Software

ADVANCED DATA ANALYTICS AND DEEP LEARNING ON APACHE SPARK WITH BIGDL

Yuhao Yang, Zhichao Li

Big Data Technology, Intel

Outline

BigDL

- Apache Spark* + High Performance + Deep Learning

Speech recognition:

- Deep Speech 2 on BigDL: ML Pipeline + BigDL

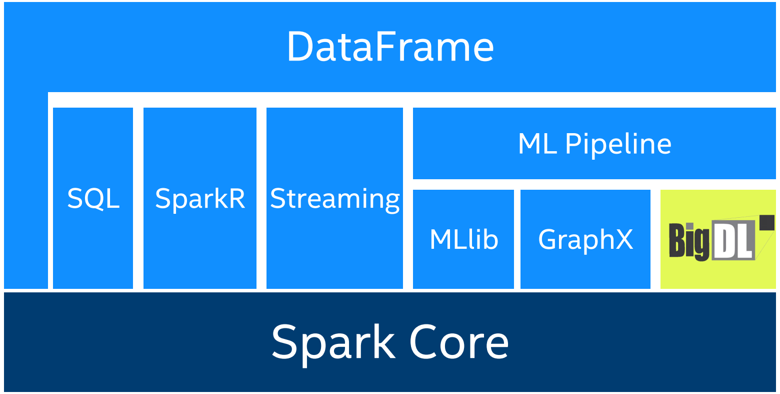
Object detection:

- SSD and use cases.

BigDL

Bringing Deep Learning To Big Data Platform

- Distributed open source deep learning framework for Apache Spark*, 2000+ star on Github
- 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



https://github.com/intel-analytics/BigDL

https://bigdl-project.github.io/

BigDL Answering The Needs

Make deep learning more accessible to big data and data science communities

- Continue the use of familiar SW tools and HW infrastructure to build deep learning applications

- Add deep learning functionalities to the Big Data (Spark) programs and/or workflow

- Leverage existing Hadoop/Spark clusters to run deep learning applications
  - **Shared with other workloads (*e.g., ETL, data warehouse, feature engineering, statistic machine learning, graph analytics, etc.*) in a dynamic and elastic fashion**

Basic Component

Tensor:

- ND-array data structure
- Generic data type
- Rich and fast math operations (powered by Intel MKL)

Layers

- 150+ layers (Conv, 3D Conv, Pooling, RNN, FC ...)

Criterion

- 20+ criterions (DiceCoefficient, ClassNLL, CrossEntropy ...)

Optimization

- SGD, Adagrad, LBFGS, Adam, Adadelta, RMSprop, Adamx

DEEP SPEECH 2 WITH BIGDL

Deep Speech 2 for Speech Recognition

- DS2 system **outperforms humans in 3 out of the 4 test sets and is competitive on the fourth**.

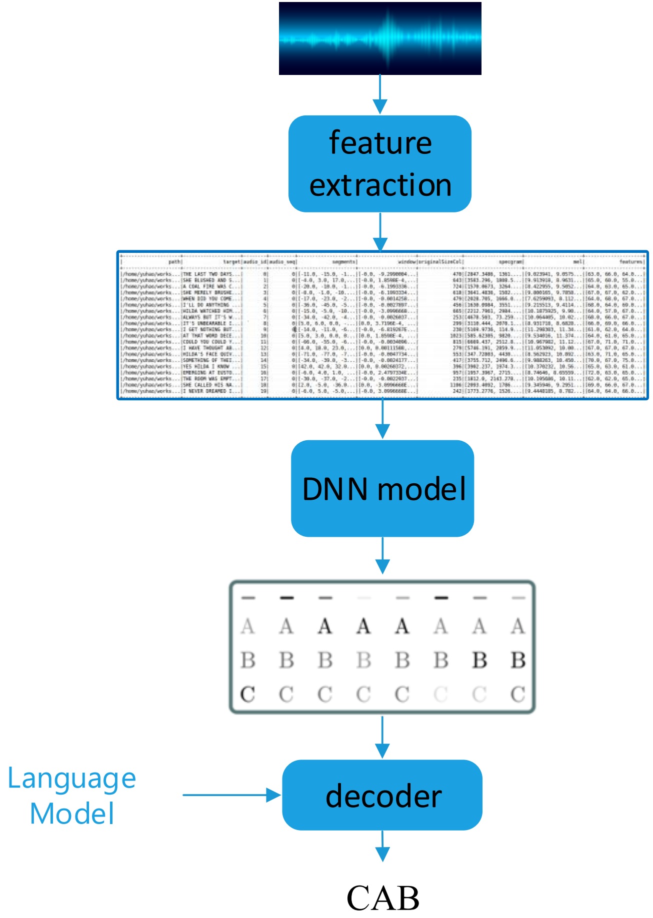
Read Speech			
Test set	DS1	DS2	Human
WSJ eval'92	4.94	3.60	5.03
WSJ eval'93	6.94	4.98	8.08
LibriSpeech test-clean	7.89	5.33	5.83
LibriSpeech test-other	21.74	13.25	12.69

Table 13: Comparison of WER for two speech systems and human level performance on read speech.

Deep Speech 2 on BigDL

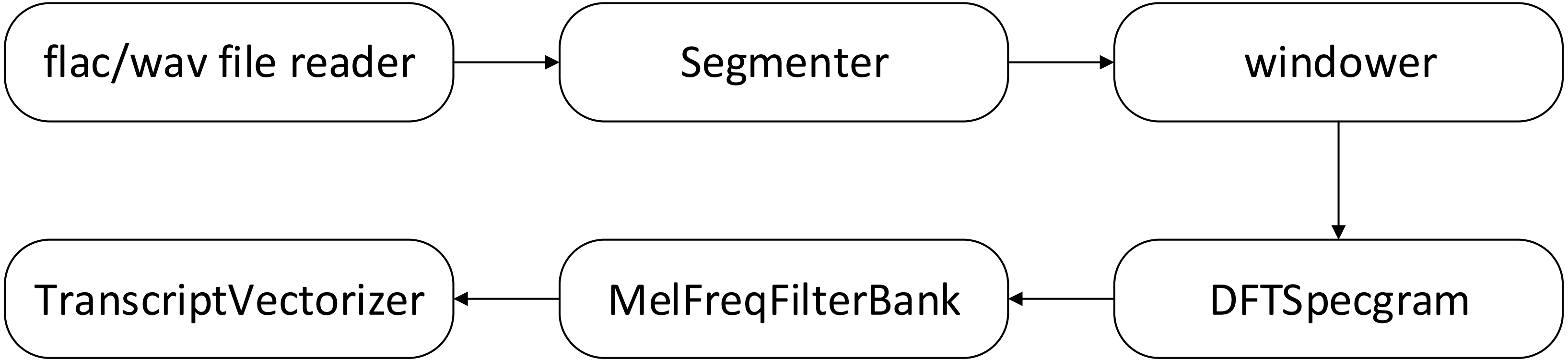
Deep Speech 2: End-to-End Speech Recognition in English and Mandarin

Baidu Research – Silicon Valley AI Lab*
Dario Amodei, Rishita Anubhai, Eric Battenberg, Carl Case, Jared Casper, Bryan Catanzaro, Jingdong Chen, Mike Chrzanowski, Adam Coates, Greg Diamos, Erich Elsen, Jesse Engel, Linxi Fan, Christopher Fougner, Tony Han, Awni Hannun, Billy Jun, Patrick LeGresley, Libby Lin, Sharan Narang, Andrew Ng, Sherjil Ozair, Ryan Prenger, Jonathan Raiman, Sanjeev Satheesh, David Seetapun, Shubho Sengupta, Yi Wang, Zhiqian Wang, Chong Wang, Bo Xiao, Dani Yogatama, Jun Zhan, Zhenyao Zhu



Deep Speech 2 on BigDL: Feature transformers

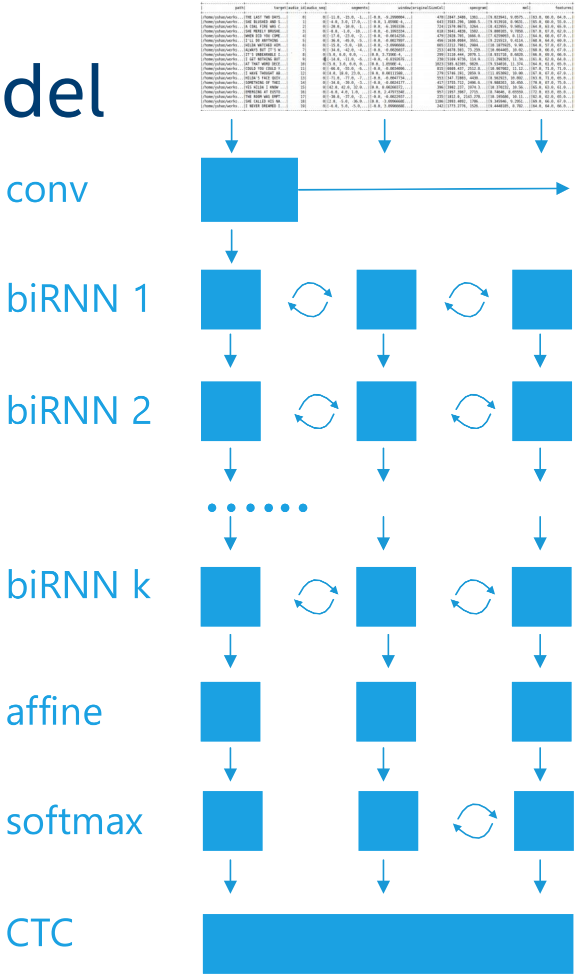
Apache Spark* ML Pipeline



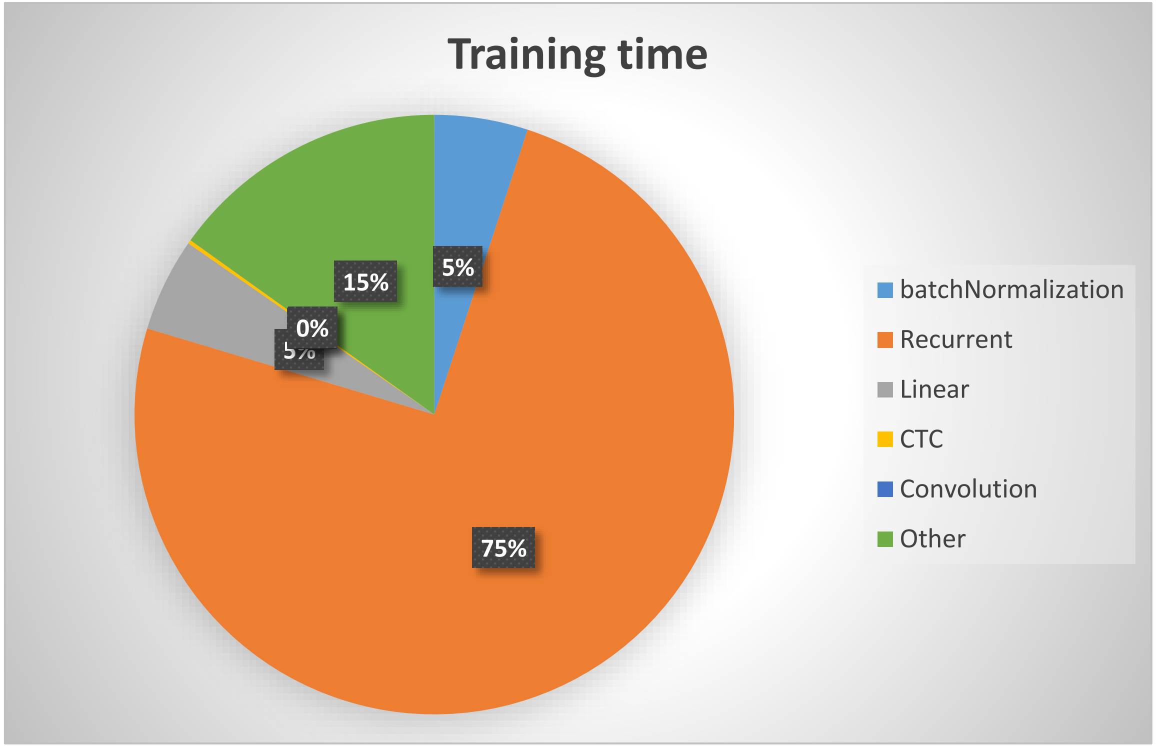
Deep Speech 2 on BigDL: Model

```
val model = Sequential[T]()
  .add(conv)
  .add(ReLU[T]())
  .add(Squeeze(4))
  .add(brnn)
  .add(linear1)
  .add(HardTanhDS[T](0, 20, true))
  .add(linear2)
```

9 layers biRNN: >50 Million parameters



Deep Speech 2 on BigDL: Model training



With libriSpeech, 5 RNN layer, 30 seconds uttLength, 30 epoches.

Deep Speech 2 with LibriSpeech

- Deep Speech 2 (12 layers, 9 RNN), uttLength 30 seconds, with arg-max decoder
  - Word Error Rate with hold-out validation dataset

	cer	wer(without LM)
Hannun, et al. (2014)	10.7	35.8
Graves-Jaitly (ICML 2014)	9.2	30.1
Hwang-Sung (ICML 2016)	10.6	38.4
BigDL	8.7	32.4

Deep Speech 2 on BigDL: Summary

Feature transformers:

- Flac/wav Reader, Windower, TimeSegmenter, TranscriptVectorizer, DFTSpecgram, MelFrequencyFilterBank

Model training and inference

- Big DL container, optimizer, Convolution, BatchNormalization, Bi-RNN

CTC (Connectionist Temporal Classification) loss

- Scala or JNI (warp-ctc)

Decoder

- ArgmaxDecoder, VocabDecoder

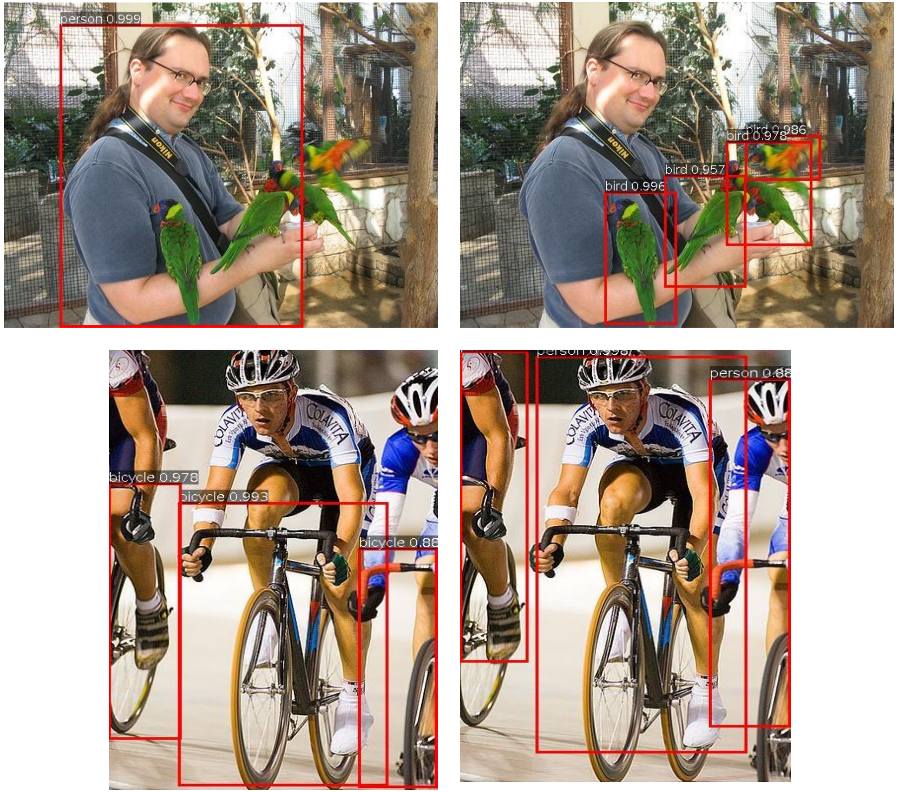
Evaluation

- wer, cer

OBJECT DETECTION WITH BIGDL

SSD: Single Shot Multibox Detector

- State-of-the-art object detection pipeline
- Single shot

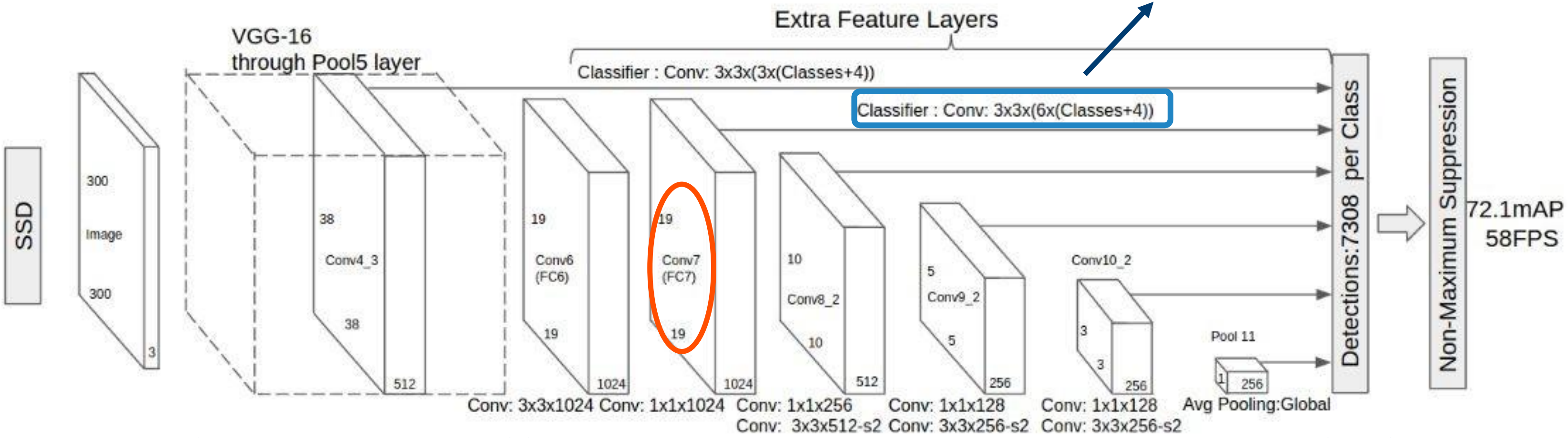


Images from PASCAL(http://host.robots.ox.ac.uk/pascal/VOC/)

Liu, Wei, et al. "SSD: Single shot multibox detector." European Conference on Computer Vision. Springer International Publishing, 2016.

The Single Shot Detector (SSD)

Convolutional predictors for detection



base network

fc6 and fc7 converted to convolutional

collection of bounding boxes and scores for each category

Multi-scale feature maps for detection: observe how conv feature maps decrease in size and allow predictions at multiple scales

SSD Pipeline

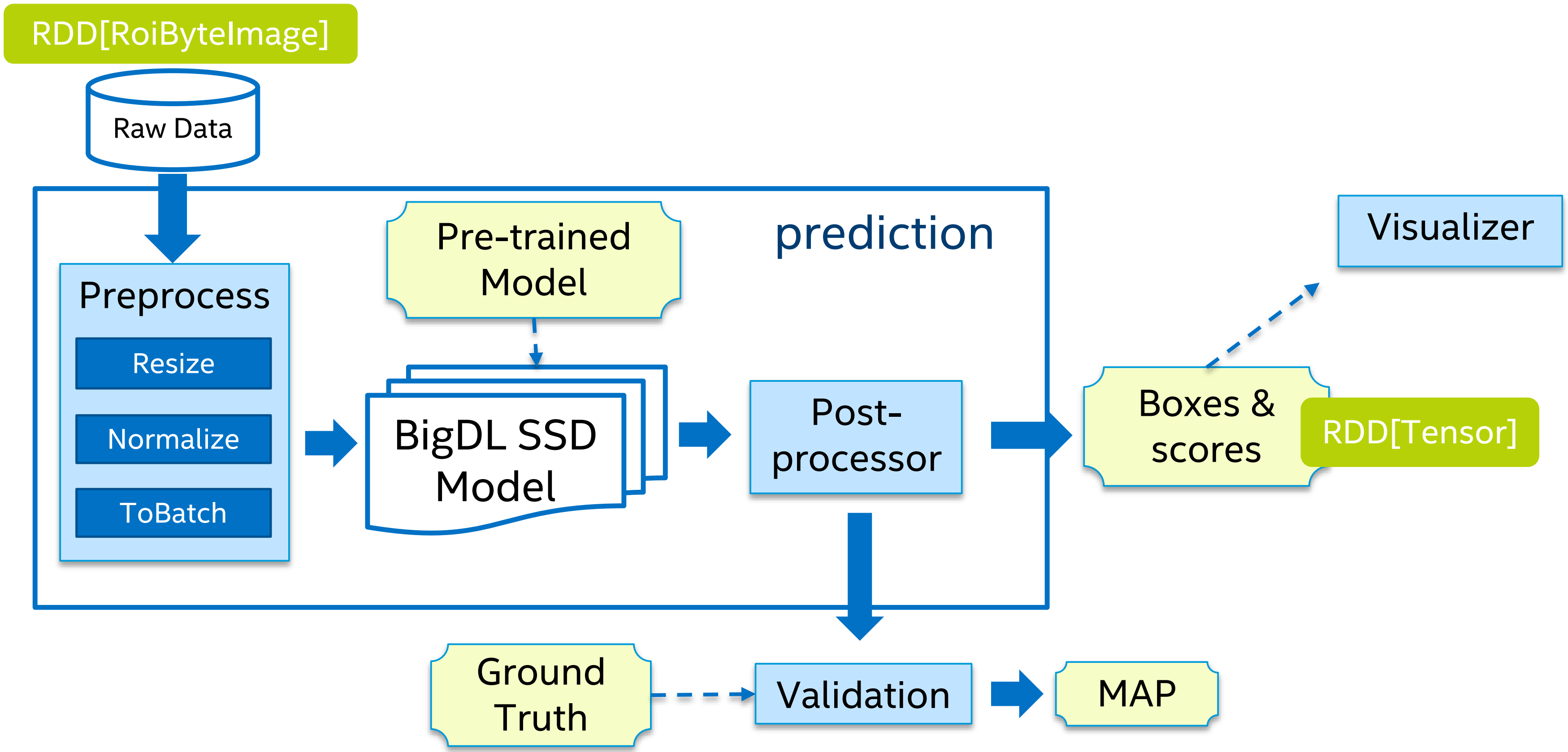


Image Pre-processing for Spark ML Pipeline

Image Transformer based on steps, use OpenCV.Mat as interchange format.

```
val steps = BytesToMat() ->
    Resize(250, 250) ->
    Flip(Flip.HORIZONTAL_FLIP) ->
    Cropper(224, 224) ->
    BGRImageNormalizer(0.485f, 0.456f, 0.406f, 0.229f, 0.224f, 0.225f) ->
    BGRToRGB() ->
    MatToFloats()

val imgTransfomer = new ImageTransformer(steps)
  .setInputCol("imagseData").setOutputCol("feature")
```

SSD + VGG test over Pascal VOC 2007

- SSD + VGG 300x300 with pretrained model over voc07+12

	Caffe Model	BigDL
Mean Average Precision	77.2	77.3

- SSD + VGG 512x512 with pretrained model over voc07+12

	Caffe Model	BigDL
Mean Average Precision	79.6	79.6

JD.com: Find visually similar products

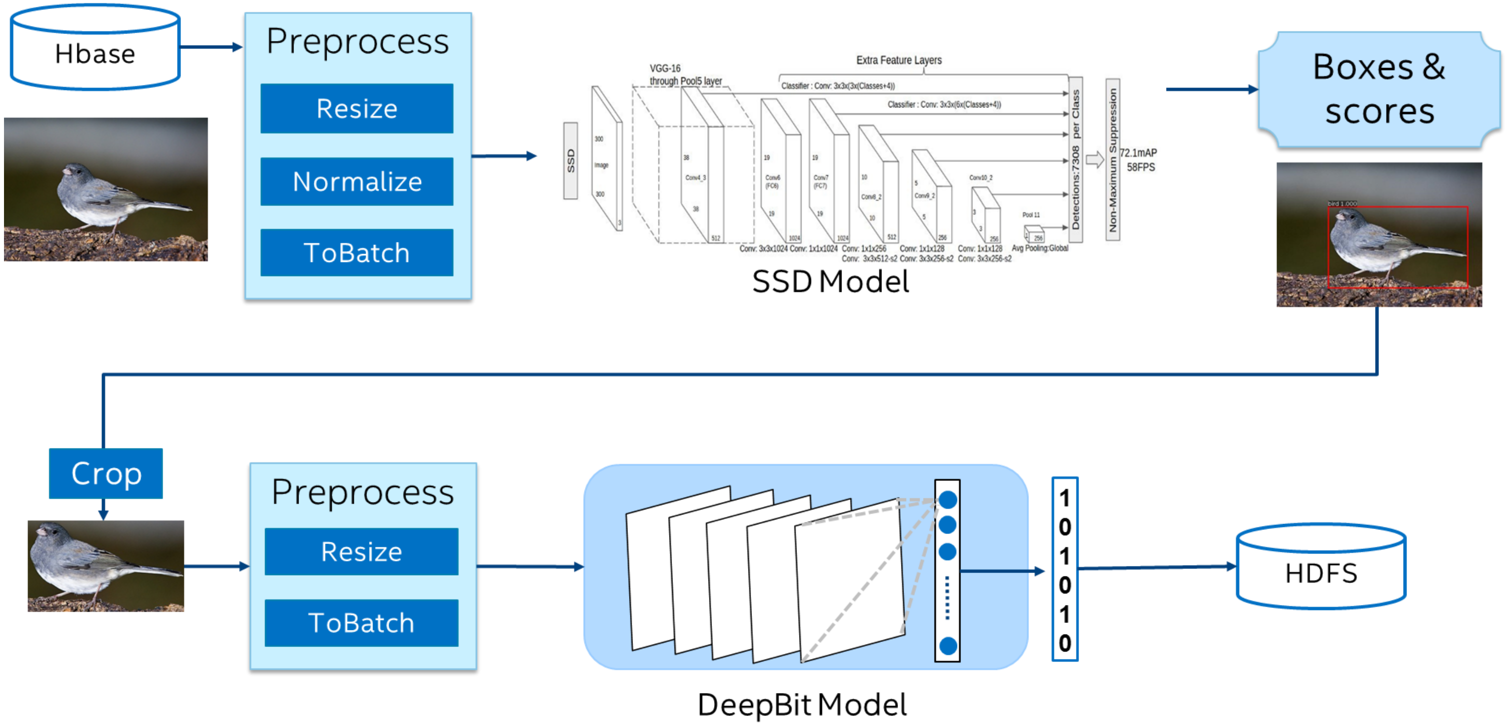
(a) Similar catalog items with and without human model

(b) Concept based similarity across spooky printed t-shirts

(c) Detail based similarity via spacing and thickness of stripes

(d) Wild Image similarity across radically different poses

JD.com: Image Detection & Extraction Pipeline (using SSD + DeepBit Models)



Similar house search

Latency: 1000 image comparison 0.03 second on single thread

Challenges of Large-Scale Processing in GPU

Reading images out takes a very long time

Image pre-processing on HBase is very complex

No existing software frameworks can be leveraged
- E.g., resource management, distributed data processing, fault tolerance, etc.

Very challenging to scale out to massive amount of pictures
- Due to SW and HW infrastructure constraints

Upgrading to BigDL Solutions

Reuse existing Hadoop/Spark clusters for deep learning with no changes

Efficiently scale out on Spark with superior performance

- Reading HBase data no longer a bottleneck

Very easy to build the end-to-end pipeline in BigDL

- Image transformation and augmentation based on OpenCV on Spark

  val preProcessor = BytesToMat() -> Resize(300, 300) -> …
  val transformed = preProcessor(dataRdd)

- Directly Load pre-trained models (BigDL/Caffe/Torch/TensorFLow) into BigDL

  val model = Module.loadCaffeModel(caffeDefPath, caffeModelPath)

Model Quantization for Efficient Inference in BigDL

Local quantization scheme converting floats to intergers

- Faster compute and smaller models
- Take advantage of SSE and AVX instructions on Xeon servers
- Supports pre-trained models (BigDL/Caffe/Torch/TensorFLow)

```
val model = Module.loadCaffeModel(caffeDefPath, caffeModelPath)
val quantizedModel = model.quantize()
```

Quantized SSD model

- ~4x model size reduction
- >2x inference speedup
- ~0.001 mAP (mean average precision) loss

Try BigDL Out

Running BigDL, Deep Learning for Apache Spark, on AWS* (Amazon* Web Service)
https://aws.amazon.com/blogs/ai/running-bigdl-deep-learning-for-apache-spark-on-aws/

Use BigDL on Microsoft* Azure* HDInsight*
https://azure.microsoft.com/en-us/blog/use-bigdl-on-hdinsight-spark-for-distributed-deep-learning/

BigDL on Alibaba* Cloud E-MapReduce*
https://yq.aliyun.com/articles/73347

BigDL on CDH* and Cloudera* Data Science Workbench*
http://blog.cloudera.com/blog/2017/04/bigdl-on-cdh-and-cloudera-data-science-workbench/

Intel's BigDL on Databricks*
https://databricks.com/blog/2017/02/09/intels-bigdl-databricks.html

BigDL Distribution in Cray Urika-XC Analytics Suite
http://www.cray.com/products/analytics/urika-xc

PARTNER WITH US

- Use BigDL & Share your Experience
- Use Intel Optimized Libraries & Frameworks
- Leverage Intel Developer Zone Resources

https://github.com/intel-analytics/BigDL http://software.intel.com/ai intel®

Software

Legal Disclaimer

Risk Factors

The above statements and any others in this document that refer to plans and expectations for the first quarter, the year and the future are forward-looking statements that involve a number of risks and uncertainties. Words such as “anticipates,” “expects,” “intends,” “plans,” “believes,” “seeks,” “estimates,” “may,” “will,” “should” and their variations identify forward-looking statements. Statements that refer to or are based on projections, uncertain events or assumptions also identify forward-looking statements. Many factors could affect Intel’s actual results, and variances from Intel’s current expectations regarding such factors could cause actual results to differ materially from those expressed in these forward-looking statements. Intel presently considers the following to be the important factors that could cause actual results to differ materially from the company’s expectations. Demand could be different from Intel's expectations due to factors including changes in business and economic conditions; customer acceptance of Intel’s and competitors’ products; supply constraints and other disruptions affecting customers; changes in customer order patterns including order cancellations; and changes in the level of inventory at customers. Uncertainty in global economic and financial conditions poses a risk that consumers and businesses may defer purchases in response to negative financial events, which could negatively affect product demand and other related matters. Intel operates in intensely competitive industries that are characterized by a high percentage of costs that are fixed or difficult to reduce in the short term and product demand that is highly variable and difficult to forecast. Revenue and the gross margin percentage are affected by the timing of Intel product introductions and the demand for and market acceptance of Intel's products; actions taken by Intel's competitors, including product offerings and introductions, marketing programs and pricing pressures and Intel’s response to such actions; and Intel’s ability to respond quickly to technological developments and to incorporate new features into its products. The gross margin percentage could vary significantly from expectations based on capacity utilization; variations in inventory valuation, including variations related to the timing of qualifying products for sale; changes in revenue levels; segment product mix; the timing and execution of the manufacturing ramp and associated costs; start-up costs; excess or obsolete inventory; changes in unit costs; defects or disruptions in the supply of materials or resources; product manufacturing quality/yields; and impairments of long-lived assets, including manufacturing, assembly/test and intangible assets. Intel's results could be affected by adverse economic, social, political and physical/infrastructure conditions in countries where Intel, its customers or its suppliers operate, including military conflict and other security risks, natural disasters, infrastructure disruptions, health concerns and fluctuations in currency exchange rates. Expenses, particularly certain marketing and compensation expenses, as well as restructuring and asset impairment charges, vary depending on the level of demand for Intel's products and the level of revenue and profits. Intel’s results could be affected by the timing of closing of acquisitions and divestitures. Intel's results could be affected by adverse effects associated with product defects and errata (deviations from published specifications), and by litigation or regulatory matters involving intellectual property, stockholder, consumer, antitrust, disclosure and other issues, such as the litigation and regulatory matters described in Intel's SEC reports. An unfavorable ruling could include monetary damages or an injunction prohibiting Intel from manufacturing or selling one or more products, precluding particular business practices, impacting Intel’s ability to design its products, or requiring other remedies such as compulsory licensing of intellectual property. A detailed discussion of these and other factors that could affect Intel’s results is included in Intel’s SEC filings, including the company’s most recent reports on Form 10-Q, Form 10-K and earnings release.