AUTOMATIC 3D MRI KNEE DAMAGE CLASSIFICATION WITH 3D CNN USING BIGDL ON SPARK

Valentina Pedoia; Center for Digital Health Innovation at UCSF
Yulia Tell and Jennie Wang; Big Data Technologies, Intel Corporation

AGENDA

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

ANALYTICS OPPORTUNITIES IN EVERY INDUSTRY

CONSUMER

HEALTH

FINANCE

RETAIL

ENERGY

TRANSPORT

INDUSTRIAL

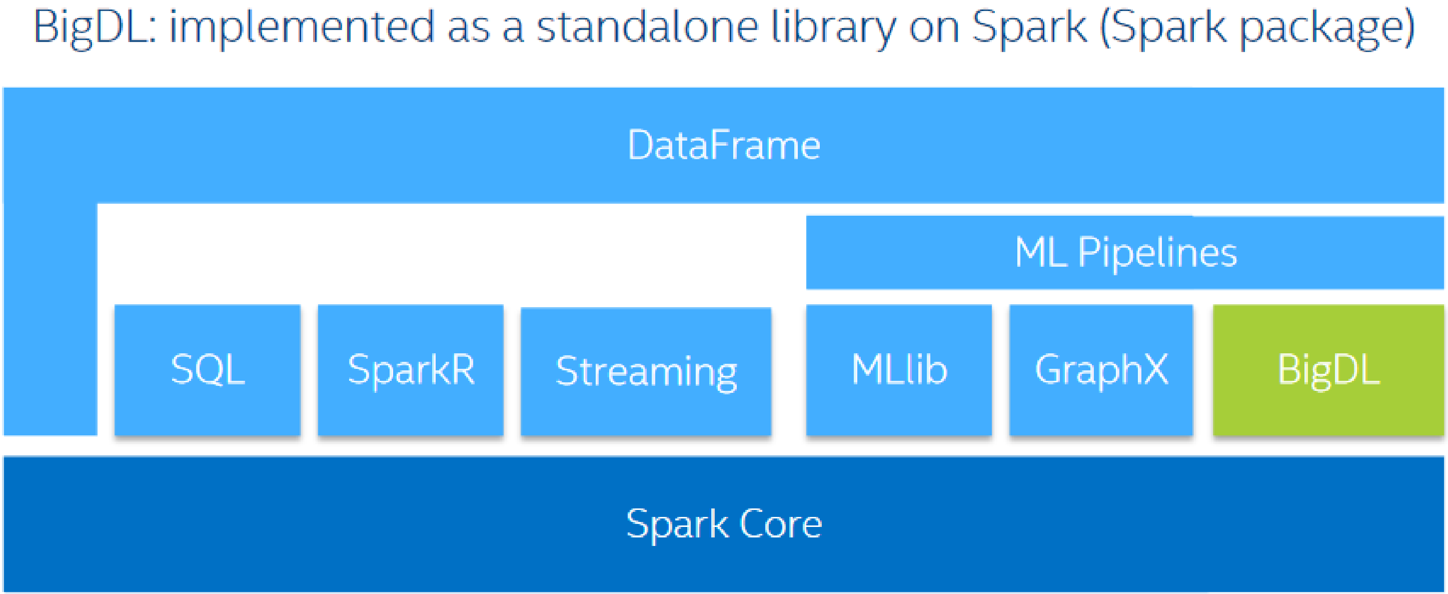
ACCELERATING BUSINESS GAINS AND COMPETITIVE ADVANTAGE

intel

BIGDL OVERVIEW

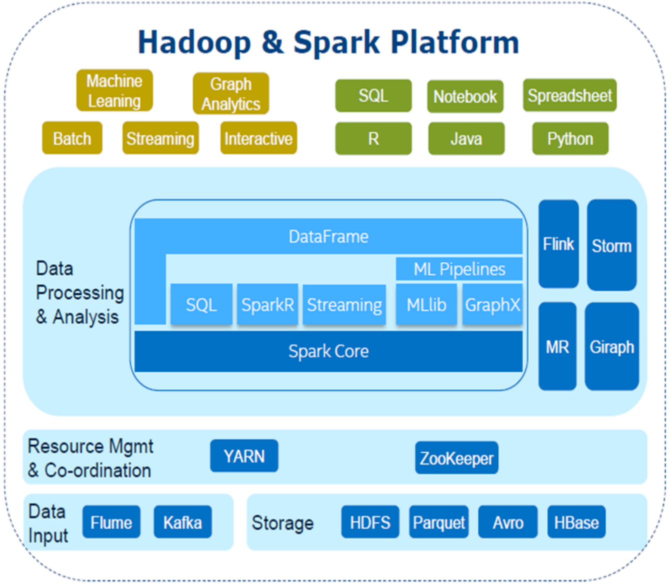
WHAT IS BIGDL?

BigDL is a distributed deep learning library for Apache Spark*



BIGDL IS DESIGNED FOR BIG DATA

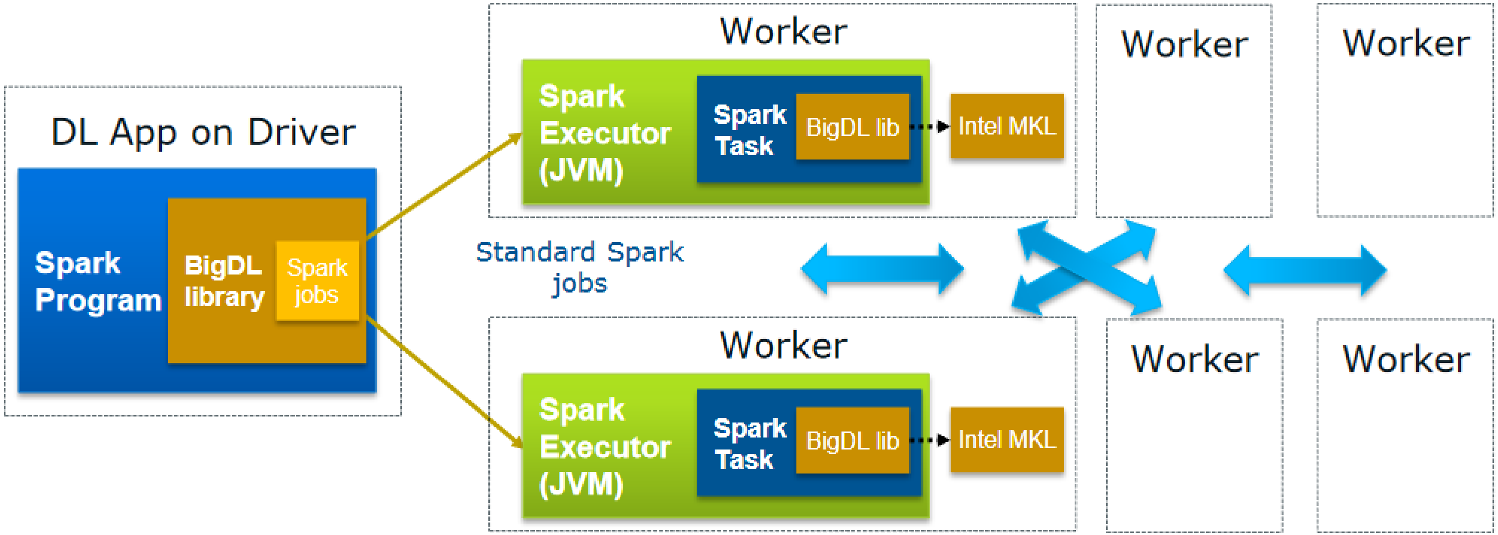
- Distributed deep learning framework for Apache Spark*
- Make deep learning more accessible to big data users and data scientists
  - Write deep learning applications as ***standard Spark programs***
  - Run on existing Spark/Hadoop clusters (***no changes needed***)
- Feature parity with popular deep 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

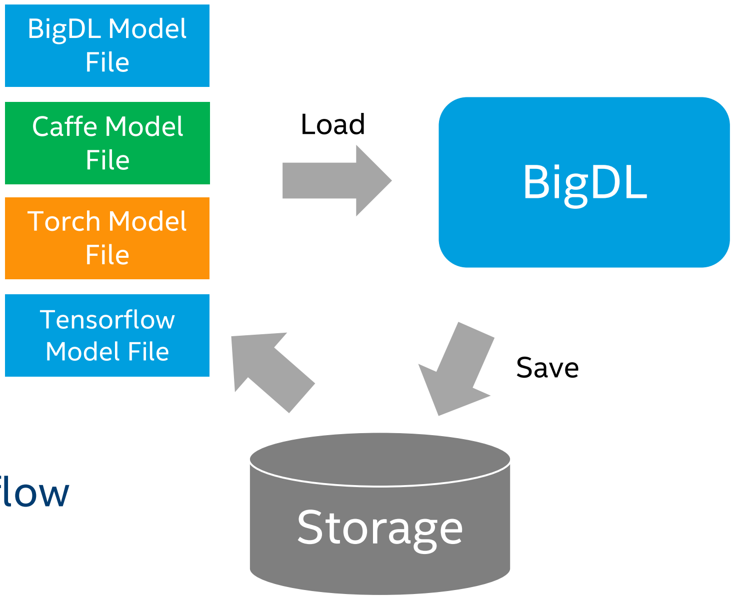
Distributed Deep learning applications (training, fine-tuning & prediction) on Apache Spark*

- 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

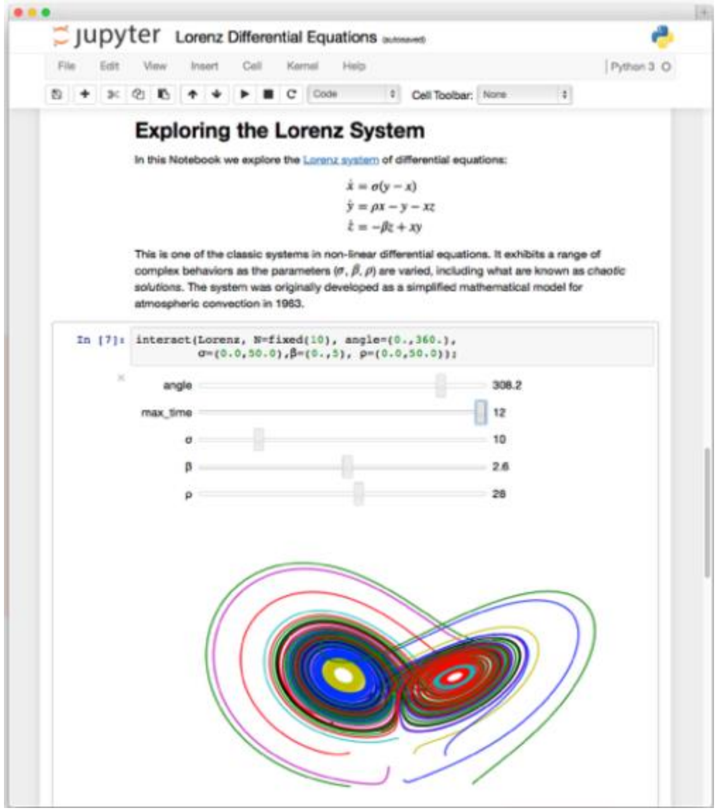
- Notebooks can be shared with others
- Easy to reproduce and track

✓ Rich Content

- Texts, images, videos, LaTeX and JavaScript
- Code can also produce rich contents

✓ Rich toolbox

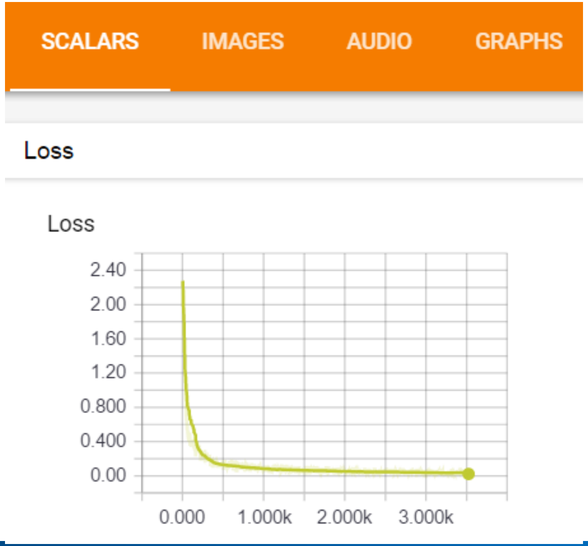
- Apache Spark, from Python, R and Scala
- Pandas, scikit-learn, ggplot2, dplyr, etc

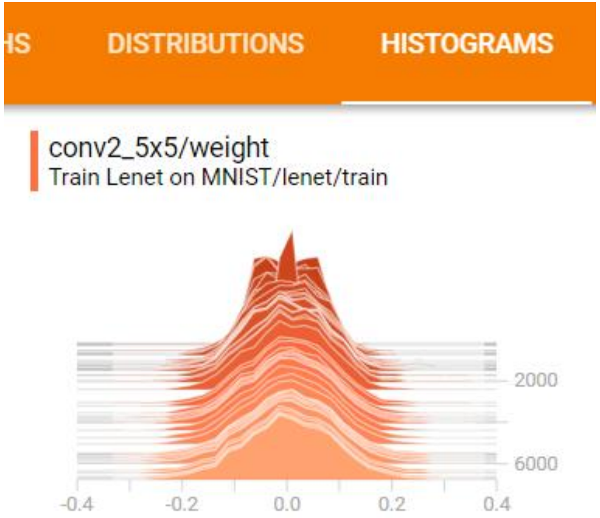


VISUALIZATION FOR LEARNING

BigDL integration with TensorBoard

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





ENABLING 3D IMAGING SUPPORT IN BIGDL

3D Convolution(VolumetricConvolution): Applies a 3D convolution over 3D input data. The input could be 3D input image, a sequence of images, or a video etc. The input tensor is expected to be a 4D tensor (nInputPlane x time(slice) x height x width).

3D Max Pooling(VolumetricMaxPooling): Applies 3D max-pooling operation in kTxkWxkH regions by step size dTxdWxdH.

The number of output features is equal to the number of input planes / dT.

ENABLING 3D IMAGING SUPPORT IN BIGDL (CONT)

3D Deconvolution(VolumetricFullConvolution): Apply a 3D full convolution over an 3D input image, a sequence of images, or a video etc.

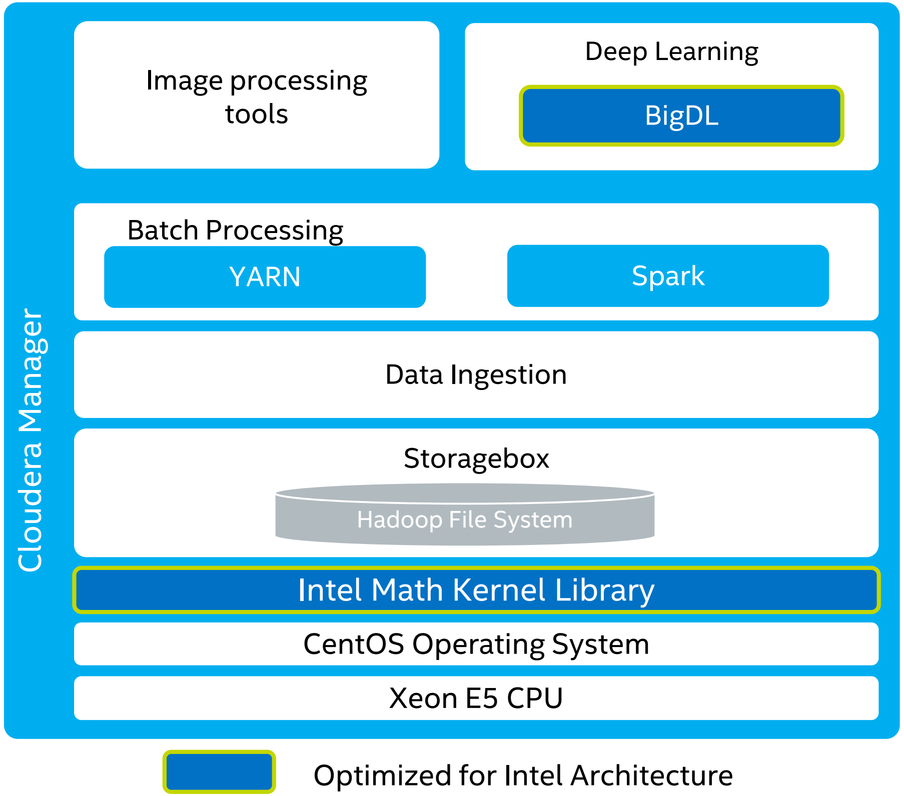
The input tensor is expected to be a 4D or 5D(with batch) tensor. This is also called "In-network Upsampling", "Full convolution", or "Upconvolution."

3D Augmentation: Image transformation library which is special for 3D images. The transformations include: cropping, rotating, affine transformation, etc.

Load Tensorflow UNet model

- Load pre-trained Tensorflow UNet model to extract features
- Train on other DL/ML model to train new models based on extracting features

BIGDL REFERENCE SOLUTION ARCHITECTURE

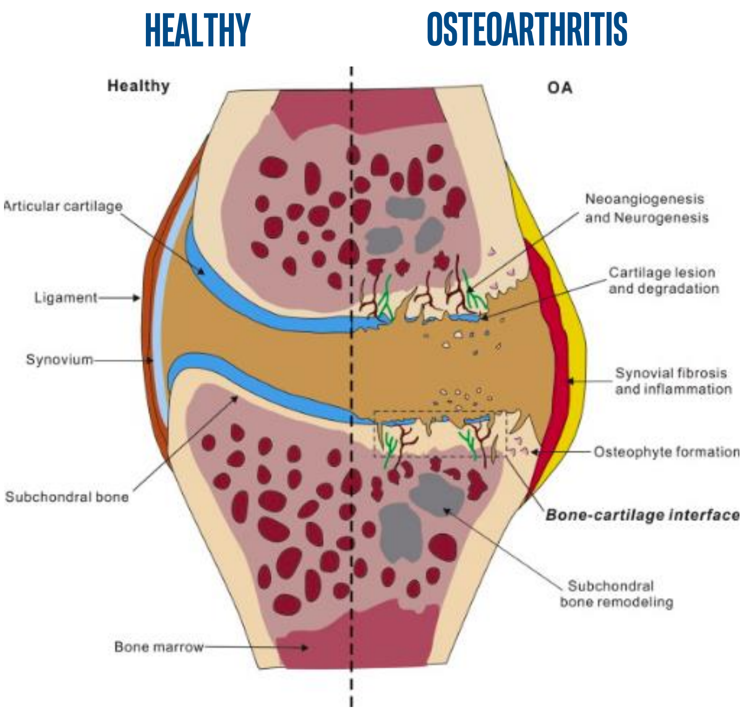


- BigDL 0.3
- CDH 5.9.0
  - Spark: 1 Spark master, 9 Spark worker
  - HDFS: 2 Name Node, 9 Data Node
- CentOS 7
- 10 Dell servers
  - 256G Ram, 6*8T HDs, 40 CPU cores, 4x10G Bandwidth

USE CASE

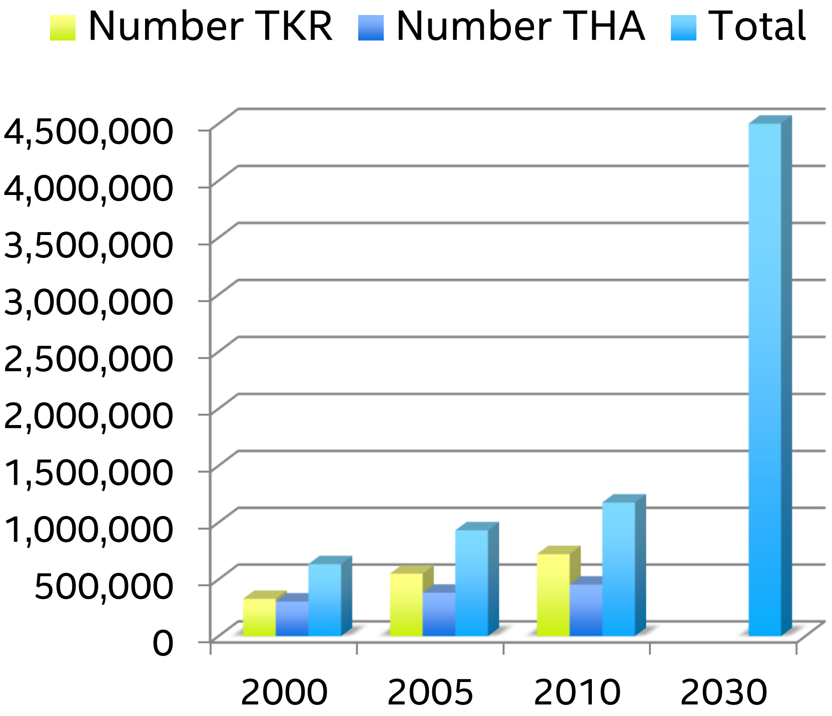
OSTEOARTHRITIS (1)

- Degenerative joint disease or “wear and tear” arthritis, osteoarthritis (OA) is **the most common chronic condition of the joints.**
- It occurs when the cartilage or cushion between joints breaks down leading to **pain**, stiffness and swelling.
- **52.5 million** (22.7%) adults - self-reported doctor-diagnosed OA
- **22.7 million** (9.8% adults) have arthritis-attributable **activity limitation.**



OSTEOARTHRITIS (2)

- Based on 2010-2012 data – **by 2040**
  - 78 million (26%) adults 18 years or older will have doctor-diagnosed OA
  - 35 million adults - arthritis-attributable activity limitations
- **Minority groups**, especially African-American and Hispanic individuals
  - At risk for poorer outcomes (such as pain and disability)
  - Less likely to undergo arthroplasty



POST TRAUMATIC OSTEOARTHRITIS

- ACL annual incidence was reported to be at least 81 per 100,000 in an age range between 10 and 64 years.
- ACL tears are a well-known risk factor for the development of early post-traumatic osteoarthritis (OA) in young active population

Danilo Gallinari, 25 years old, NBA Player

ACL surgical reconstruction

Nuggets – Magic **Gallinari scored 40 Points**

Radiographic Evidence of Osteoarthritis

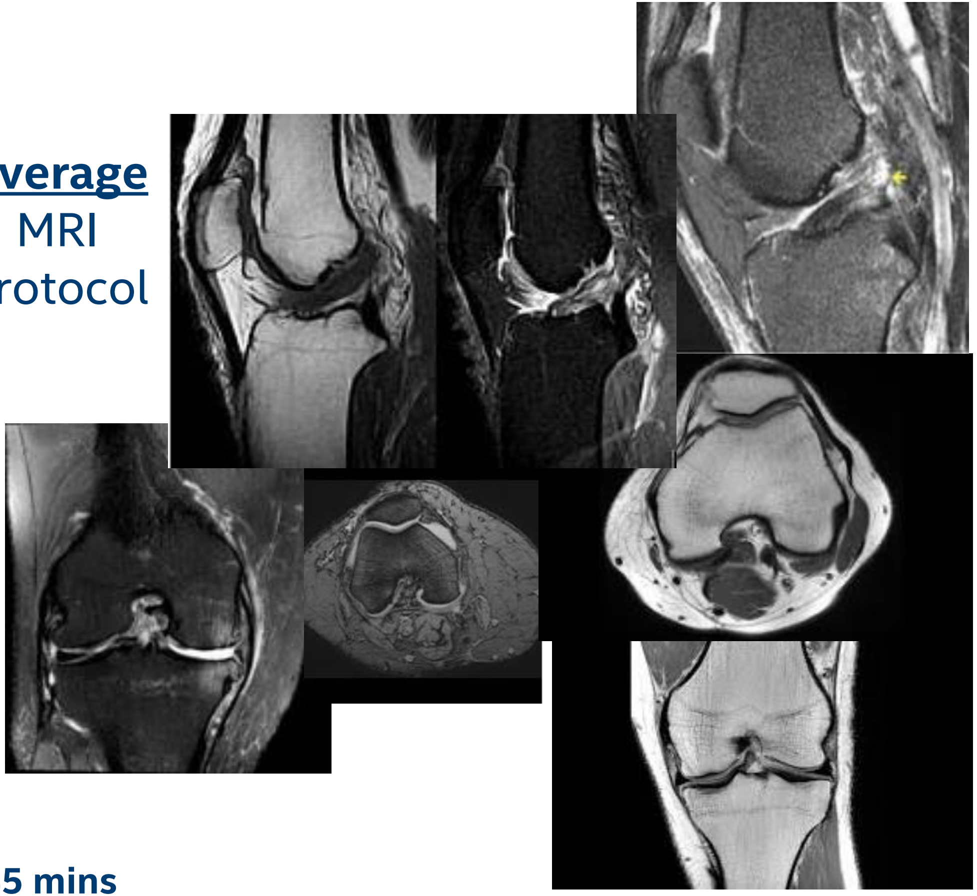
2013 2014 2015 ... 2020-30 **??**

IMAGING PRECISION MEDICINE

ACL
Injury

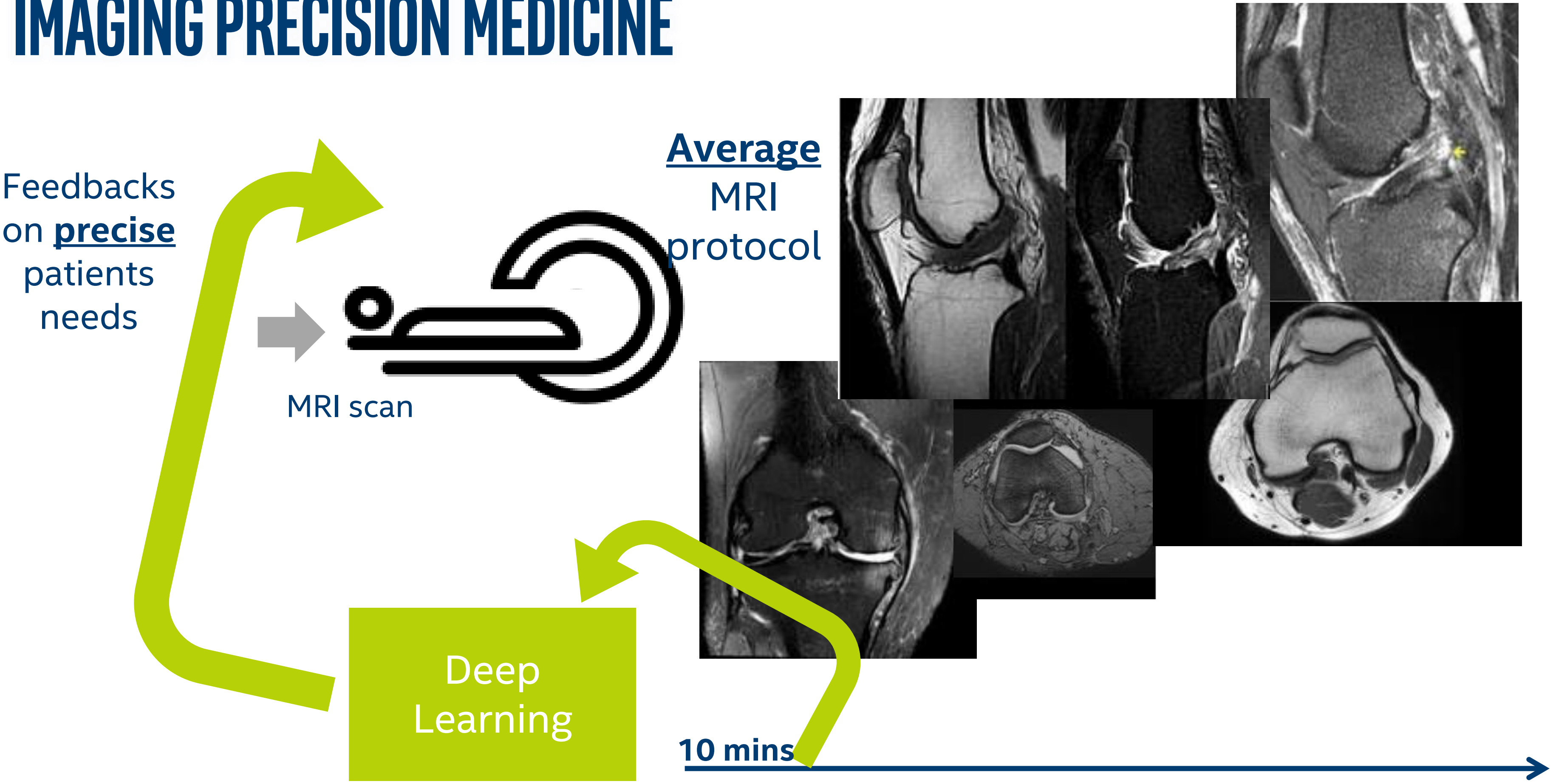
MRI scan

<u>Average</u>
MRI
protocol



45 mins

IMAGING PRECISION MEDICINE

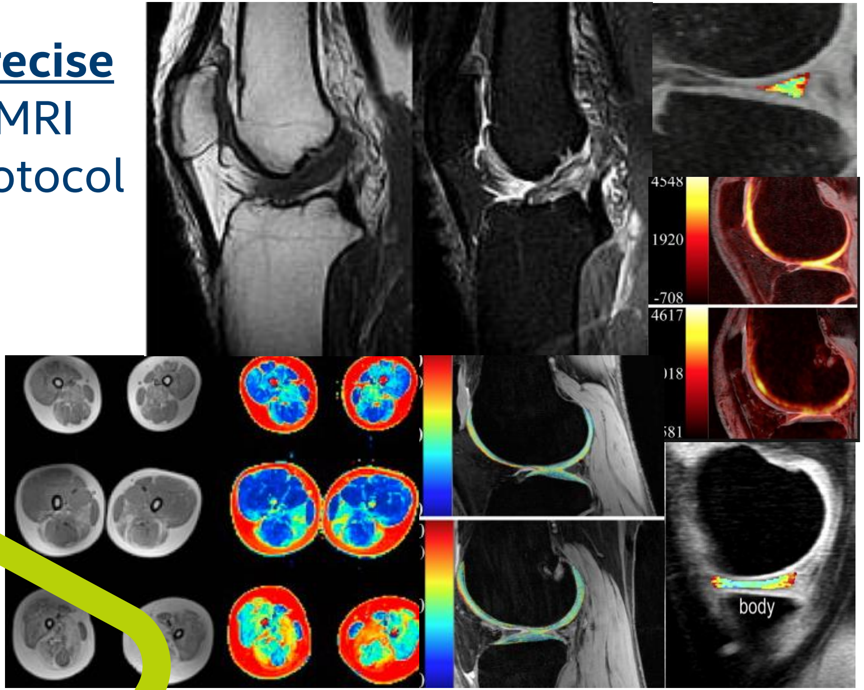


IMAGING PRECISION MEDICINE

Feedbacks on **precise** patients needs

MRI scan

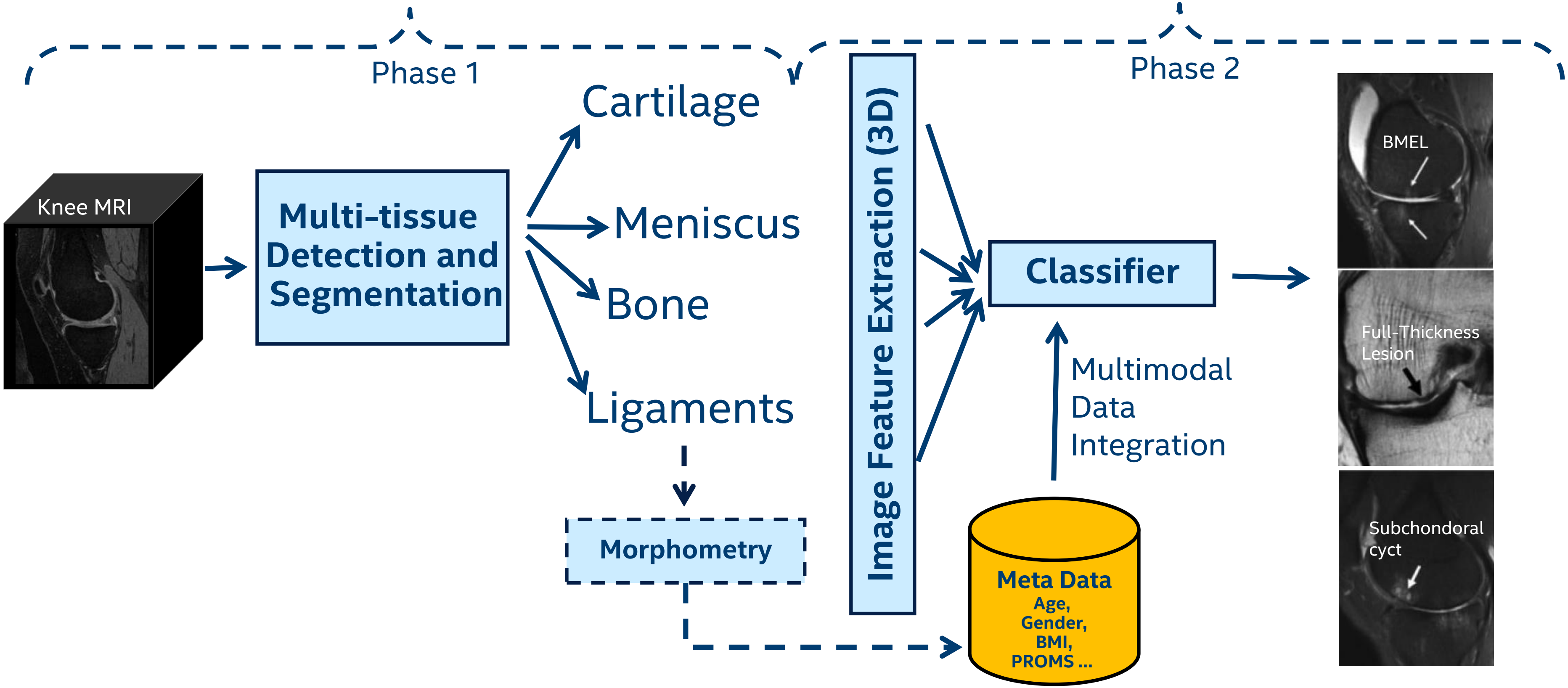
Precise MRI protocol



Deep Learning

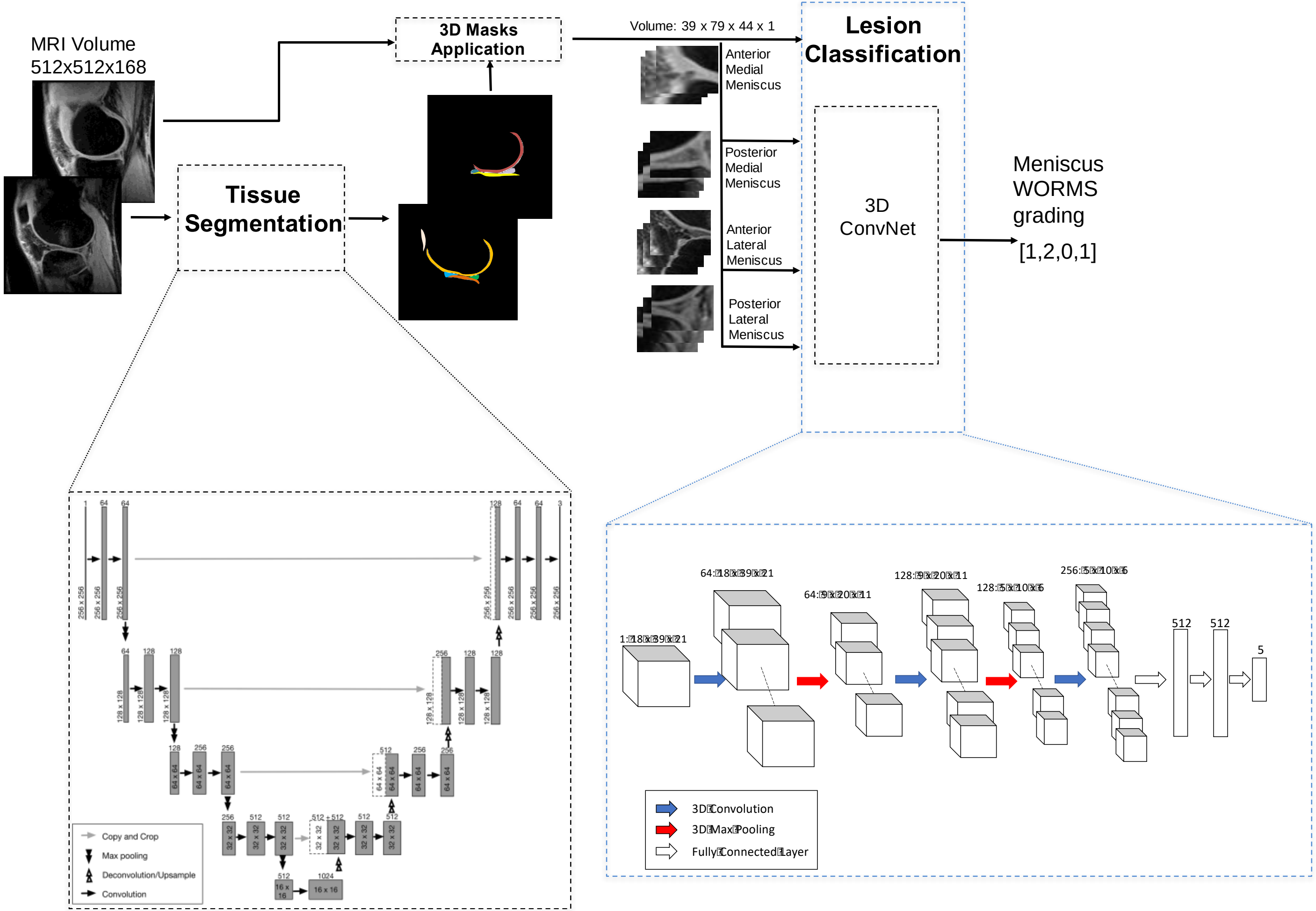
10 mins

CLASSIFY OA DEGENERATIVE FEATURES

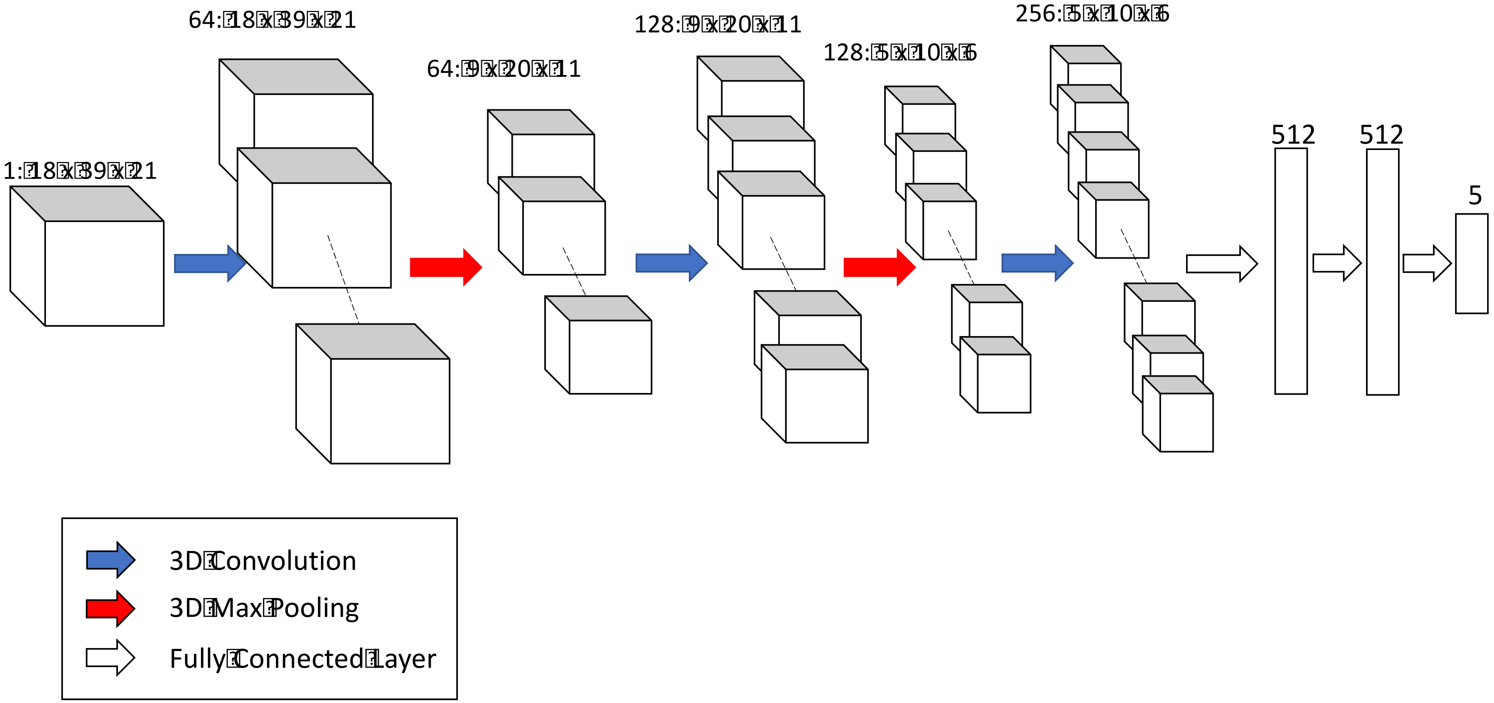


MODEL DEVELOPMENT

OVERALL PIPELINE



3D CNN FOR LESION DETECTION



PHASE 1 MULTI TISSUE SEGMENTATION: DATASET

Dataset 1: 464 3D MRI -> 11.136 2D images

<u>Parameters:</u> SPGR-T1weighted (TR/TE 9/2.6 ms, time of recovery 1500 ms, field of view 14 cm, matrix 256 × 128, slice thickness 4 mm, and bandwidth 62.5 kHz)

- Using for extraction of $T_{1\rho}$ and T_2 relaxation times
- 415 volumes

Subjects	OA (N=85)	Control (N=215)	ACL inj. (N=115)
Train	69	174	89
Test	16	41	26

Dataset 2: 174 3D MRI -> 29.540 2D images

<u>Parameters:</u> 3D-DESS (TE 4.7 ms, time of recovery TR 16.2 ms, field of view 14 cm, matrix 307 × 384, slice thickness 0.7 mm, and bandwidth 185 kHz).

- Using for extraction of volume and thickness
- 174 high resolution volumes

Subjects	OA (N=144)	Control (N=30)
Train	113	22
Test	31	8

PHASE 2 LESIONS DETECTION (MENISCUS): DATASET

- **1,478 knee MRI** subjects with and without osteoarthritis and after ACL injury were collected from three previous studies (age = 42.79±14.75 year, BMI = 24.28 ± 3.22 Kg/m^2, 48/52 male/female split) conducted on a GE 3T scanner.
- All studies used a high resolution 3D fast spin-echo (FSE) CUBE sequence TR/TE = 1500/26.69 ms, field of view = 14 cm, matrix = 384 x 384, slice thickness = 0.5 mm, bandwidth = 50.0 kHz).
- All the MRI volumes were graded using **WORMS** by MSK radiologists

WORMS Grade	0	1	2	3	4
Grading Description	Normal	Intra-substance abnormalities	Non-displaced tear	Displaced or complex tear	Complete destruction and maceration
Count (%)	4507 (76.23%)	735 (12.43%)	373 (6.31%)	192 (3.25%)	105 (1.78%)

Class 1 **Class 2** **Class 3**

RESULTS

RESULTS

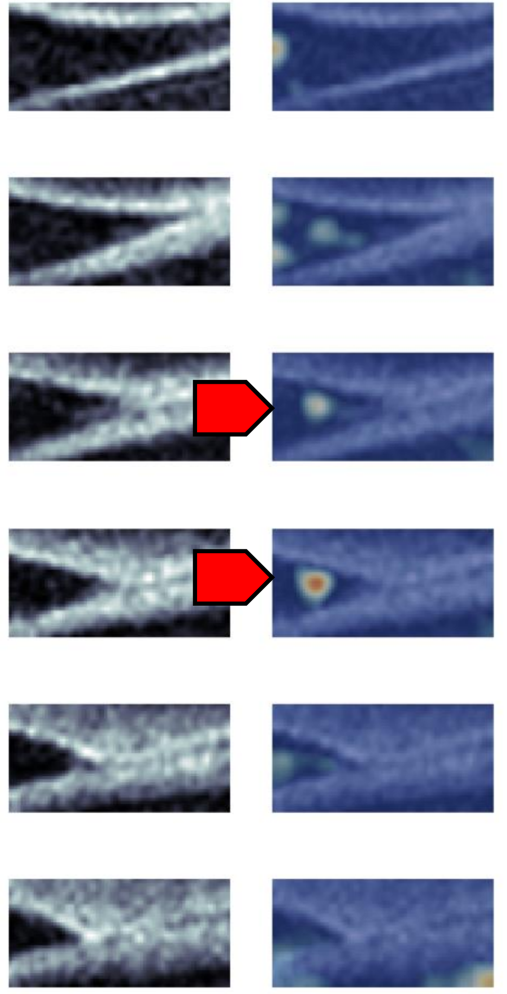
Intra-class correlation coefficients for intra-observer agreement for WORMS grading of the meniscus ranges from 0.801 to 0.928

Binary Model:

- Top model training accuracy: **No lesion: 98.5%, lesion: 99.7%**
- Top model testing accuracy: **No lesion: 94.1%, lesion: 67.6%**

Three Class Model:

- Top model training accuracy: No lesion: 99.9%, mild lesion: 100% , severe lesion: 99.9%
- Top model testing accuracy: **No lesion: 96.0%, mild lesion: 52.6%, severe lesion: 50.0%**

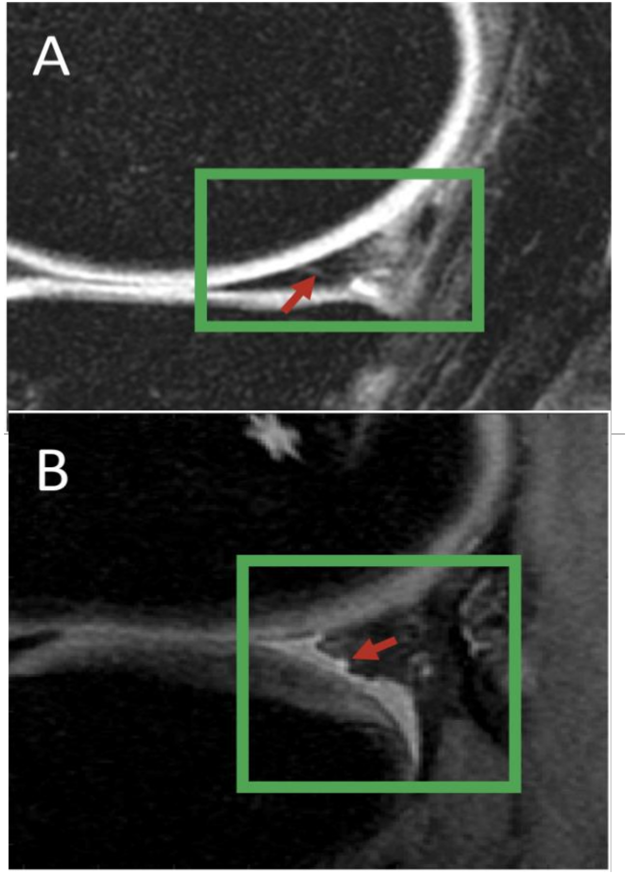


RESULTS VISUALIZATION

- A handful of the misclassified cases from the binary model were reviewed by a clinical radiologist to better understand why and if the model was incorrect.

- **For the majority of these cases, the radiologist agreed that there were features that could make the argument for switching the true grading to the predicted one**

MANUAL INSPECTION FROM CLINICAL RADIOLOGIST



This meniscus was graded as having no lesion but the model predicted there was one.

There does appear to be small lesion (indicated by the red arrow) that may extend to the surface which would classify it as a lesion

This meniscus was graded as having a lesion but the model predicted there was no lesion.
This meniscus is severely damaged and deformed so it was graded as having a complex tear even though a traditional looking complex tear did not exist.

SUMMARY

- We showed the feasibility of implementing 3D CNN in BigDL to analyze MRI scans and classify OA degenerative changes
- We plan to further implement 3D densely connected architectures and to study integration with demographics factors
- BigDL provides rich 3D imaging support and makes deep learning more accessible both for big data users and data scientists
- BigDL leverages existing Spark/Hadoop infrastructure to run deep learning applications on the same cluster where the data is stored

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