CLASSIFYING IMAGES IN SPARK USING BIGDL

Maurice Nsabimana,
World Bank Development Data Group

Yulia Tell,
Big Data Technologies, Intel Corporation

AGENDA

- Introduction
- BigDL Overview and Key Features
- Solution Architecture
- Use Case: Problem Statement
- Use Case: Dataset, Model Development and Training
- Use Case: Current Results and Next Steps
- Summary

INTRODUCTION

AI AND ANALYTICS OPPORTUNITIES IN EVERY INDUSTRY

CONSUMER

HEALTH

FINANCE

RETAIL

ENERGY

TRANSPORT

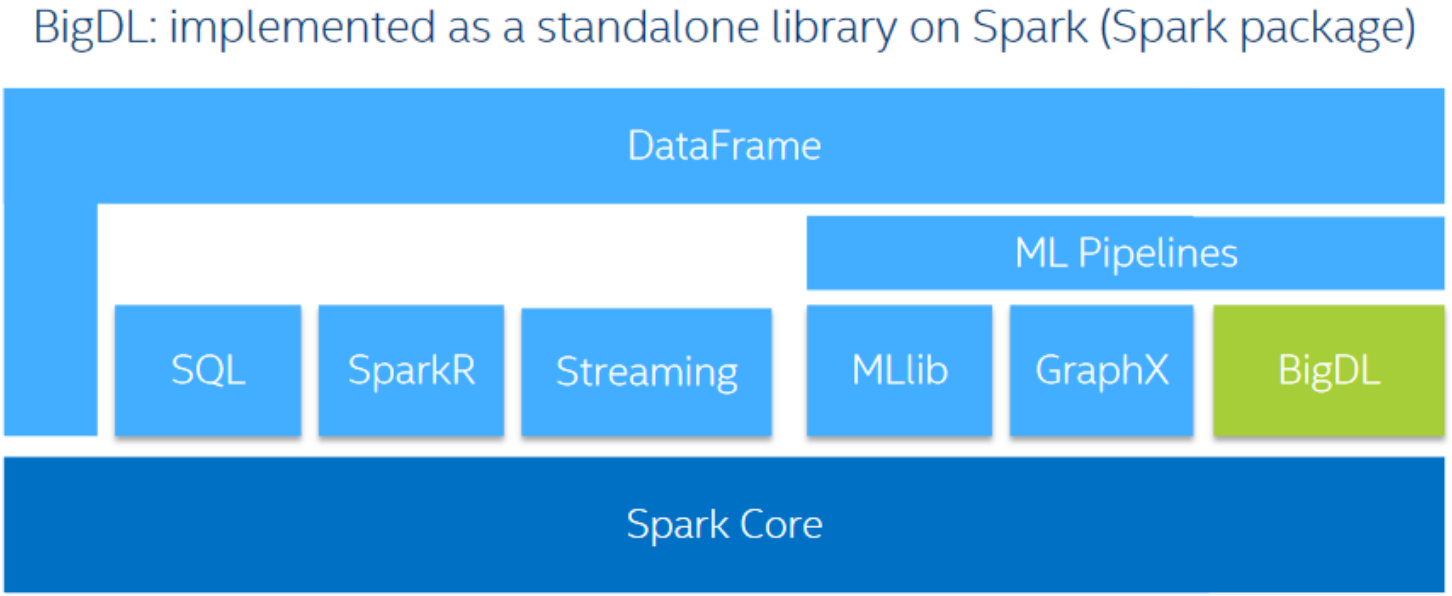
INDUSTRIAL

ACCELERATING BUSINESS GAINS AND COMPETITIVE ADVANTAGE

BIGDL OVERVIEW

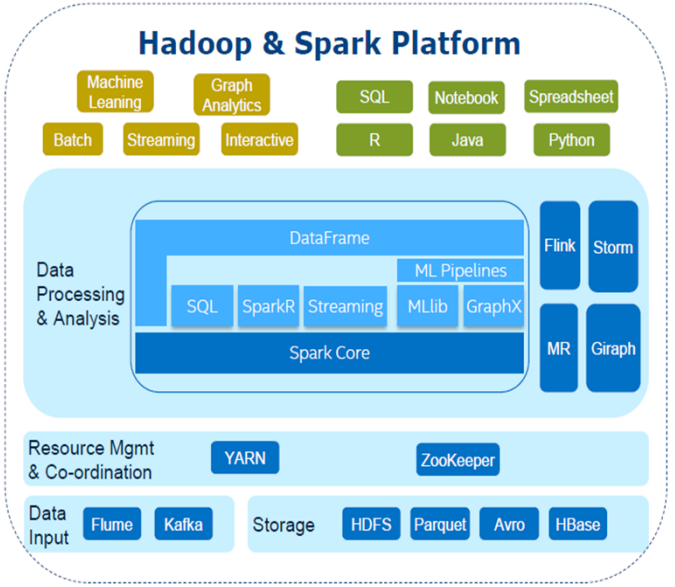
WHAT IS BIGDL?

BigDL is a distributed deep learning library for Apache Spark*



BIGDL IS DESIGNED FOR BIG DATA

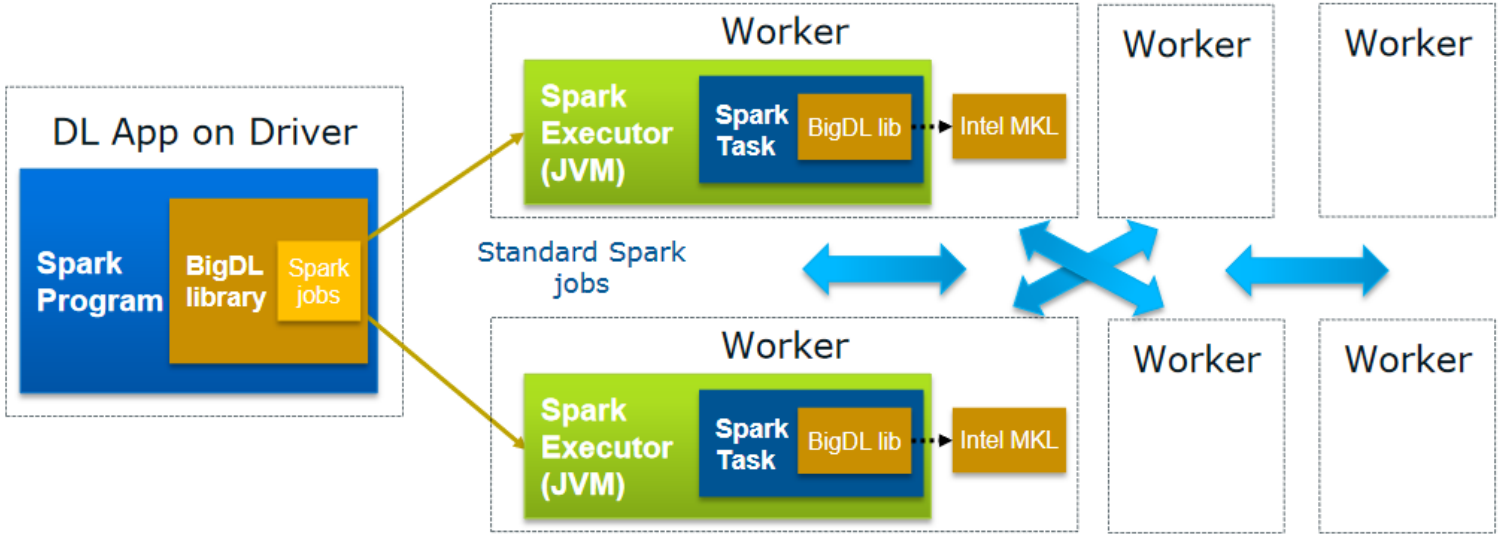
- Distributed deep learning framework for Apache Spark*
- Make deep learning more accessible to big data users and data scientists
  - Write deep learning applications as ***standard Spark programs***
  - Run on existing Spark/Hadoop clusters (***no changes needed***)
- Feature parity with popular deep learning frameworks
  - E.g., Caffe, Torch, Tensorflow, etc.
- High performance
  - Powered by Intel MKL and multi-threaded programming
- Efficient scale-out
  - Leveraging Spark for distributed training & inference



BIGDL AS A STANDARD SPARK PROGRAM

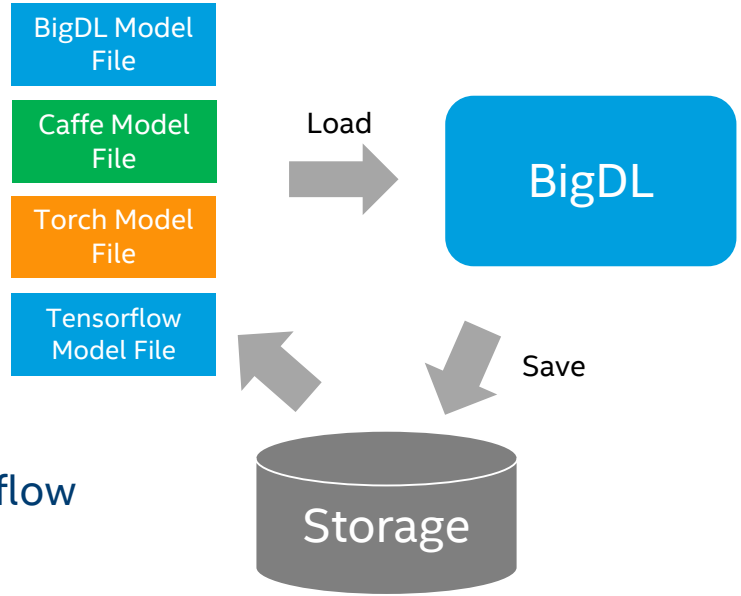
Distributed Deep learning applications (training, fine-tuning & prediction) on Apache Spark*

- No changes to the existing Hadoop/Spark clusters needed



MODELS INTEROPERABILITY SUPPORT

- Model Snapshot
  - Long training work checkpoint
  - Model deployment and sharing
  - Fine-tune
- Caffe/Torch/Tensorflow Model Support
  - Model file load
  - Easy to migrate your Caffe/Torch/Tensorflow work to Spark



- **NEW** - BigDL supports loading pre-defined Keras models (Keras 1.2.2)

BIGDL: PYTHON API

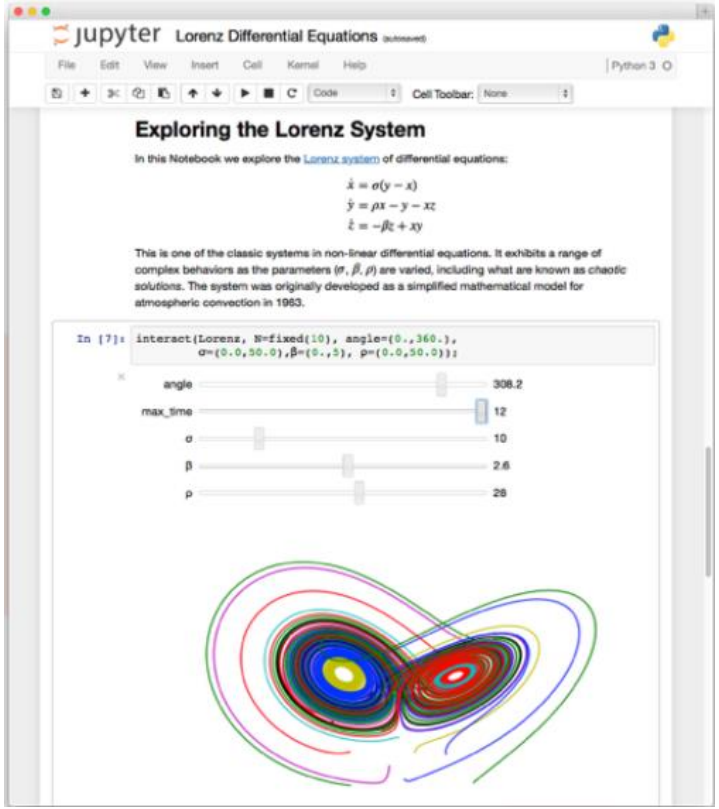
- Support deep learning model training, evaluation, inference
- Support Spark v1.5/1.6/2.X
- Support **Python 2.7/3.5/3.6**
- Based on PySpark, **Python API** in BigDL allows use of existing Python libs (Numpy, Scipy, Pandas, Scikit-learn, NLTK, Matplotlib, etc)

```
train_data = get_minst("train").map(
    normalizer(mnist.TRAIN_MEAN, mnist.TRAIN_STD))
test_data = get_minst("test").map(
    normalizer(mnist.TEST_MEAN, mnist.TEST_STD))
state = {"batchSize": int(options.batchSize),
         "learningRate": 0.01,
         "learningRateDecay": 0.0002}
optimizer = Optimizer(
    model=build_model(10),
    training_rdd=train_data,
    criterion=ClassNLLCriterion(),
    optim_method="SGD",
    state=state,
    end_trigger=MaxEpoch(100))
optimizer.setvalidation(
    batch_size=32,
    val_rdd=test_data,
    trigger=EveryEpoch(),
    val_method=["top1"]
)
optimizer.setcheckpoint(EveryEpoch(), "/tmp/lenet5/")
trained_model = optimizer.optimize()
```

WORKS WITH NOTEBOOK

Running BigDL applications directly in Jupyter, Zeppelin, Databricks notebooks, etc.

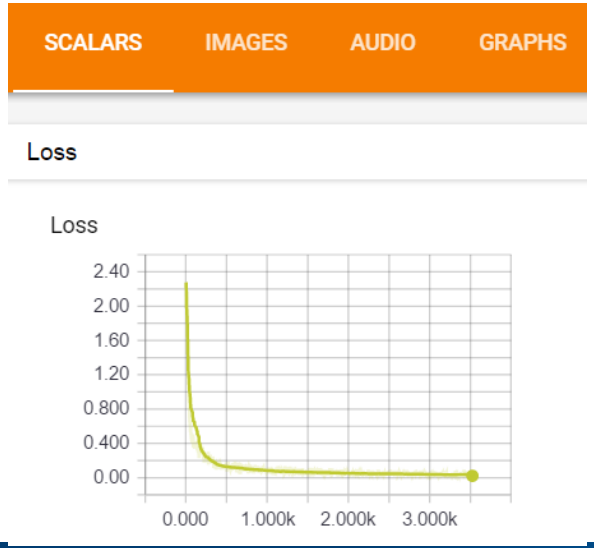
- ✓ Share and Reproduce
  - Notebooks can be shared with others
  - Easy to reproduce and track
- ✓ Rich Content
  - Texts, images, videos, LaTeX and JavaScript
  - Code can also produce rich contents
- ✓ Rich toolbox
  - Apache Spark, from Python, R and Scala
  - Pandas, scikit-learn, ggplot2, dplyr, etc

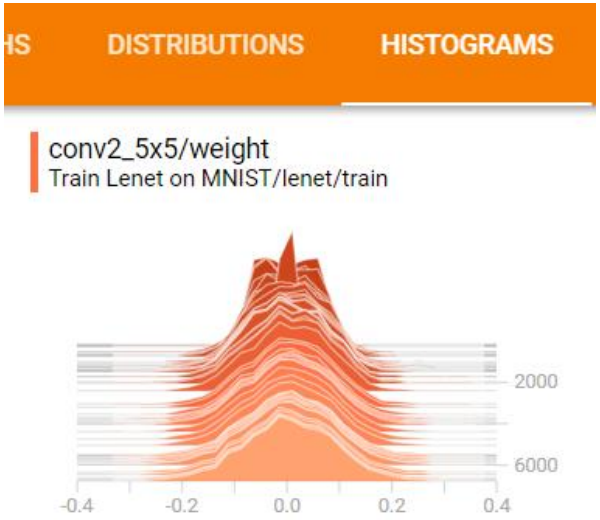


VISUALIZATION FOR LEARNING

BigDL integration with TensorBoard

- TensorBoard is a suite of web applications from Google for visualizing and understanding deep learning applications





CURRENT RELEASE BIGDL 0.5.0

Open Source Community support:
2270+ STARS | 500+ FORKS | 50 CONTRIBUTORS

- Support more Tensorflow operations, e.g., loading Tensorflow dynamic models (e.g. LSTM, RNN) in BigDL
- Support combining data pre-processing and neural network layers in the same model (to make model deployment easy)
- Keras-like APIs (Scala and Python) for users to run their Keras code on BigDL
- Speedup various modules in BigDL (BCECriterion, RMSprop, LeakyRelu, etc.)
- Add DataFrame-based image reader and transformer

Please refer to the release note at https://github.com/intel-analytics/BigDL/releases/tag/v0.5.0 for more details

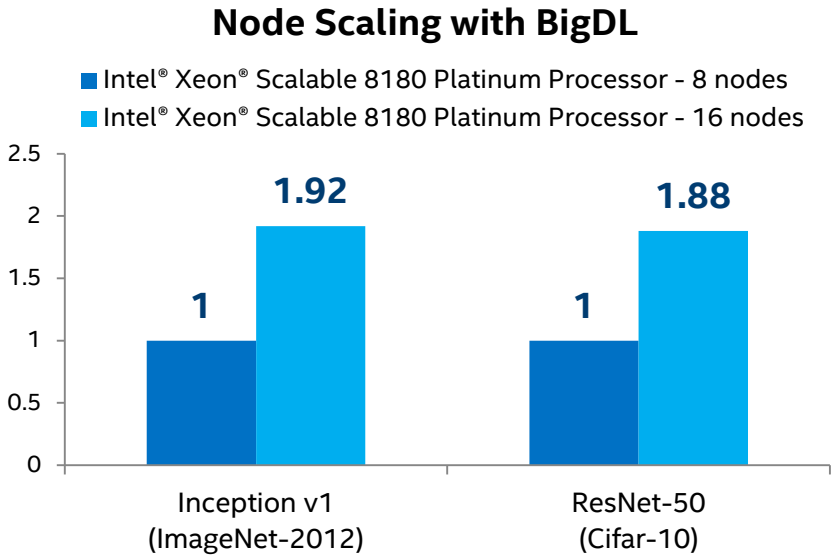
BIGDL ANALYTICS ZOO

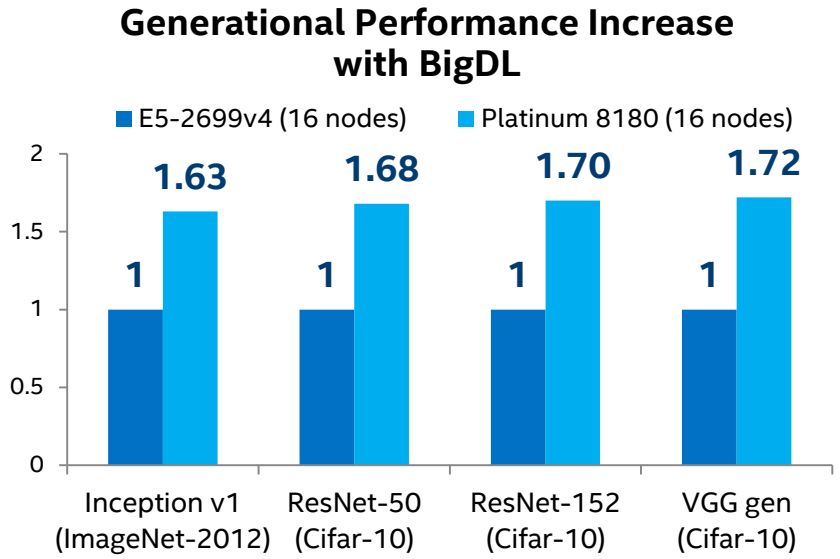
Analytics + AI Pipelines for Spark and BigDL

“Out-of-the-box” ready for use

- Reference use cases
  - Fraud detection, time series prediction, sentiment analysis, chatbot, etc.
- Predefined models
  - Object detection, image classification, text classification, recommendations, etc.
- Feature transformations
  - Vision, text, 3D imaging, etc.
- High level APIs
  - DataFrames, ML Pipelines, Keras/Keras2, etc.

DEEP LEARNING WITH BIGDL/SPARK





Excellent scaling & generational performance with your existing hardware

BUILDING AND DEPLOYING WITH BIGDL

TECHNOLOGY

bluedata

Lightbend

cloudera

databricks

KINGSOFT

CRAY THE SUPERCOMPUTER COMPANY

DELL

Qubole

CLOUD SERVICE PROVIDERS

Alibaba Cloud aliyun.com

Azure

aws

Google Cloud Platform

IBM Cloud

END USERS

UnionPay 银联

JD.COM 京东

GIGASPACES innovate with confidence

cdhi

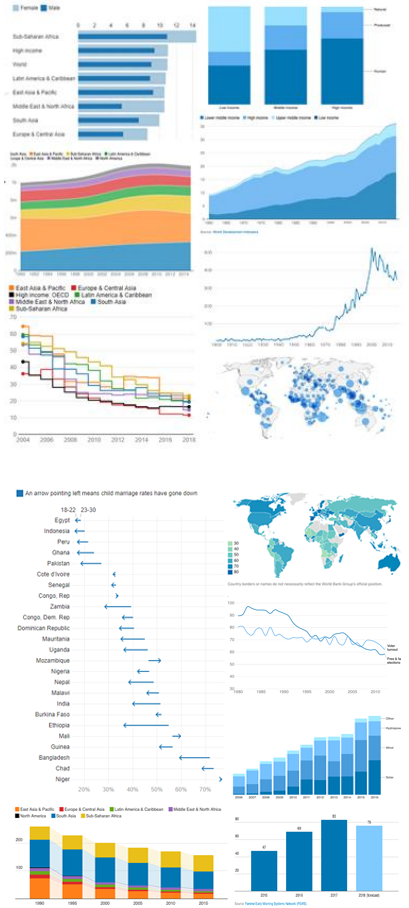
USE CASE

WORLD BANK

The World Bank is a vital source of financial and technical assistance to developing countries around the world. It is not a bank in the ordinary sense but a unique partnership to reduce poverty and support development. The World Bank Group comprises five institutions managed by their member countries.

Established in 1944, the **World Bank Group** is headquartered in Washington, D.C. It has more than 10,000 employees in more than 120 offices worldwide.

The Development Data Group provides high-quality national and international statistics to clients within and outside the World Bank and to improve the capacity of member countries to produce and use statistical information.



PROBLEM STATEMENT

The International Comparison Program team in the World Bank Development Data Group collected crowdsourced images for a pilot data collection study through a privately-operated network of paid on-the-ground contributors that had access to a smartphone and a data collection application designed for the pilot.

Nearly 3 million labeled images were collected as ground truth/metadata attached to each price observation of 162 tightly specified items for a variety of household goods and services. The use of common item specifications aimed at ensuring the quality, as well as intra- and inter-country comparability, of the collected data.

Goal is to reduce labor intensive tasks of manually moderating (reviewing, searching and sorting) the crowd-sourced images before their release as a public image dataset that could be used to train various deep learning models.

OUR CHALLENGES

- Crowdsourced images are of different quality (resolution, close-up, etc.)
- Images sourced from 15 different countries – different language groups represented in the text example of images
- Some text is typed, some text is handwritten

CLASSIFYING REAL FOOD IMAGES IS NOT A CAT VS DOG PROBLEM

PROJECT LAYOUT

Phase 1:

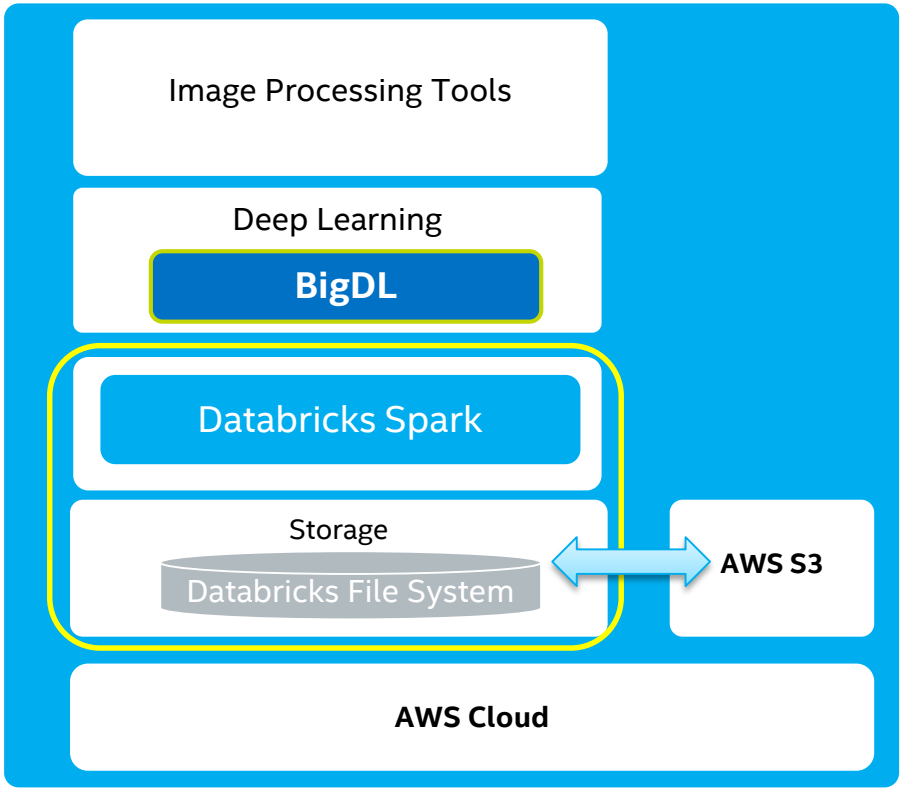
- Define image quality (eliminate poor quality images)
- Classify images (by food type) to validate existing labels

Phase 2:

- Identify images with text in the existing dataset; circle text
- Text recognition (words/sentences in the image text)
- Determine whether text contains PII (personal identifiable information)
- Blur areas with PII text

MODEL DEVELOPMENT & RESULTS

SOLUTION ARCHITECTURE



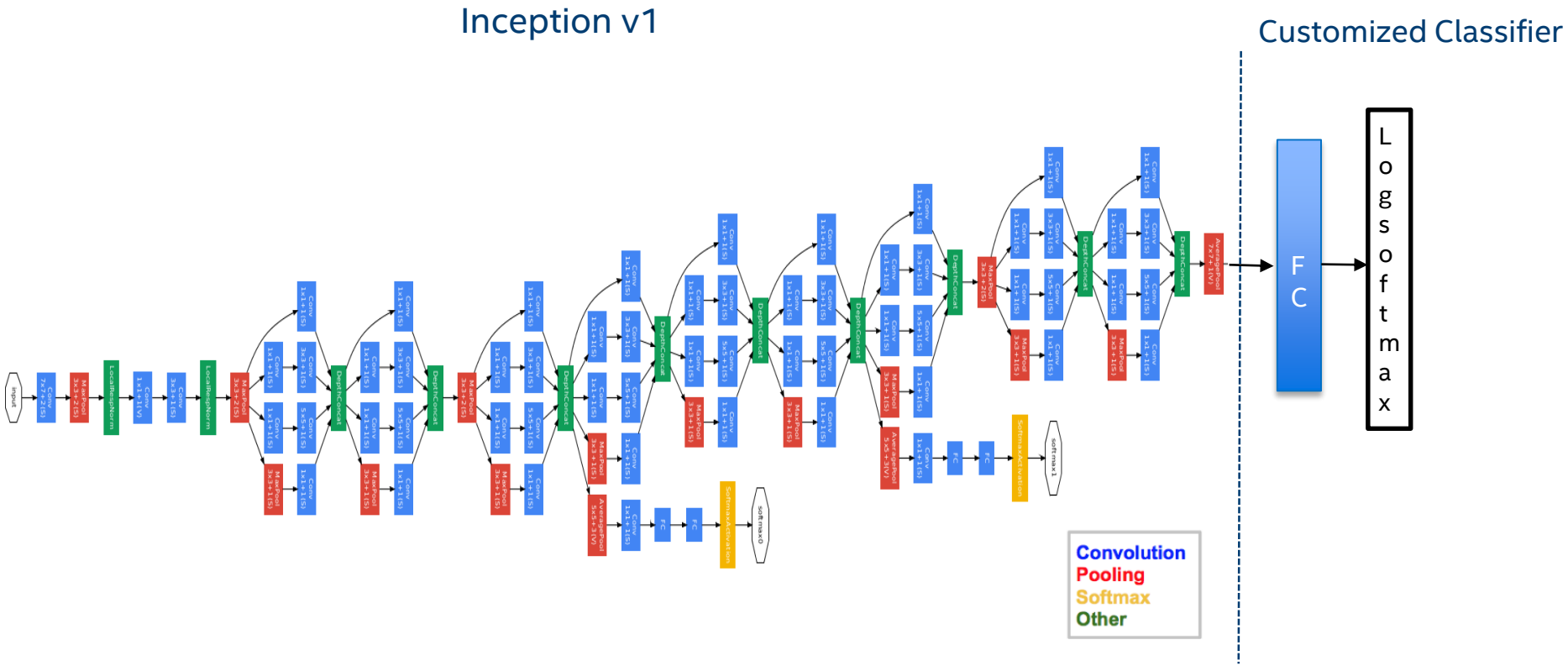
- BigDL 0.5
- Databricks Spark
- AWS S3
- AWS R4 instance

MODEL DEVELOPMENT - PHASE 1

Transfer learning from pre-trained Inception model to do classification

- Load pre-trained Caffe Inception-v1 model to BigDL
- Add FC layer with SoftMax classifier (9 classes)
- Train on Food dataset with pre-trained weights using BigDL on Spark
- Reduce training time and improve model accuracy when compared with training Inception model from scratch
- Scale training on multi-node cluster in AWS Databricks to train large whole dataset

MODEL DEVELOPMENT - PHASE 1



RESULTS - PHASE 1

Transfer learning vs Training from scratch

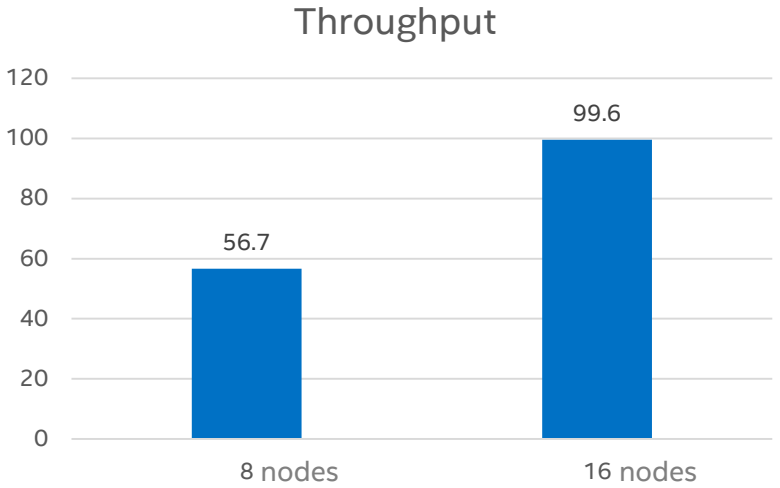
Dataset: 1927 images, 9 categories

	Epoch	Training Time(s)	Accuracy
Transfer Learning	40	210	65.4
Training from scratch	40	1266	23.9

* Accuracy numbers are in that range due to a small part of dataset being used

SCALING RESULTS – PHASE 1

Nodes	BatchSize	Epoch	Throughput	Training Time
8	256	20	56.7	745.6
16	256	20	99.6	424.7
8	128	20	55.3	764.9
16	128	20	80.7	524.4



Test was run on AWS R4 instance that include the following features:

- dual socket Intel Xeon E5 Broadwell processors (2.3 GHz)
- DDR4 memory
- Hardware Virtualization (HVM) only

Model	vCPUs	Memory (GiB)	Networking Performance
r4.xlarge	4	30.5	Up to 10 Gigabit

NEXT STEPS – PHASE 2

- Identify images with text in the existing dataset; circle text
- Text recognition (words/sentences in the image text)
- Determine whether text contains PII (personal identifiable information)
- Blur areas with PII text

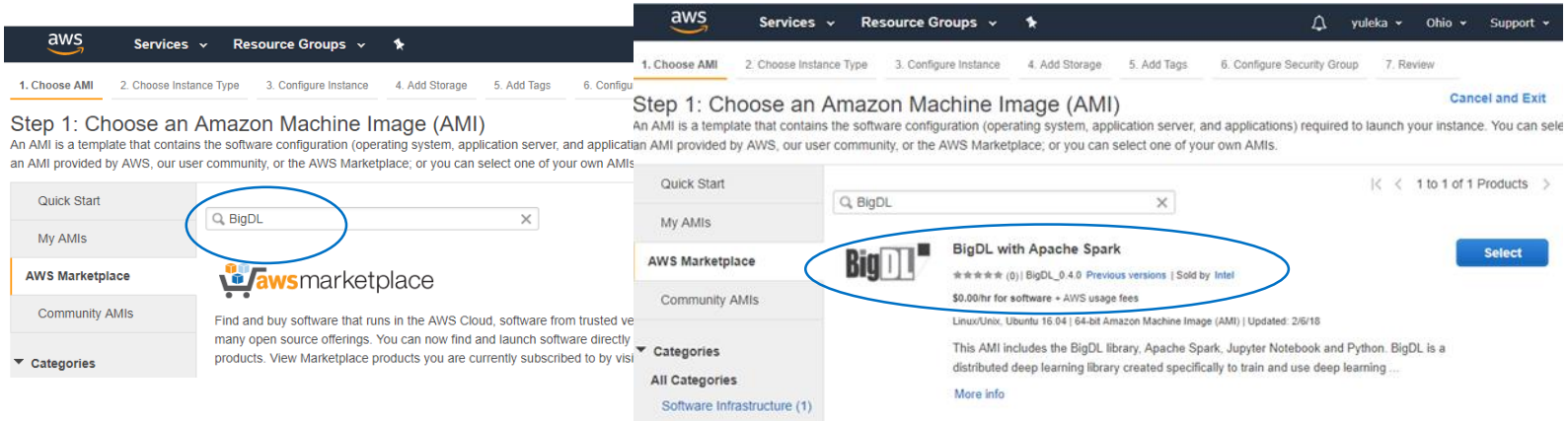
SUMMARY

KEY TAKEAWAYS

- AI on Apache Spark is a reality with usecases like World Bank
- BigDL makes distributed deep learning and AI more accessible both for big data users and data scientists
- BigDL can leverage existing on prem Spark/Hadoop infrastructure and also runs deep learning applications in the cloud (AWS, Azure, GCP, …)
- Join in and contribute to the project github.com/intel-analytics/BigDL

CALL TO ACTION

- Try BigDL on AWS - lookup BigDL AMI on AWS Marketplace



- Try image classification with BigDL – this usecase code is shared on https://github.com/intel-analytics/WorldBankPoC

BIGDL RESOURCES

HTTPS://GITHUB.COM/INTEL-ANALYTICS/BIGDL

SOFTWARE.INTEL.COM/BIGDL

Join Our Mail List
bigdl-user-group+subscribe@googlegroups.com

Report Bugs And Create Feature Request
https://github.com/intel-analytics/BigDL/issues


intel®
AI

NOTICES AND DISCLAIMERS

The findings, interpretations, and conclusions expressed in this document do not necessarily reflect the views of the World Bank Group.

Software and workloads used in performance tests may have been optimized for performance only on Intel microprocessors. Performance tests, such as SYSmark and MobileMark, are measured using specific computer systems, components, software, operations and functions. Any change to any of those factors may cause the results to vary. You should consult other information and performance tests to assist you in fully evaluating your contemplated purchases, including the performance of that product when combined with other products. For more complete information visit: **http://www.intel.com/performance**.

Tests document performance of components on a particular test, in specific systems. Differences in hardware, software, or configuration will affect actual performance. Consult other sources of information to evaluate performance as you consider your purchase. For more complete information about performance and benchmark results, visit **www.intel.com/benchmarks**.

Intel technologies' features and benefits depend on system configuration and may require enabled hardware, software or service activation. Performance varies depending on system configuration. No computer system can be absolutely secure. Check with your system manufacturer or retailer or learn more at intel.com.

The products described may contain design defects or errors known as errata which may cause the product to deviate from published specifications. Current characterized errata are available on request.

Intel, the Intel logo, Xeon, Xeon Phi and Nervana are trademarks of Intel Corporation in the U.S. and/or other countries.

*Other names and brands may be claimed as the property of others