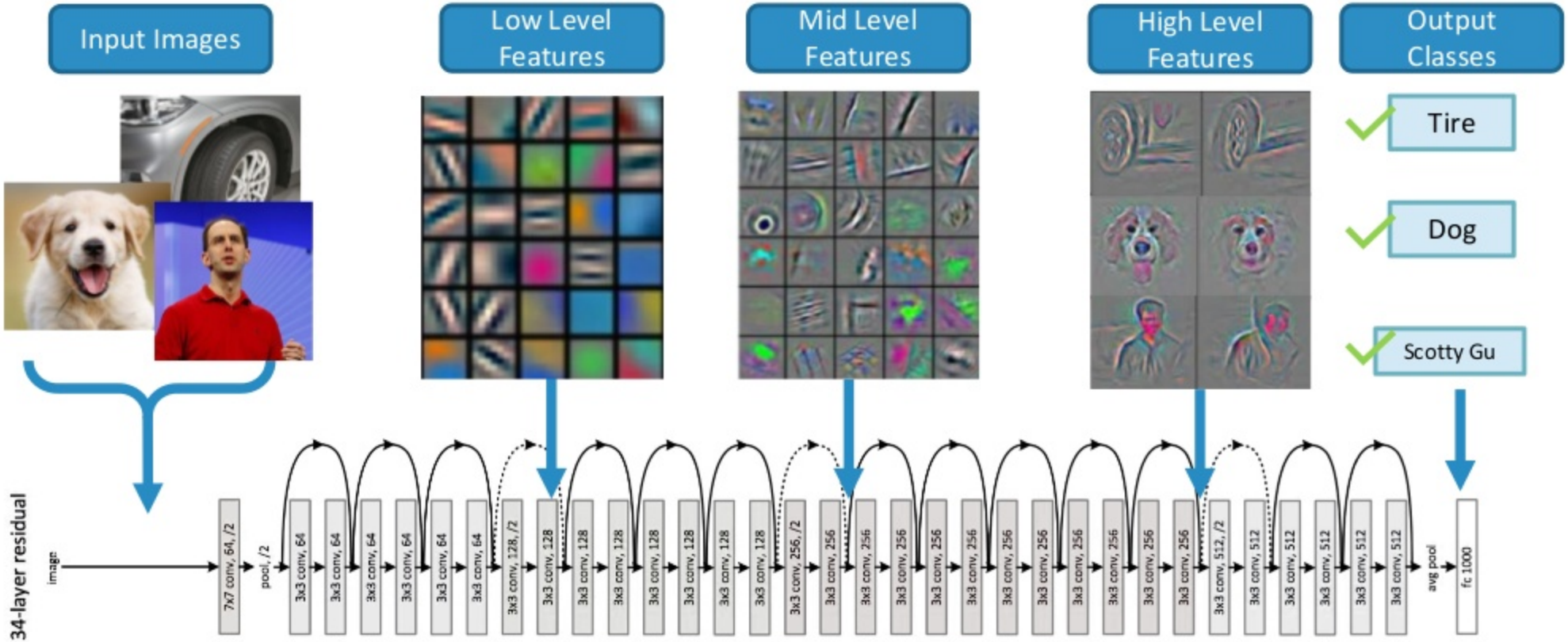




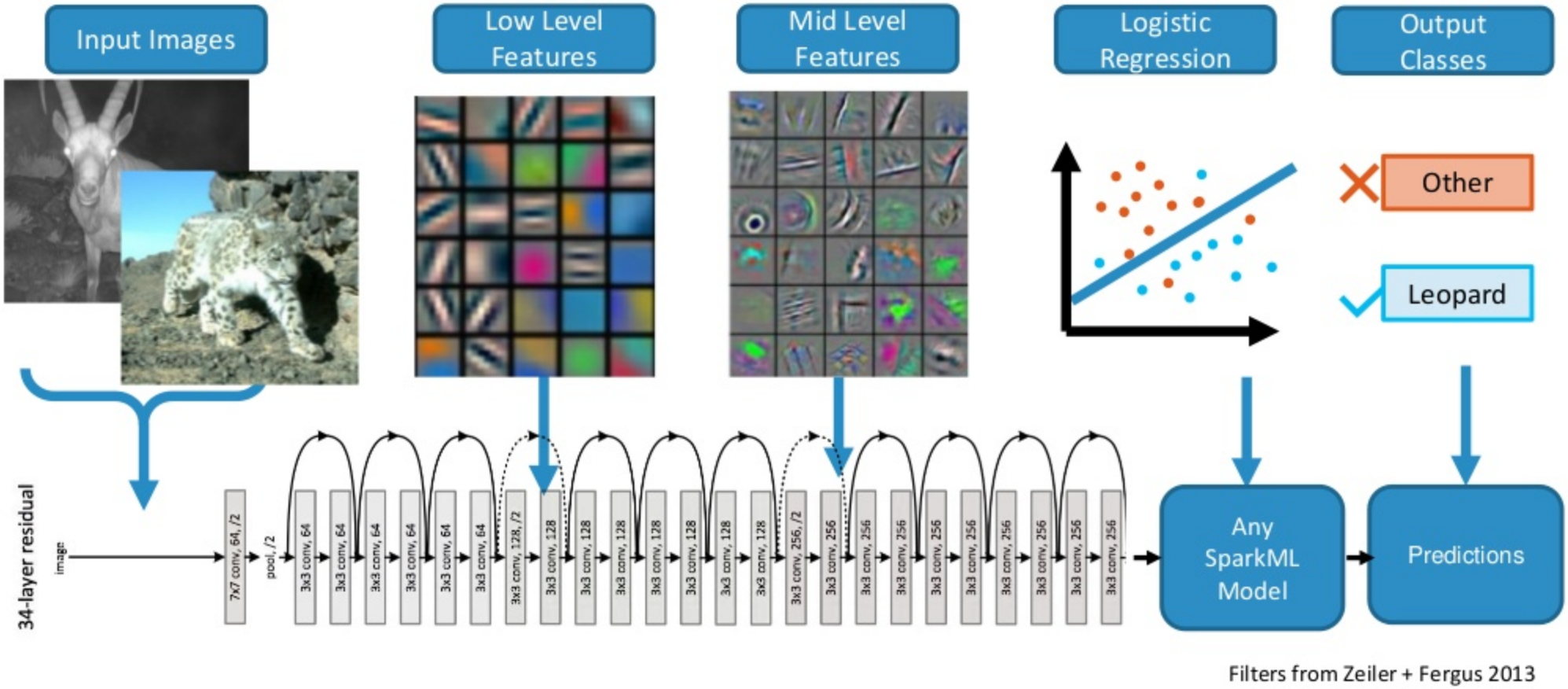

Deep Learning

- Brain-like algorithms trained with gradient descent
- Has become a recent favorite because:
  - Spectacular performance in many domains
  - Quick training
  - Low memory
  - Large space of possible model architectures
  - Automatic differentiation software makes it very easy





Filters from Zeiler + Fergus 2013

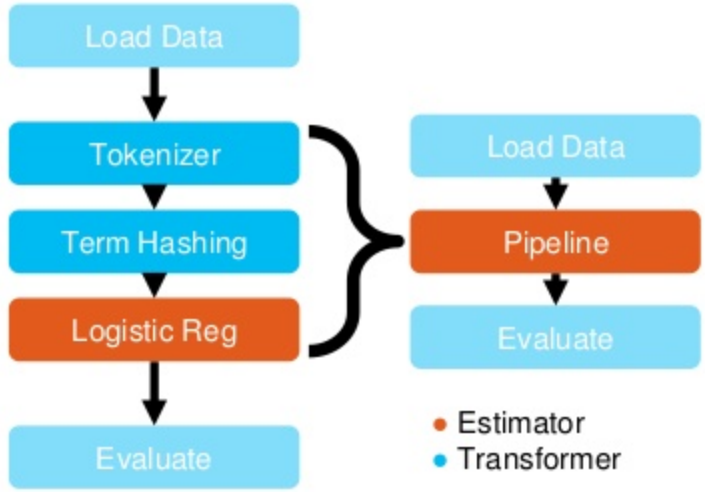


Filters from Zeiler + Fergus 2013

Apache Spark ML

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

- High level library for distributed machine learning
- More general than SciKit-Learn
- All models have a uniform interface
  - Can compose models into complex pipelines
  - Can save, load, and transport models

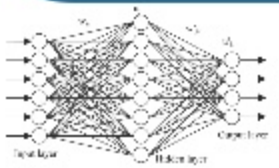


Distributed Computing

- Large scale parallelism
- Fault tolerance
- Auto-scaling / elasticity
- High throughput streaming
- Data source + orchestrator agnostic

Deep Learning

- GPU/CPU acceleration
- Automated gradient calculations
- Flexible language for defining huge space of models
- State of the art performance and accuracy
- ONNX Runtime



Microsoft ML for Apache Spark

www.aka.ms/spark

- **Open Source** Distributed ML library
- Unifies major computing paradigms and Microsoft tools into a single library and API
  - Python, Java, Scala, R
  - Batch, Streaming, Serving
  - Local, Static Cluster, Dynamic Cluster, Serverless
  - Databricks, HDI, AZTK, Kubernetes, …

CNTK on Spark

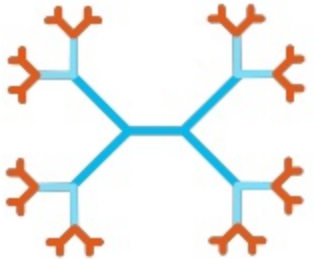
Scalable Deep Learning

Spark Serving

Stress Free Serving

HTTP on Spark

Distributed Microservices



LightGBM on Spark

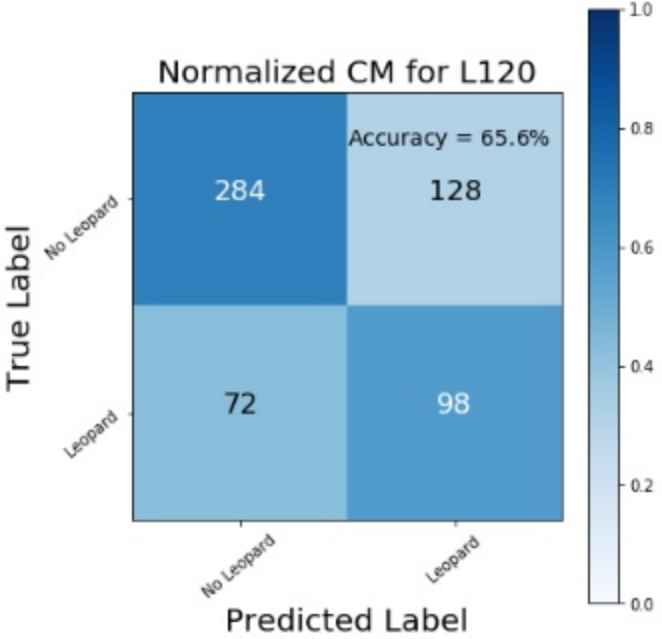
Fast Gradient Boosting

Training a Deep Transfer Learner:

```
1  import mmlspark as mml
2
3  network = mml.ModelDownloader(...).downloadByName("ResNet50")
4
5  model = Pipeline(stages = [
6      mml.ImageSetAugmenter(flipHorizontal=True),
7      mml.ImageFeaturizer(cutOutputLayers=2).setModel(network),
8      LogisticRegression()
9      mml.EnsembleByKey(keys=["group","location"], cols=["prob"])])
10
11 train, test = spark.read.parquet(...).split(...)
12
13 fit_model = model.fit(train)
14 results = fit_model.transform(test)
```

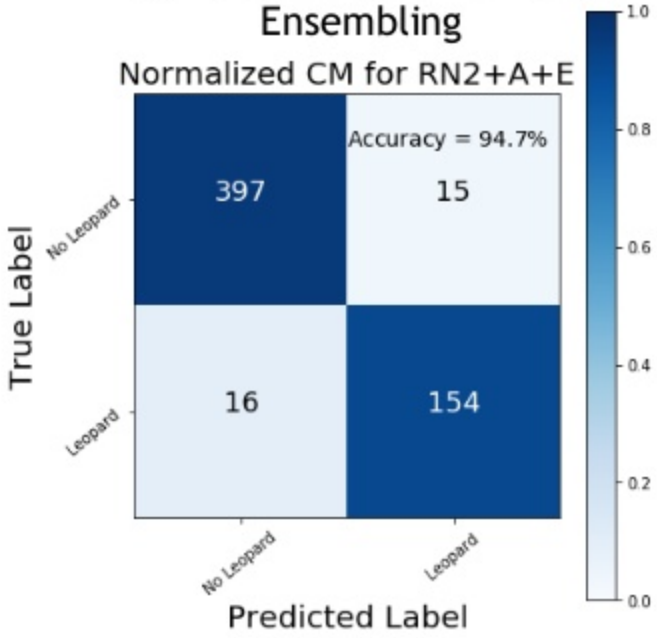
Performance

Without Deep Featurization

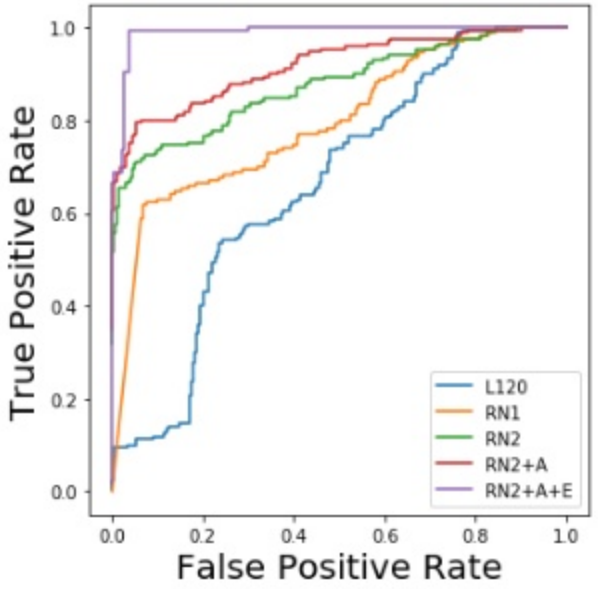


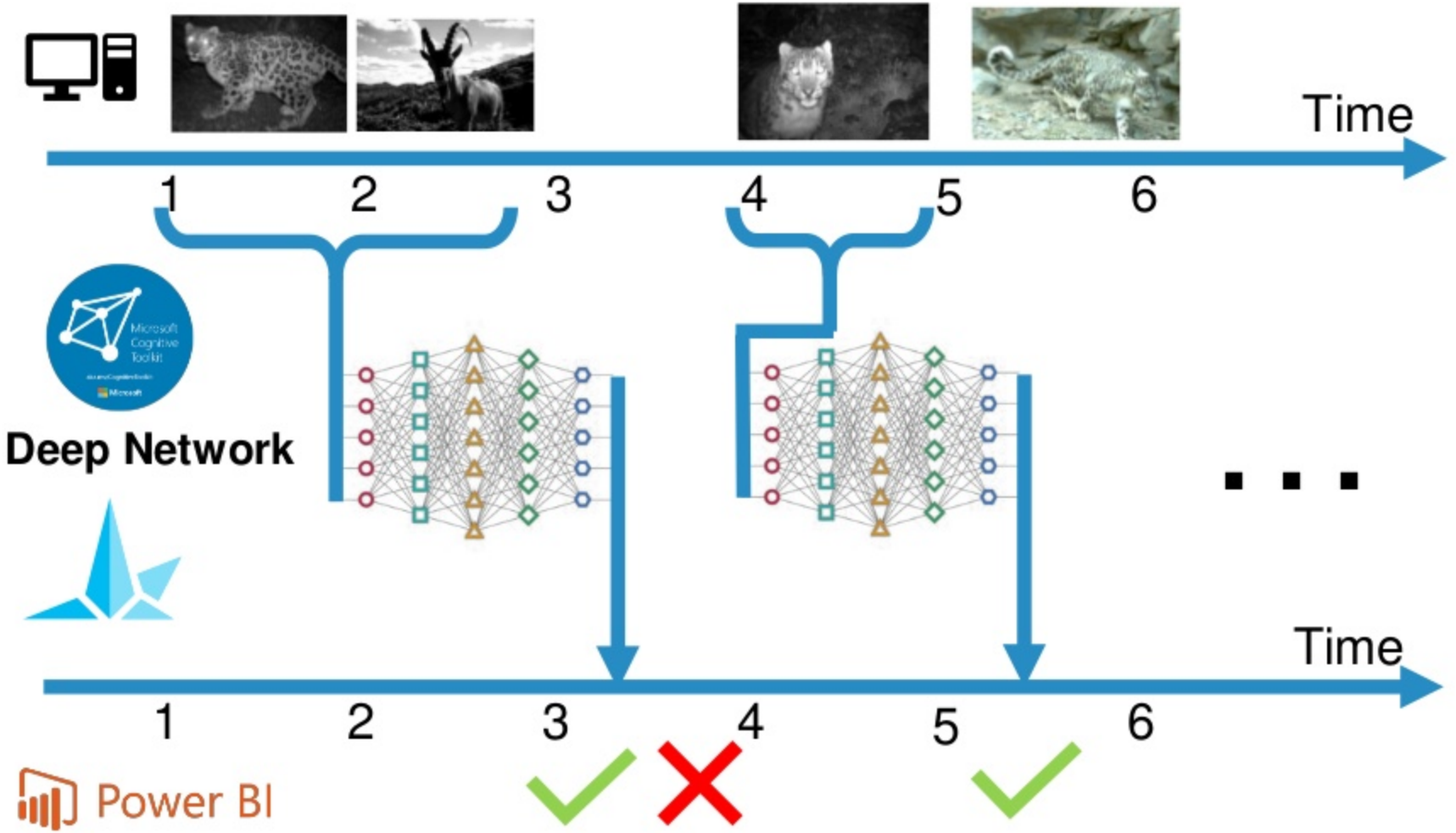
Accuracy 65.6%

With Deep Featurization, Augmentation, and Temporal Ensembling

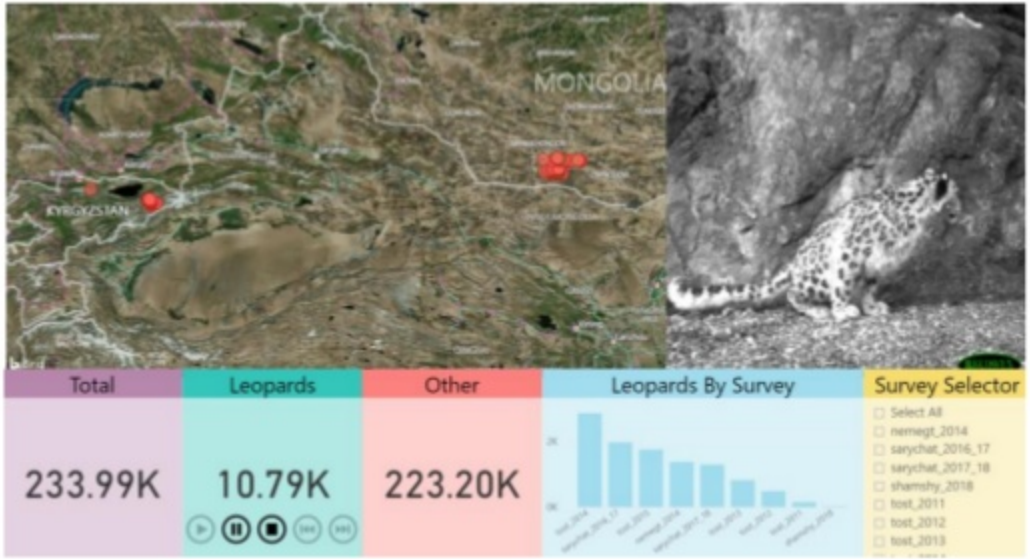


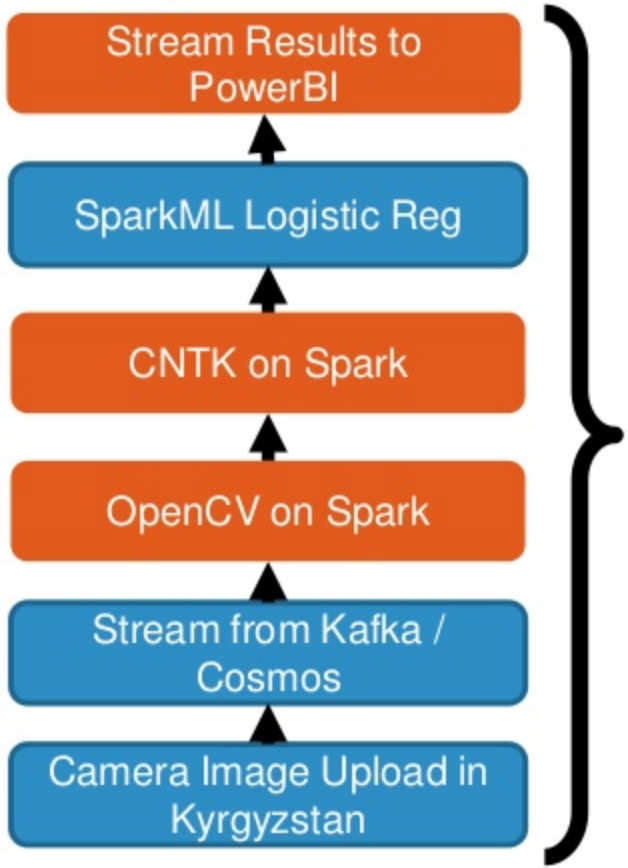
Accuracy 94.7%




Time
1
2
3
4
5
6
Deep Network
Time
1
2
3
4
5
6
Power BI

Real Time monitoring with PowerBI





27 Lines of Code

~1k imgs/s throughput on CPU

~3 second latencies

MMLSpark Contribution

Simple APIs matter

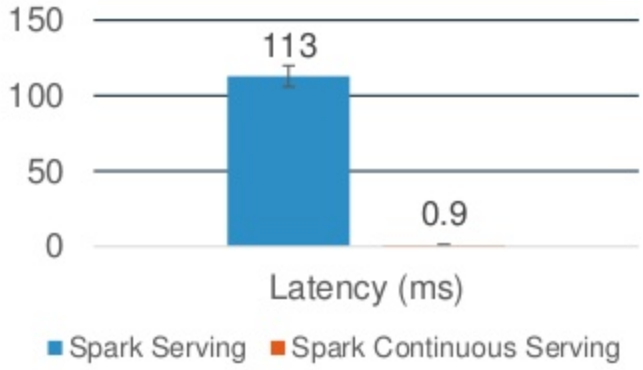
```
16  streaming_df = spark.readStream.file(...)
17  fit_model.transform(streaming_df)
18  stream = PowerBIWriter.stream(results, powerbi_url).start()
```

Apache Spark Serving

Lightning Fast Web Services on ***Any*** Spark Cluster

- Sub-millisecond latencies
- Fully Distributed
- Spins up in seconds
- Same API as Batch and Streaming
- Scala, Python, R and Java
- Fully Open Source

Announcing: **100x Latency Reduction** with MMLSpark v0.14



www.aka.ms/spark

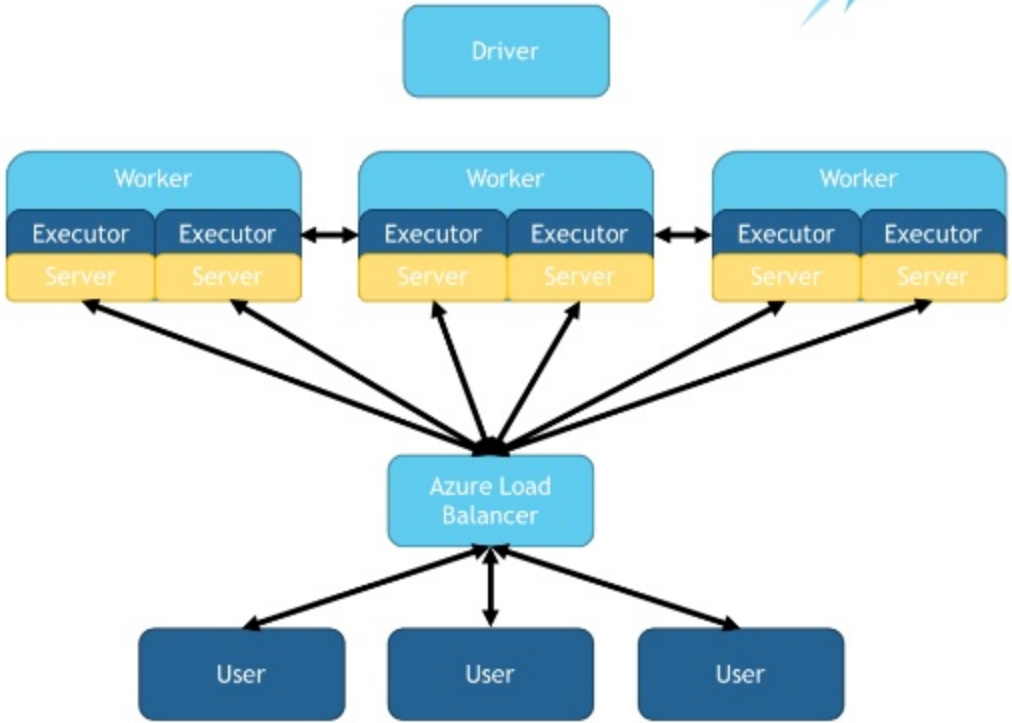
JIRA: SPARK-25350

Web Serving with the Same API:

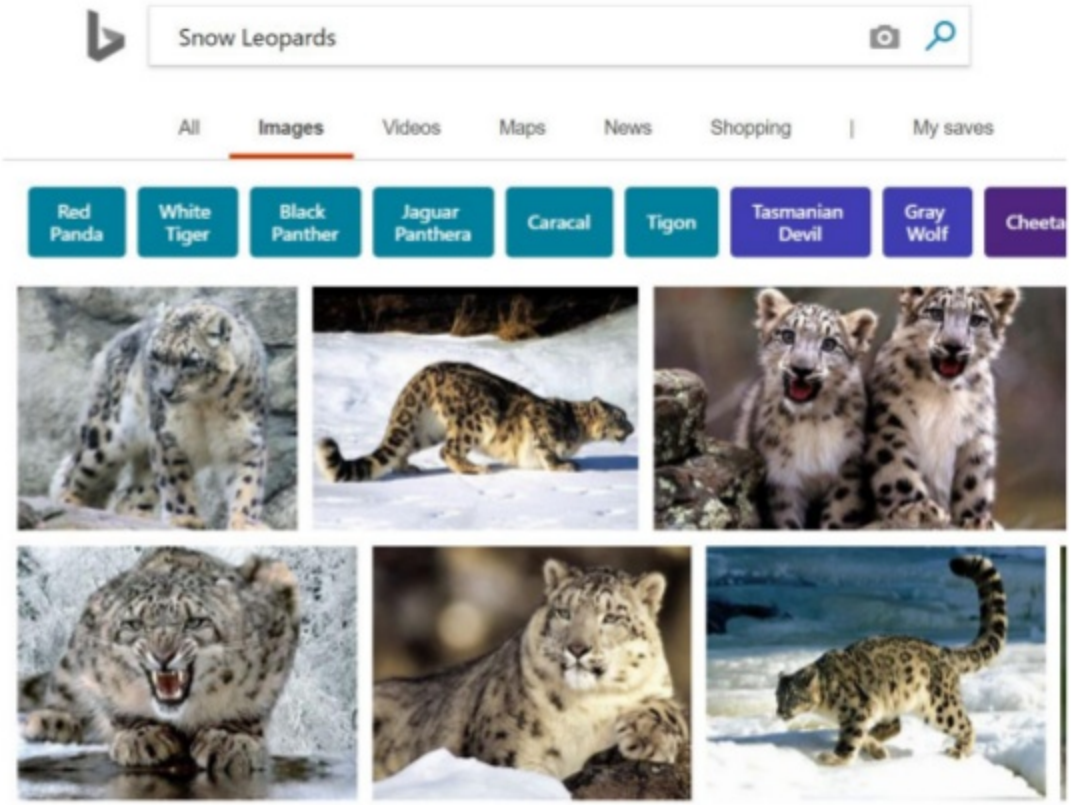
```
4  serving_df = spark.readStream.server()
5    .address("0.0.0.0", 8888, "leopard_detector")
6    .load()
7    .parseRequest(BinaryType())
8
9  results = fitModel.transform(serving_df)
10
11 results.makeReply("probability")
12   .writeStream
13   .server()
14   .replyTo("leopard_detector")
15   .start()
16
```

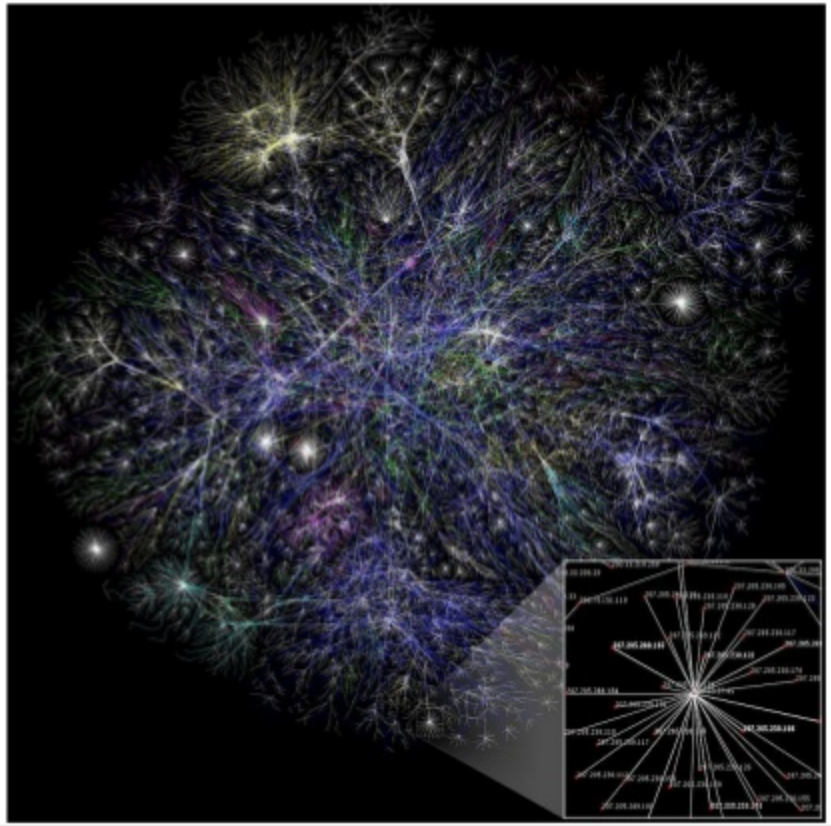
Spark Serving Architecture

- 3 main modes:
  - server: 1 service on the head node
  - distributedServer: 1 service per executor
  - continuousServer: 1 service per partition
- Built on top of Spark Streaming
- Each worker keeps a running service, and a routing table
- Use HTTP on Spark objects to represent requests and responses in Spark SQL

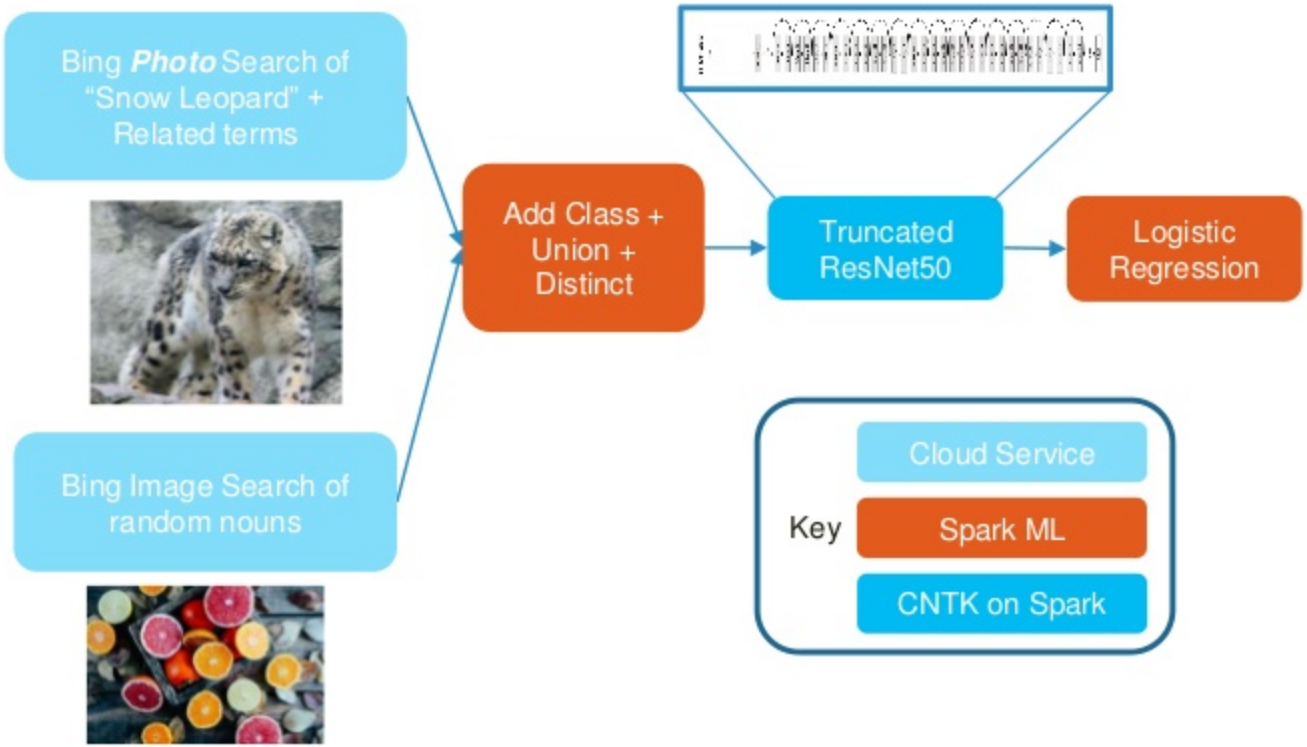


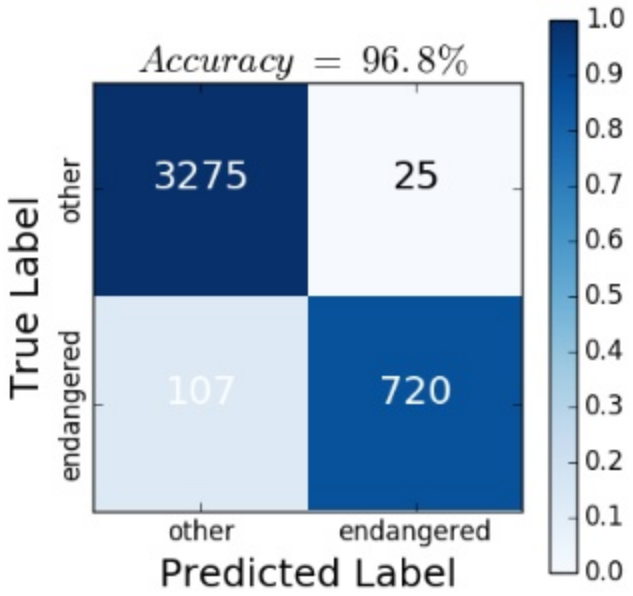
Going Further: No Labels, No Problem





Bing Powered Snow Leopard Detection Pipeline





Bing Image Search API

```
endangeredLinks =  requestParameters \
  .transform(BingImageSearch()                          # Apply Bing Image Search
    .setSubscriptionKey(os.environ['API_KEY'])
    .setOffsetCol("offsets")
    .setQueryCol("queries")
    .setCount(10)
    .setImageType("photo")
    .setOutputCol("images"))\
  .transform(GetBingURLs(inputCol="images", outputCol="urls")) \
  .withColumn("labels", lit("endangered"))          # Add the labels
```

```
randomLinks = spark.read.table("random_words") \
  .transform(BingImageSearch()
    .setSubscriptionKey(os.environ['API_KEY'])
    .setCount(10)
    .setQueryCol("words")
    .setOutputCol("images")) \
  .transform(GetBingURLs(inputCol="images", outputCol="urls")) \
  .withColumn("label", lit("other"))
```

urls	labels
	endangered
	endangered
	endangered
	endangered

urls	label
	other
	other
	other
	other

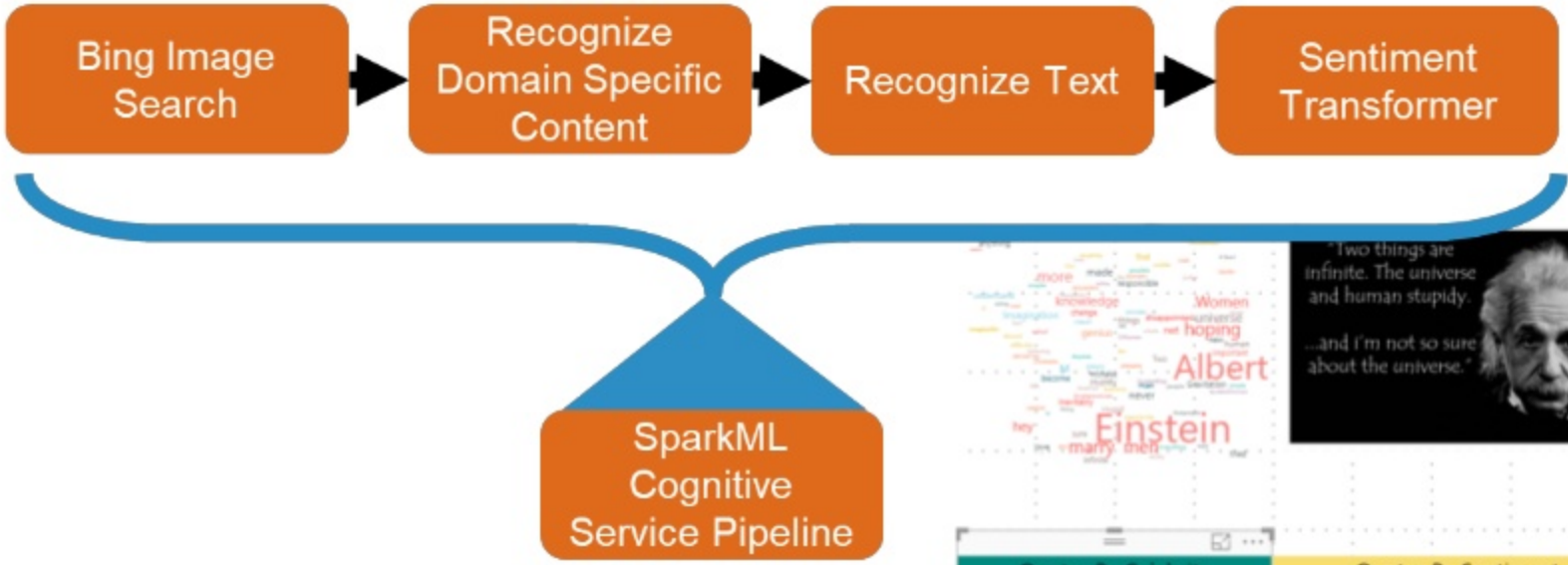
Microsoft
Cognitive Services

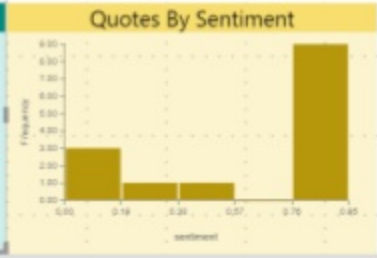
+

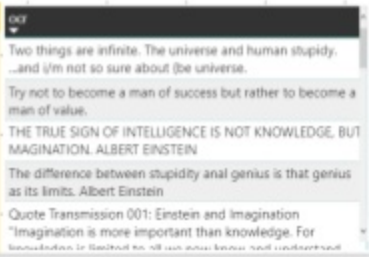
Announcing: Intelligent and Distributed Microservices in SparkML

www.aka.ms/spark

Celebrity Quote Analysis

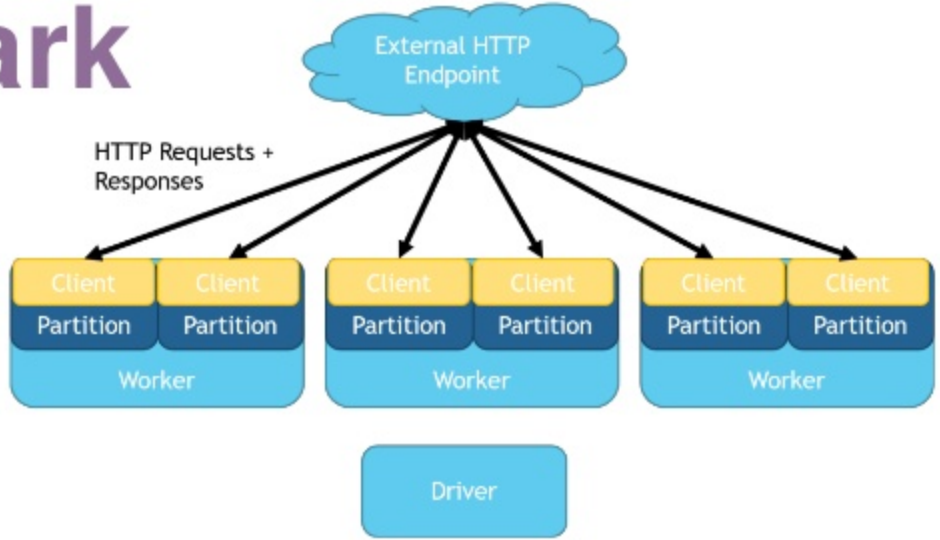






HTTP on Spark

- Allows Spark users to call into ***any*** HTTP service
- Leverage parallel networking of cluster
- Allows Spark to work with microservice architectures
- Mini-batching, asynchrony, and buffering supported
- Statically typed and usable from Spark, PySpark, and SparklyR




```
replies = SimpleHTTPTransformer()
    .setOutputParser(new JSONOutputParser().setDataType(...)) \
    .setUrl("http://my_favorite_webservice.com") \
    .setInputCol("data") \
    .setOutputCol("results") \
    .setConcurrency(3) \
    .transform(df)
```

Model Interpretability with LIME

(a) Original Image (b) Explaining *Electric guitar* (c) Explaining *Acoustic guitar* (d) Explaining *Labrador*

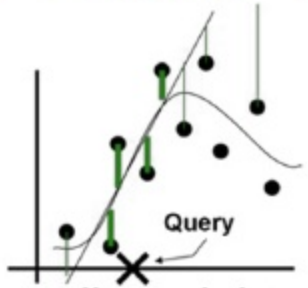
Figure 4: Explaining an image classification prediction made by Google's Inception network, highlighting positive pixels. The top 3 classes predicted are "Electric Guitar" ($p = 0.32$), "Acoustic guitar" ($p = 0.24$) and "Labrador" ($p = 0.21$)

Source: Marco Tulio Ribeiro et al.

Model Interpretability with LIME

Original Image
P(tree frog) = 0.54

Perturbed Instances	P(tree frog)
	0.85
	0.00001
	0.52



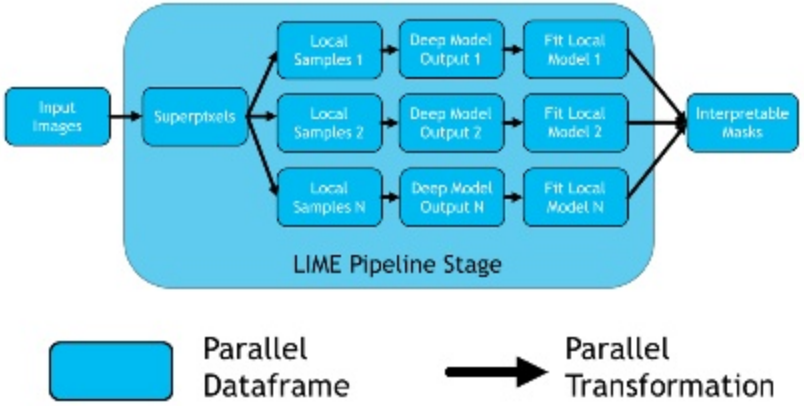
Locally weighted regression

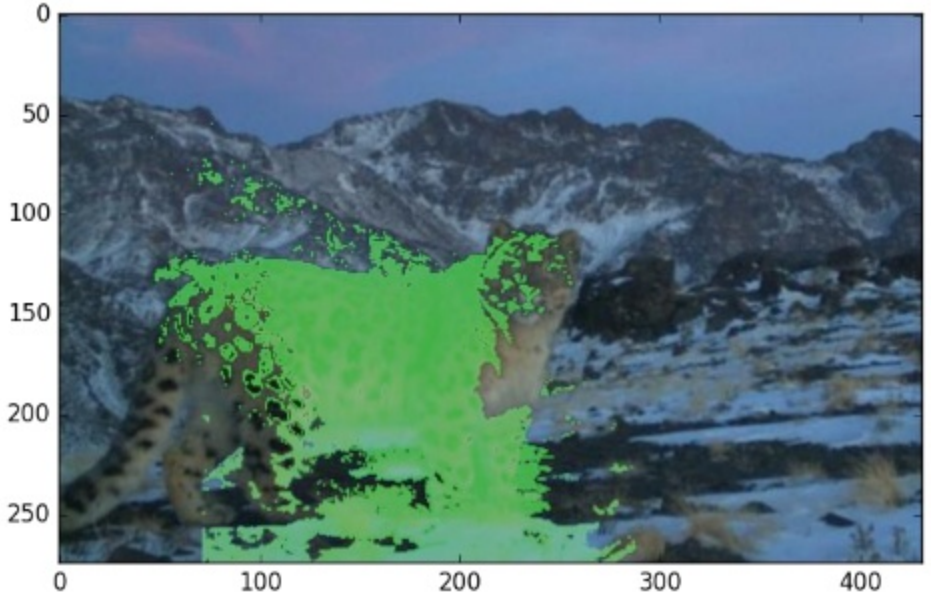
Explanation

Source: Marco Tulio Ribeiro et al.

Announcing: LIME on Spark

- Elastic and Distributed
- Works with ***any*** image classification model
- Example notebook instructions at:
  - www.aka.ms/spark





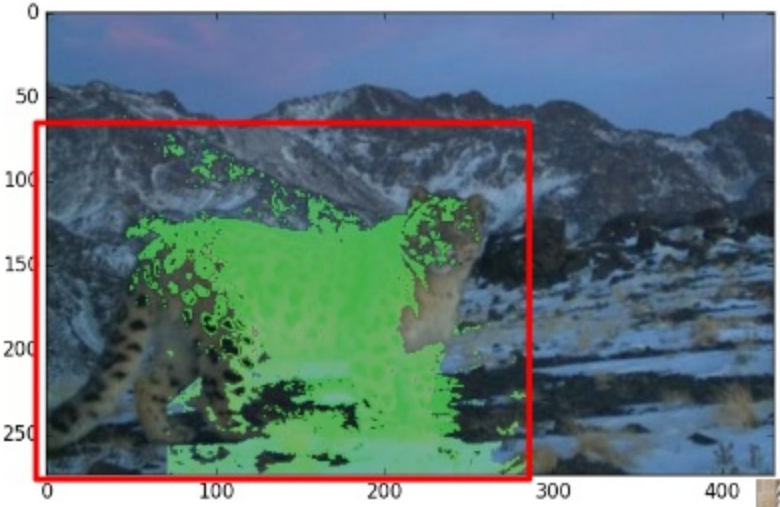
Using LIME to generate bounding boxes

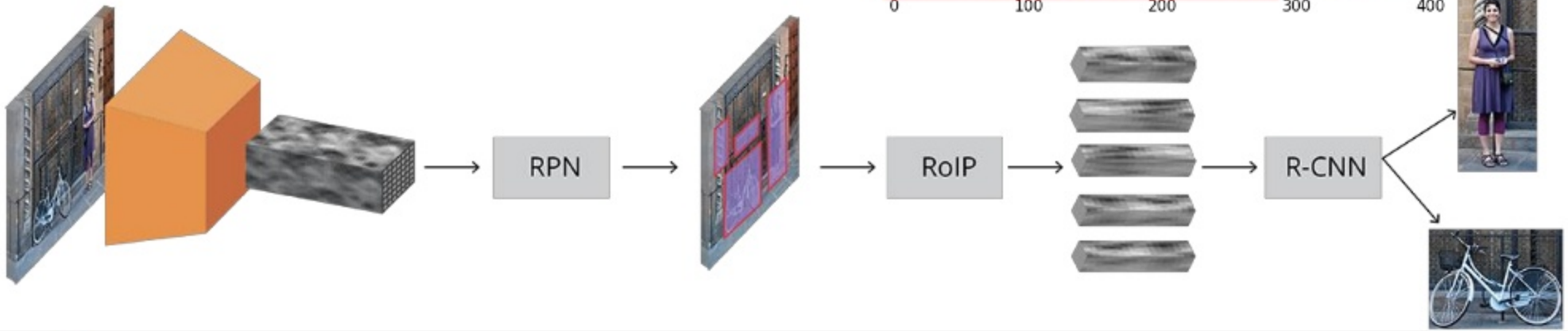
Transferring LIME's Knowledge to FasterRCNN

$$mask = union(superpixels \mid top\ 70\%\ of\ weights)$$

$$x1, y1 = \min(mask_x), \min(mask_y)$$

$$x2, y2 = \max(mask_x), \max(mask_y)$$

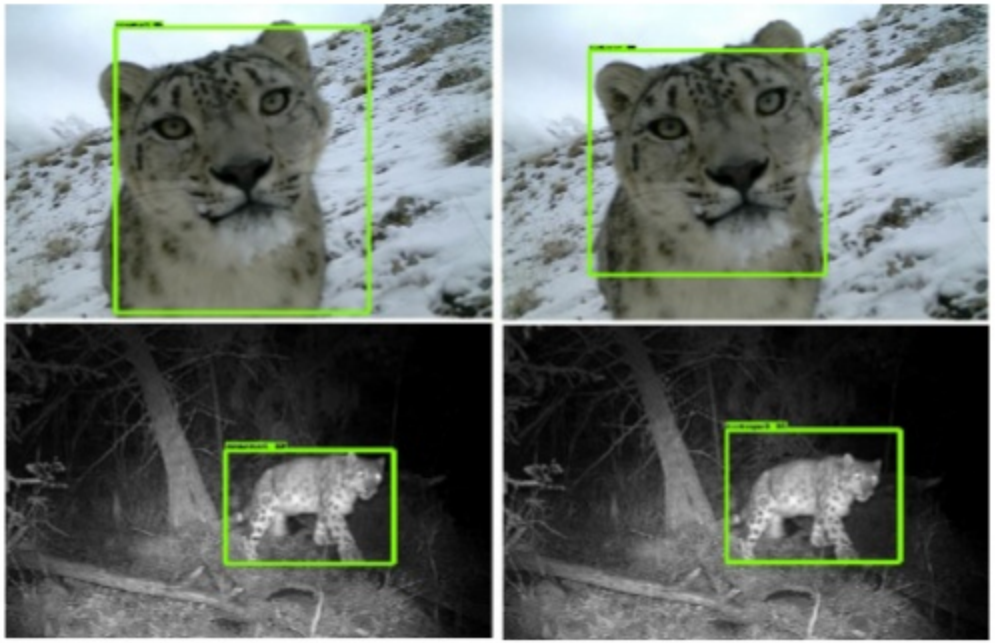




Results

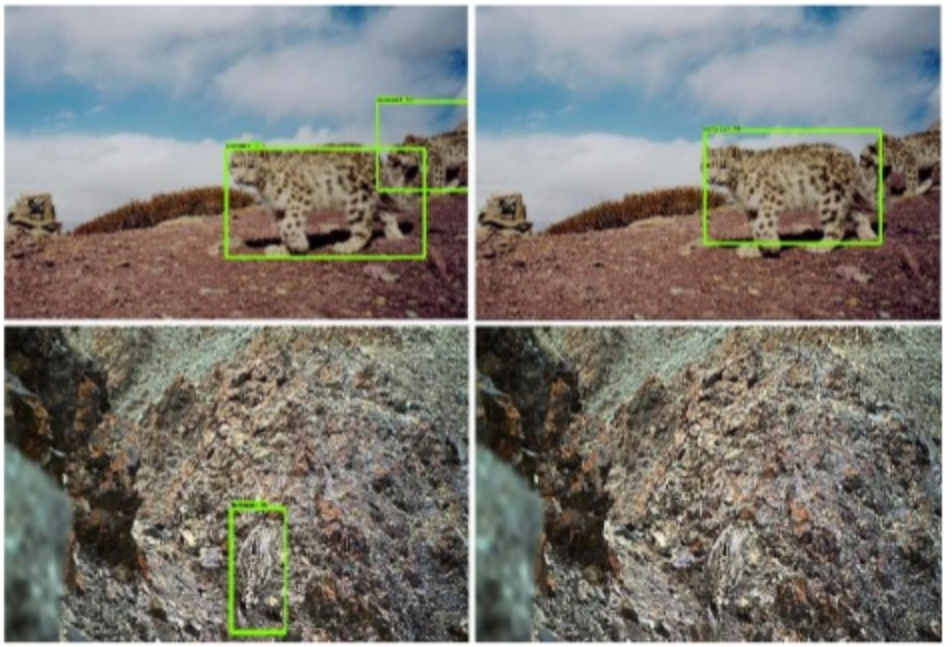
Human Labels

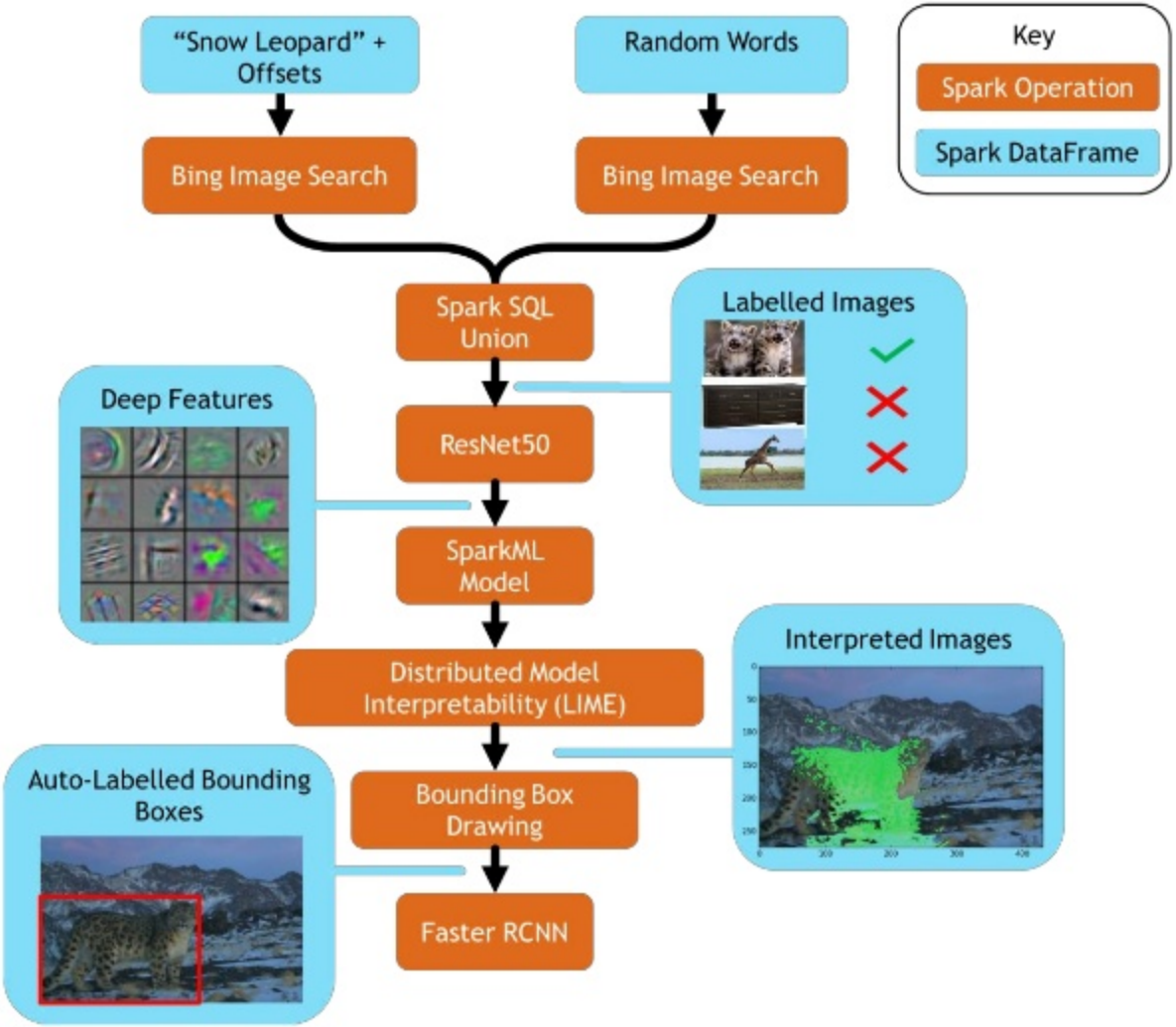
Unsupervised FRCNN Outputs



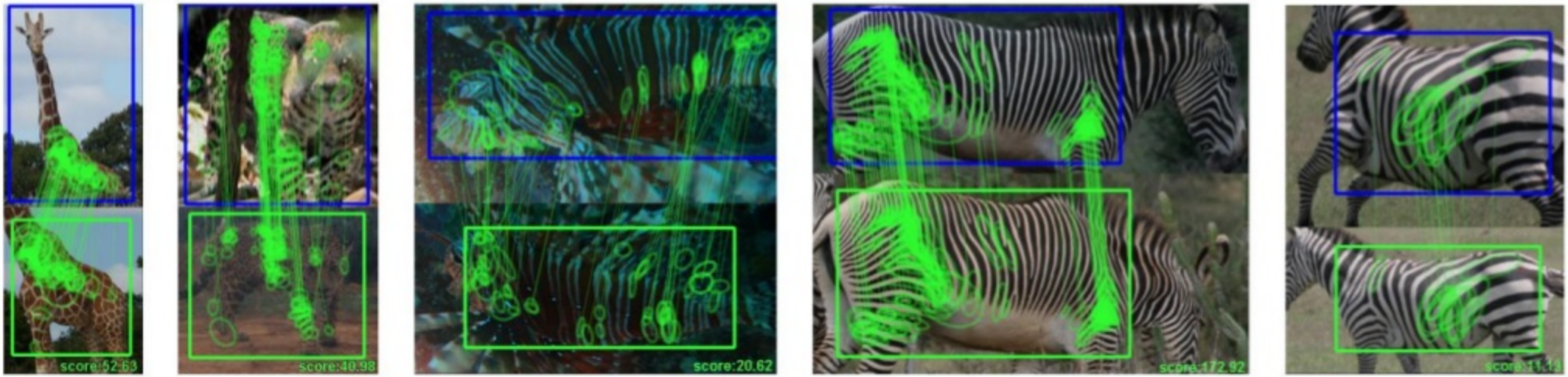
Human Labels

Unsupervised FRCNN Outputs




"Snow Leopard" + Offsets
Random Words
Key
Spark Operation
Spark DataFrame
Bing Image Search
Bing Image Search
Spark SQL Union
Labelled Images
Deep Features
ResNet50
SparkML Model
Distributed Model Interpretability (LIME)
Interpreted Images
Auto-Labelled Bounding Boxes
Bounding Box Drawing
Faster RCNN

Next Steps: Hotspotter



Source: HotSpotter - Patterned Species Instance Recognition

In Conclusion

- Big Releases in v0.14:
  - **Cognitive Services on Spark**
  - **Sub-Millisecond latency distributed web services**
  - **Distributed Model Interpretability**
- Easy batch, streaming apps, PowerBI dashboards, or RESTful web services
- We aim to give all work back to the community!
- Easy to Get Started on Databricks:
  - 16 Jupiter notebook guide

www.aka.ms/spark

Contributions Welcome!
github.com/Azure/mmlspark

Thanks to

- You all!
- **Microsoft AI Development Acceleration Program: Abhiram Eswaran, Ari Green, Courtney Cochrane, Janhavi Suresh Mahajan, Karthik Rajendran, Minsoo Thigpen, Casey Hong, Soundar Srinivasan**
- MMLSpark Team: Sudarshan Raghunathan, Ilya Matiach, Eli Barzilay, Tong Wen, Ben Brodsky
- The Snow Leopard Trust: Rhetick Sengupta, Koustubh Sharma, Jeff Brown, Michael Despines
- Microsoft: Joseph Sirosh, Lucas Joppa, WeeHyong Tok

MMLSpark Website: aka.ms/spark

Get in touch: marhamil@microsoft.com