# What is Analytics Zoo

BigDL

Distributed, High-Performance

Deep Learning Framework

for Apache Spark

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

ANALYTICS ZOO

Unified Analytics + AI Platform

Distributed TensorFlow, Keras, PyTorch and BigDL on Apache Spark

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

**Accelerating Data Analytics + AI Solutions At Scale**

ANALYTICS ZOO

# 用ZOO实现基于深度学习的胸腔疾病AI诊疗辅助

***Analytics Zoo**: Building Unified Big Data Analytics and AI Pipelines*

龚奇源 (Qiyuan Gong)
机器学习与数据隐私专家

2020/04/30

# Agenda

## Background

- Big data & Deep Learning
- Analytics Zoo

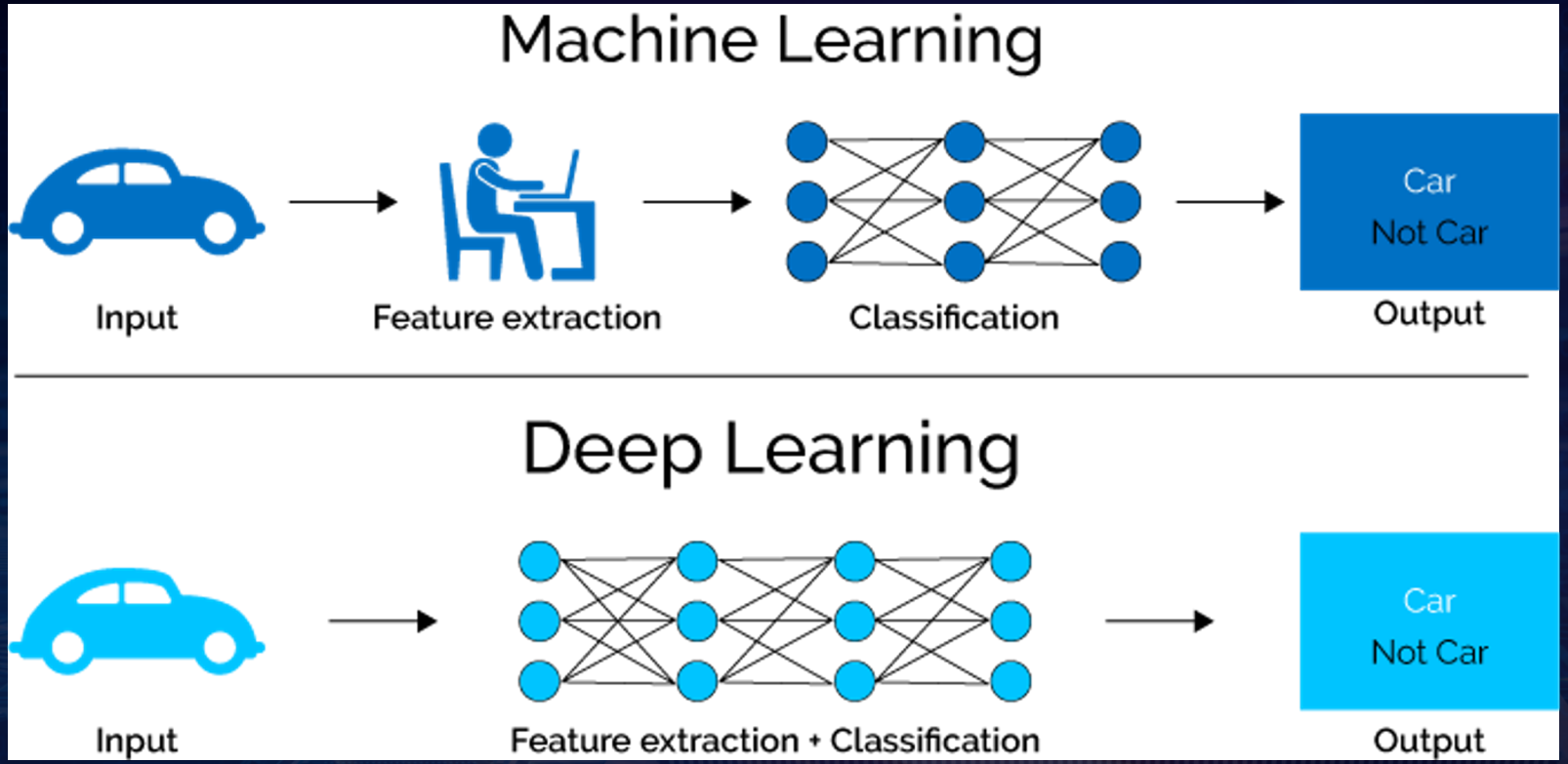
## AI-assisted Radiology Using Apache Spark and Analytics Zoo

- Motivation & Problem statement
- Training Chest X-ray Model
- Inference & Serving

## Q&A

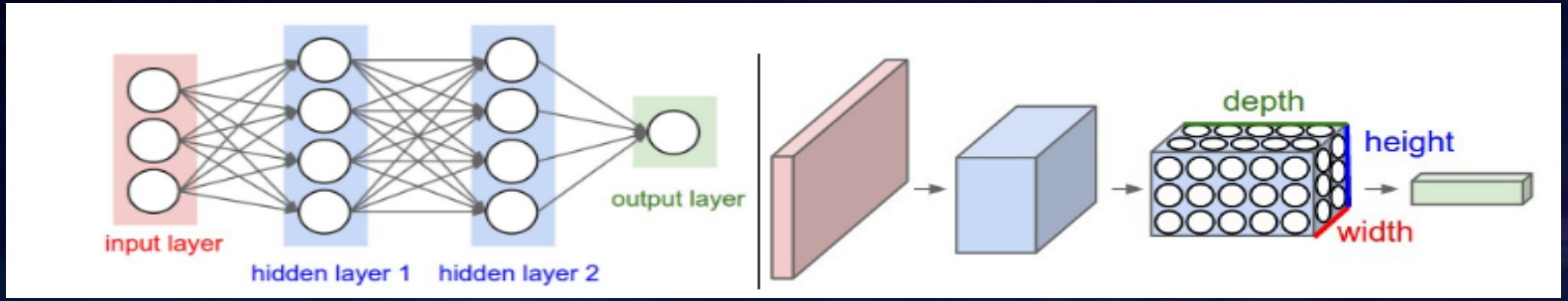
# Background

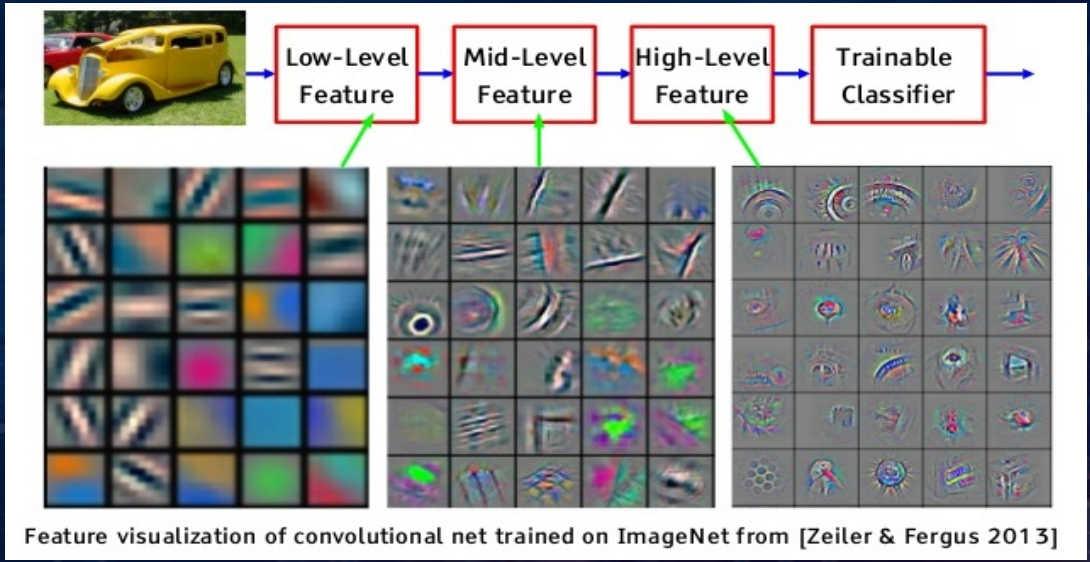
# Deep Learning




https://www.quora.com/What-is-the-difference-between-deep-learning-and-usual-machine-learning

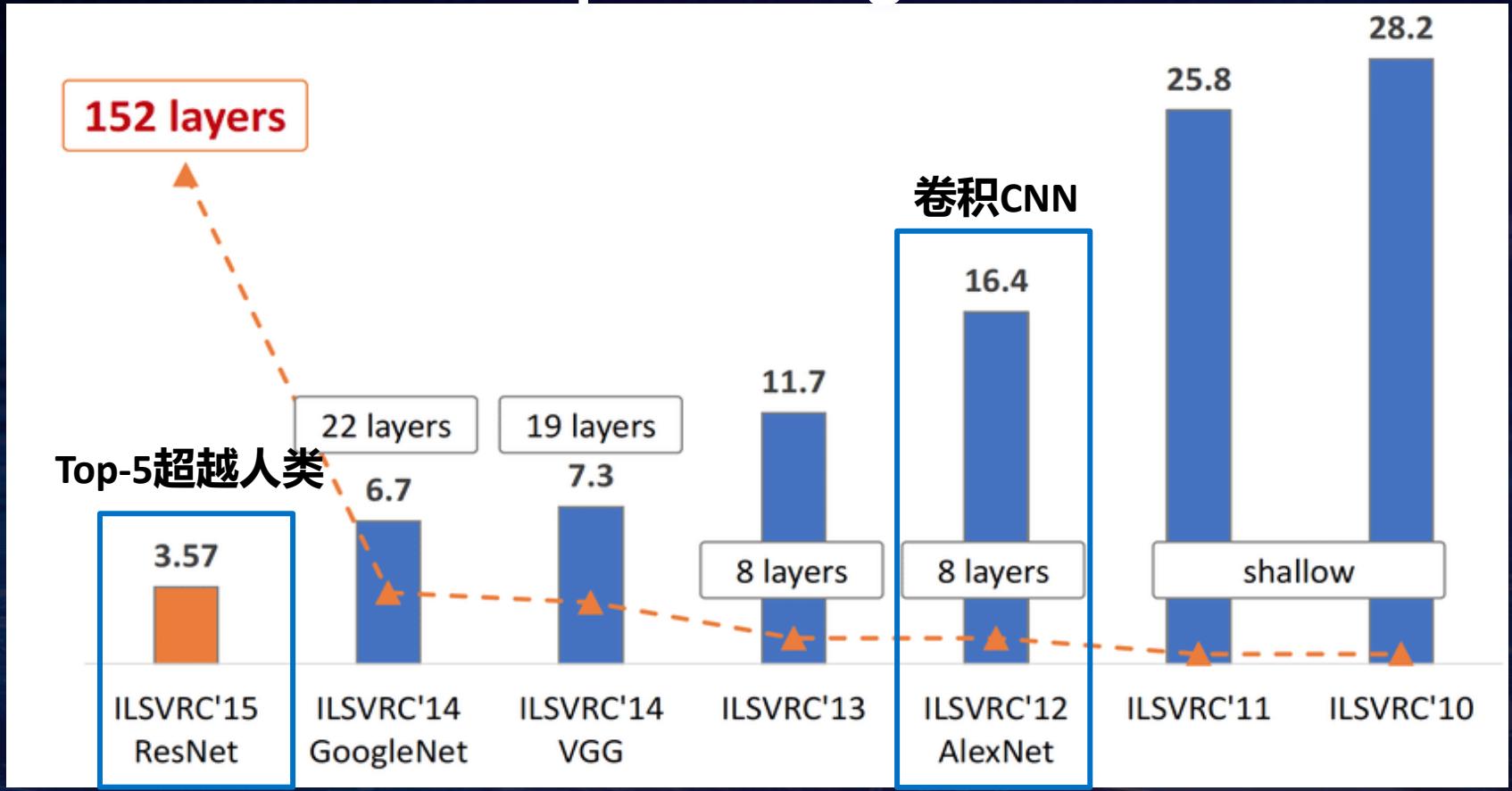
# Deep Learning



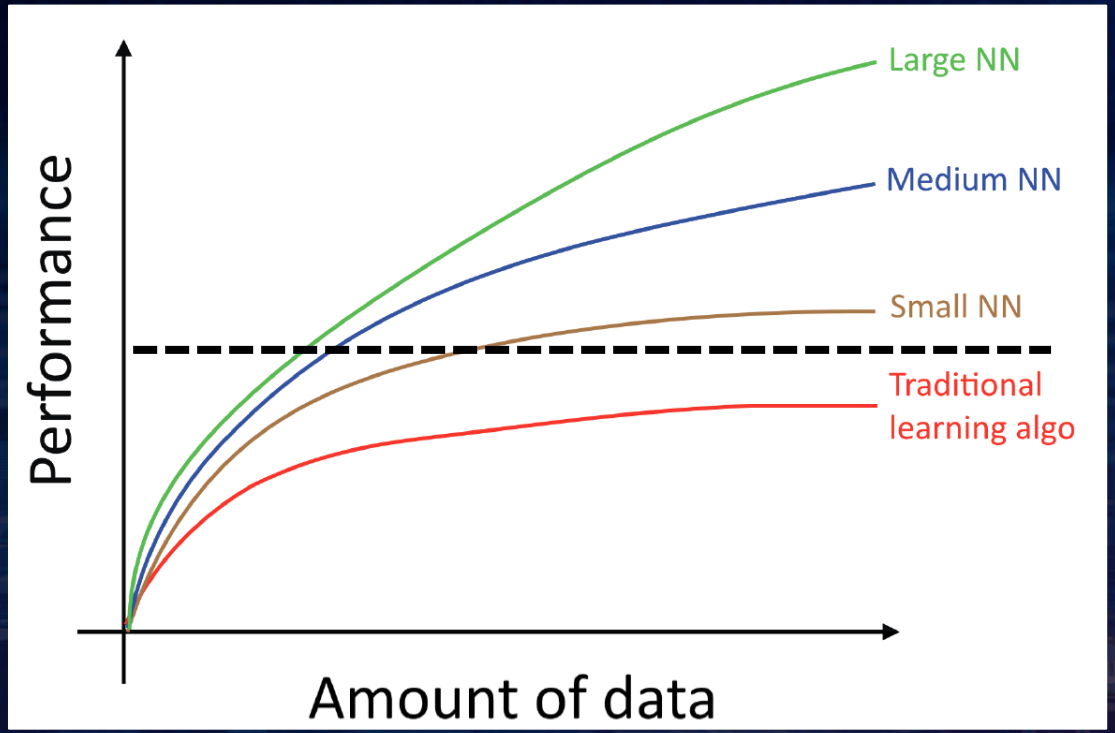

Feature visualization of convolutional net trained on ImageNet from [Zeiler & Fergus 2013]

# Deep Learning - CV

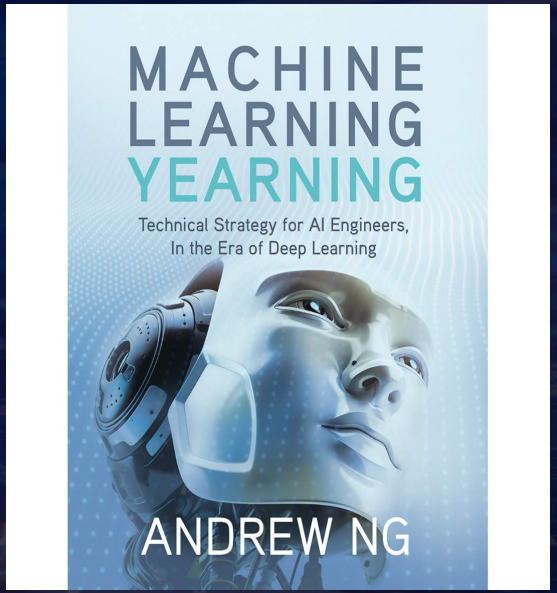



Kaiming etc Deep Residual Learning for Image Recognition,2015

# Deep Learning




"Machine Learning Yearning", Andrew Ng, 2016



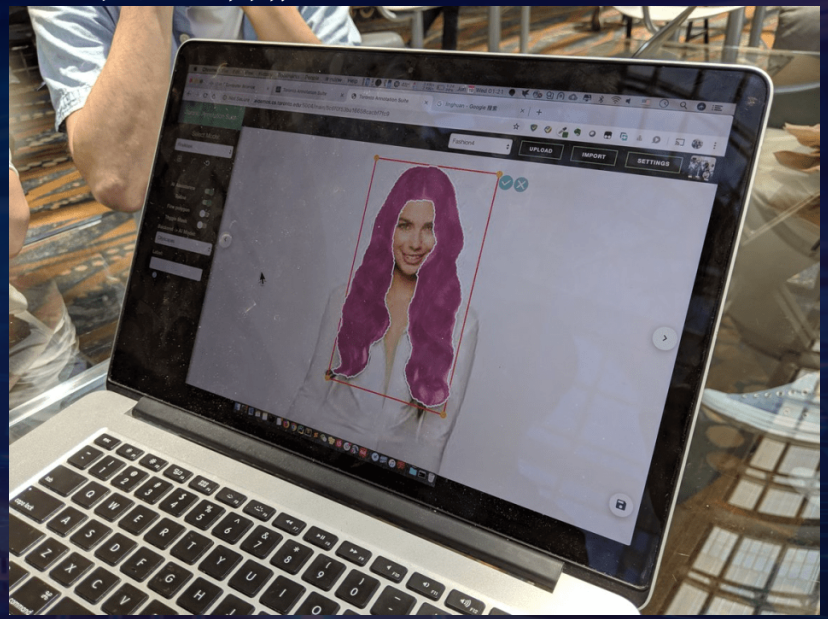
# Deep Learning

但天下没有免费的午餐 (no free launch)

- Deep Learning需要大量算力 (计算密集)
- Deep Learning需要大量数据 (data hungry)

足够的存储和算力

“人工”智能




https://medium.com/syncedreview/data-annotation-the-billion-dollar-business-behind-ai-breakthroughs-d929b0a50d23

# Deep Learning

**2016年 Google Alpha Go 击败李世石**

- 现场的Alpha Go所使用的计算资源
  - **48 CPU, 8 GPU ≈ 24 Servers**

- 其实还有一个Distributed Alpha Go
  - **1202 CPU, 176 GPU ≈ 601 Servers**

https://www.nature.com/articles/nature16961

https://newsroom.intel.com/editorials/re-architecting-data-center-intel-xeon-processor-scalable-family/#gs.hi35lo

# Real-World ML/DL Applications Are Complex Data Analytics Pipelines

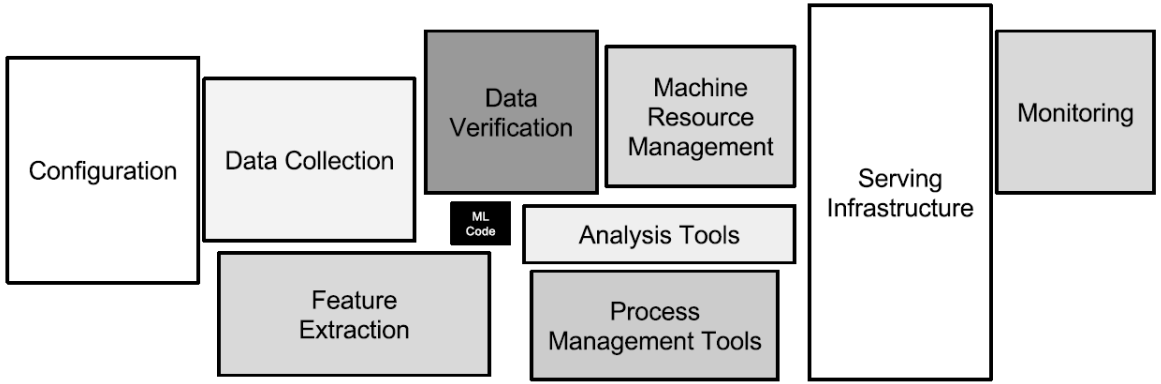



Figure 1: Only a small fraction of real-world ML systems is composed of the ML code, as shown by the small black box in the middle. The required surrounding infrastructure is vast and complex.

"Hidden Technical Debt in Machine Learning Systems", Sculley et al., Google, NIPS 2015 Paper

# Deep Learning

获取 / 存储 → 清洗 / 准备 → 分析 / 建模 → 部署 / 可视化

集成的数据流水线














数据管理

数据分析

数据科学及人工智能

# Unified Data Analytics and AI Platform

## Seamless Scaling from Laptop to Distributed Big Data


- Easily prototype end-to-end pipelines that apply AI models to big data
- "Zero" code change from laptop to distributed cluster
- Seamlessly deployed on production Hadoop/K8s clusters
- Automate the process of applying machine learning to big data

# Analytics Zoo

## Unified Analytics + AI Platform for Big Data

**Models & Algorithms**

- Recommendation
- Time Series
- Computer Vision
- NLP

**Automated ML Workflow**

- AutoML for Time Series
- Automatic Cluster Serving

**Integrated Analytics & AI Pipelines**

- Distributed TensorFlow & PyTorch on Spark
- RayOnSpark
- Spark Dataframes & ML Pipelines for DL
- InferenceModel

**Compute Environment**

- Laptop
- K8s Cluster
- Hadoop Cluster
- Spark Cluster

- DL Frameworks (TF/PyTorch/OpenVINO/...)
- Distributed Analytics (Spark/Flink/Ray/...)
- Python Libraries (Numpy/Pandas/sklearn/...)

**Powered by oneAPI**

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

# AI-assisted Radiology Using Apache Spark and Analytics Zoo


AI-assisted Radiology Using Distributed Deep Learning on Apache Spark and Analytics Zoo

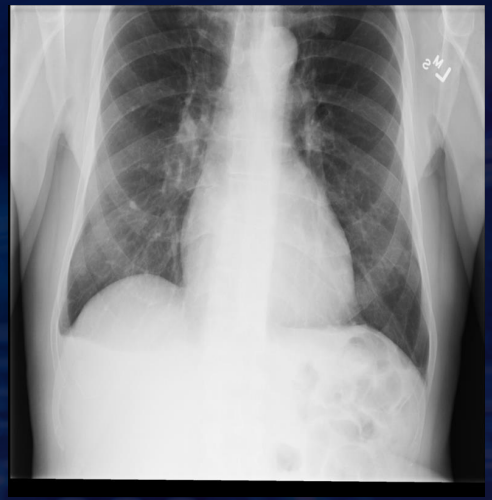
Using Deep Learning on Apache Spark to diagnose thoracic pathology from chest X-rays

https://github.com/dell-ai-engineering/BigDL-ImageProcessing-Examples

# Motivation

## 胸透X光的拍摄速度远远超过了阅片速度

- 拍片N秒钟
- 阅片N分钟
  - 需要有经过培训的医护人员
  - 需要专用设备
  - 长期重复劳动



## Worst case

In the United Kingdom, the care quality commission recently reported that – over the preceding **12 months – a total of 23,000 chest X-rays (CXRs) were not formally reviewed** by a radiologist or clinician at Queen Alexandra Hospital alone

https://www.nature.com/articles/s41598-019-42294-8

# Motivation

## Pain points

- Review X-ray is a bottleneck during X-ray related diagnosis
  - Long analysis time
  - Need radiologist or clinician
  - Duplicated hard work

## Can we use deep learning to accelerate X-ray analysis?

- Do we have training dataset? (YES, ChestX-ray14)
- What/which we can do?
  - Multiple-label image classification (YES)
  - Object detection (No, dataset don't have enough annotations)
  - Image segmentation (No, dataset don't have enough annotations)

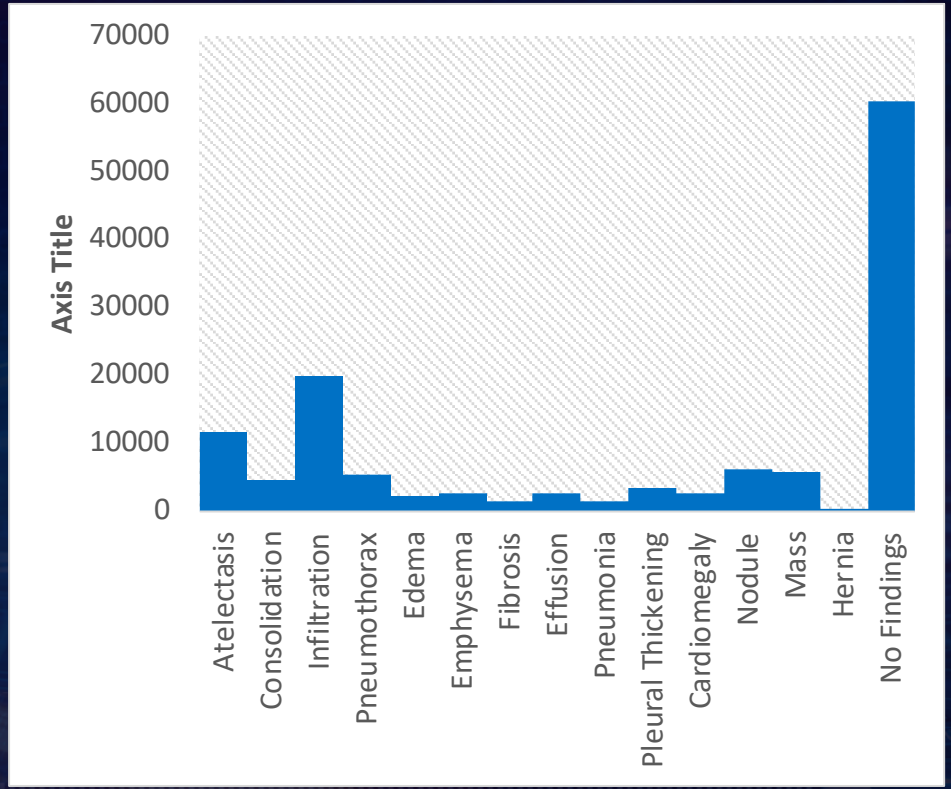
# Dataset: ChestX-ray14 from NIH

## Open dataset ChestX-ray14 from NIH

- 112,120 images from over 30000 patients
- Multi label (14 diseases)

```
00000013_005.png,Emphysema | Infiltration | Pleural_Thickening
00000013_006.png,Effusion|Infiltration
00000013_007.png,Infiltration
00000013_008.png,No Finding
```

- Unbalanced datasets
  - Close to 50% of the images have 'No findings'
  - Infiltration get the most positive samples (19894) and Hernia get the least positive samples (227)




https://www.nih.gov/news-events/news-releases/nih-clinical-center-provides-one-largest-publicly-available-chest-x-ray-datasets-scientific-community
https://www.kaggle.com/nih-chest-xrays/data

# Related works

## ChestX-Ray8 (published ChestXray14)

- Transfer learning with Alexnet, Googlenet, Resnet50
- Resnet50 perform best

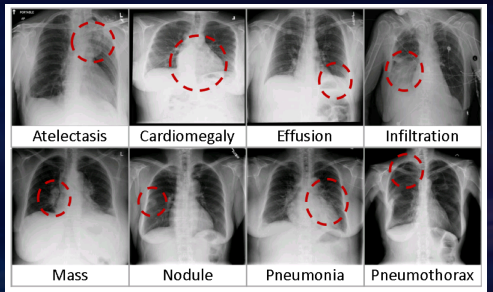
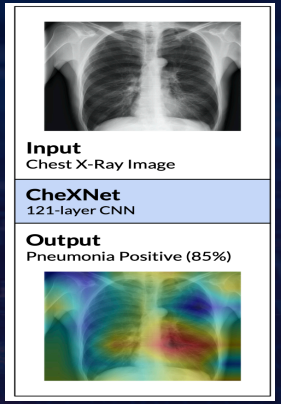



Figure 1. *Eight common thoracic diseases observed in chest X-rays that validate a challenging task of fully-automated diagnosis.*

## CheXNet (by Stanford, Andrew NG etc)

- Densenet121 for Pneumonia & multi-label
- AI for Medical Diagnosis




## Others (check references)

ChestX-ray8: Hospital-scale Chest X-ray Database and Benchmarks on Weakly-Supervised Classification and Localization of Common Thorax Diseases

https://stanfordmlgroup.github.io/projects/chexnet/

https://www.coursera.org/learn/ai-for-medical-diagnosis/

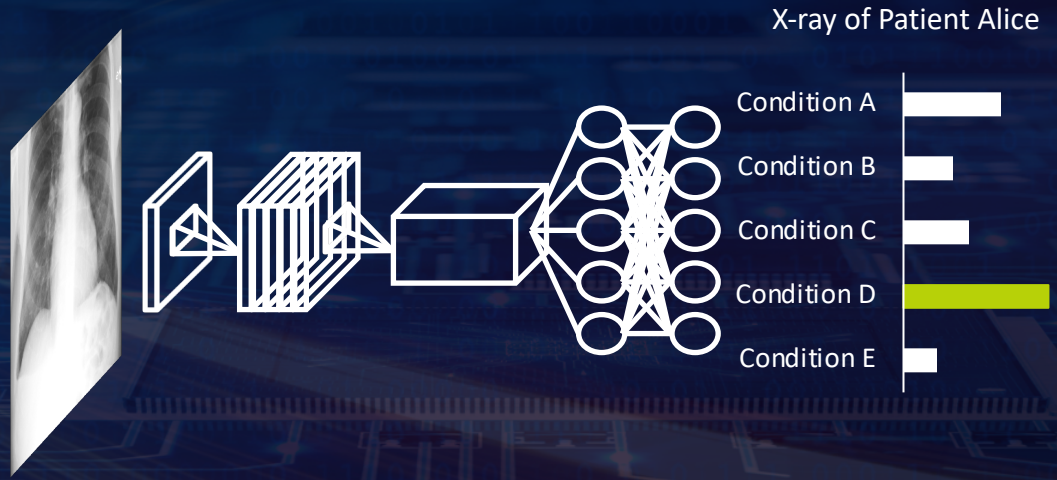
# Predicting diseases in Chest X-rays

## Develop a ML pipeline in Apache Spark and train a deep learning model to predict disease in Chest X-rays

- An integrated ML pipeline with Analytics Zoo on Apache Spark
- Demonstrate feature engineering and transfer learning APIs in Analytics Zoo
- Use Spark worker nodes to train at scale

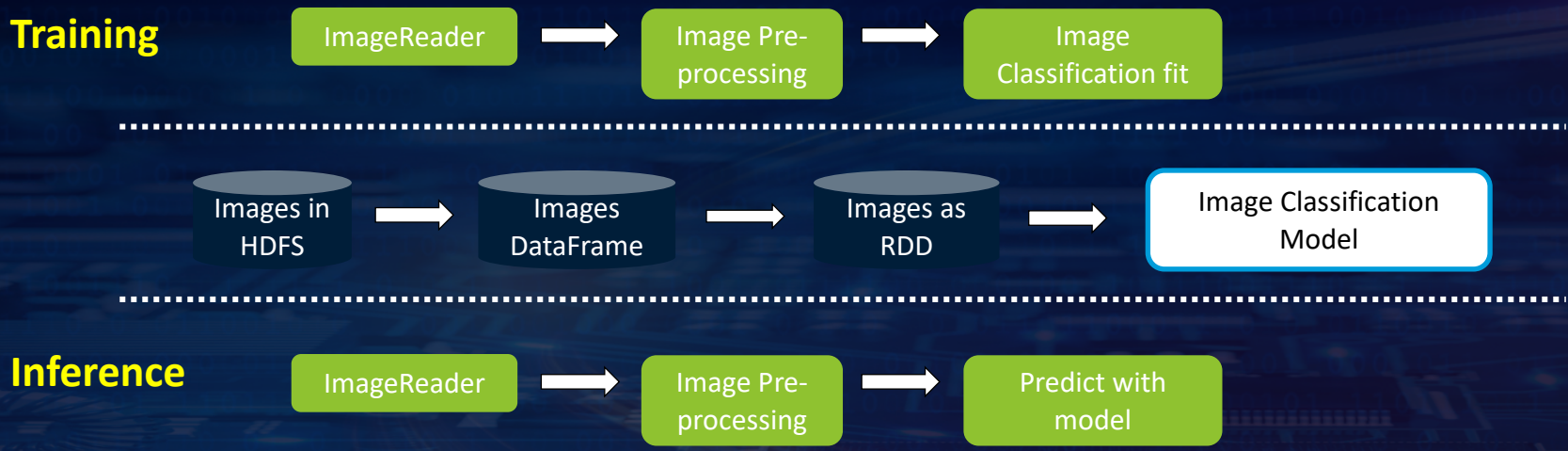
## AI-assisted Radiology

- Give real-time Suggestions
- Review results




ChestX-ray8: Hospital-scale Chest X-ray Database and Benchmarks on Weakly-Supervised Classification and Localization of Common Thorax Diseases

# Data Pipeline

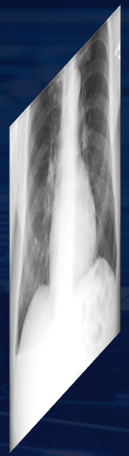


# Training Neural networks

```
def training(images, labels):
    # Deep Learning
    return model
```

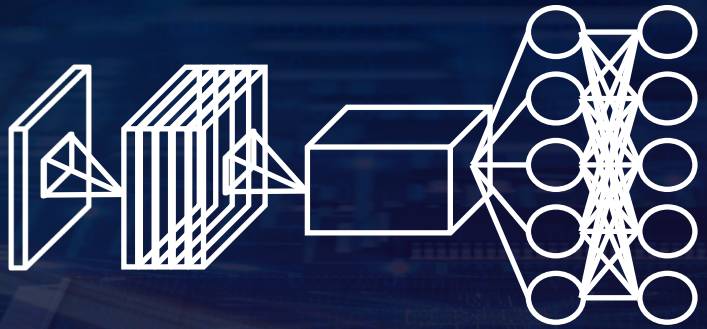
Training dataset

image label

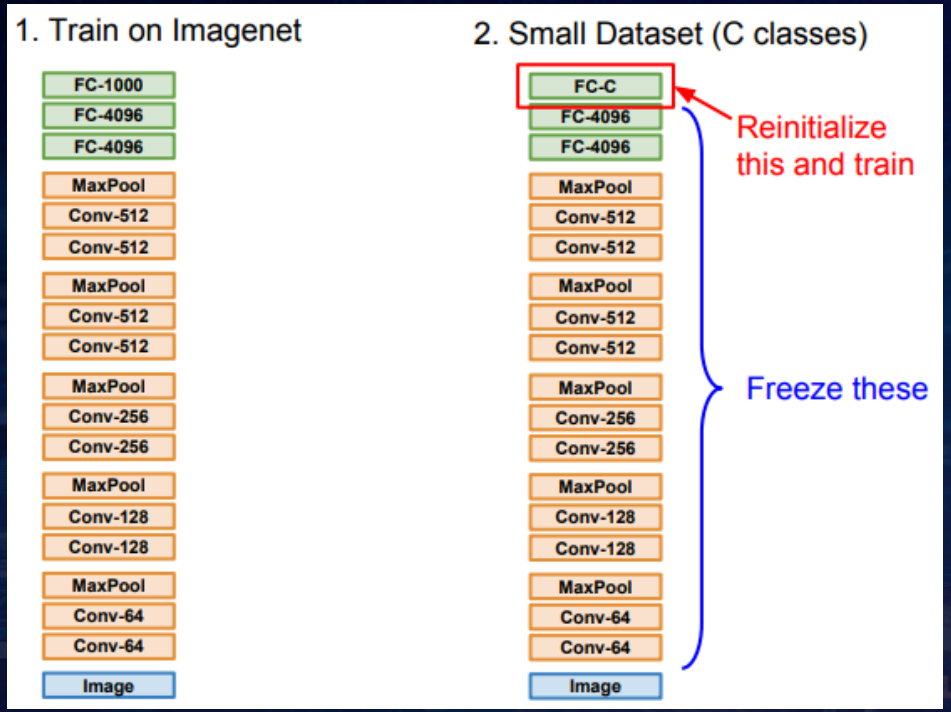


Condition A
Condition B
Condition C

model



# Transfer Learning with Pre-trained models



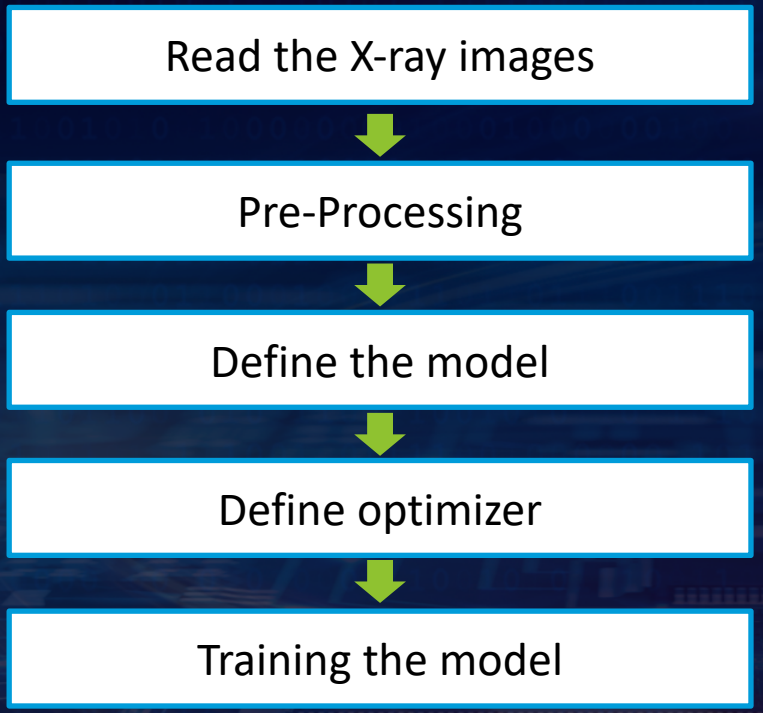

**Transfer Learning**

- Less data requirement
- Less computation and less training time

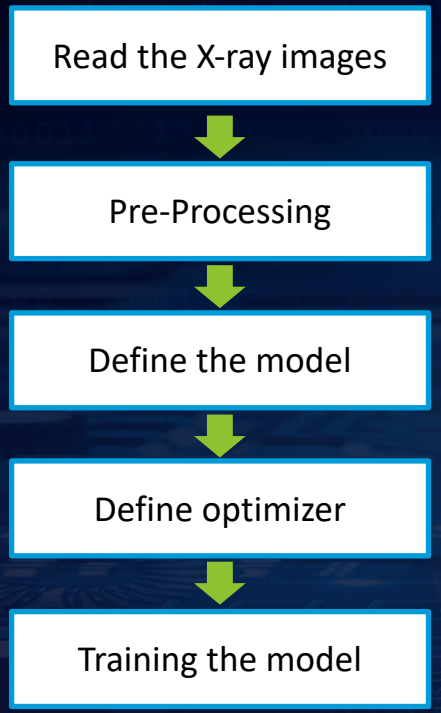
| | Similar dataset | Different dataset |
|---|---|---|
| Small datasaet | Replace top layer | Tough work |
| Large dataset | Finetune a few layers | Finetune more layers |

http://cs231n.stanford.edu/slides/2017/cs231n_2017_lecture7.pdf

# Training: Building the X-ray Model



# Read the X-ray images as Spark DataFrames



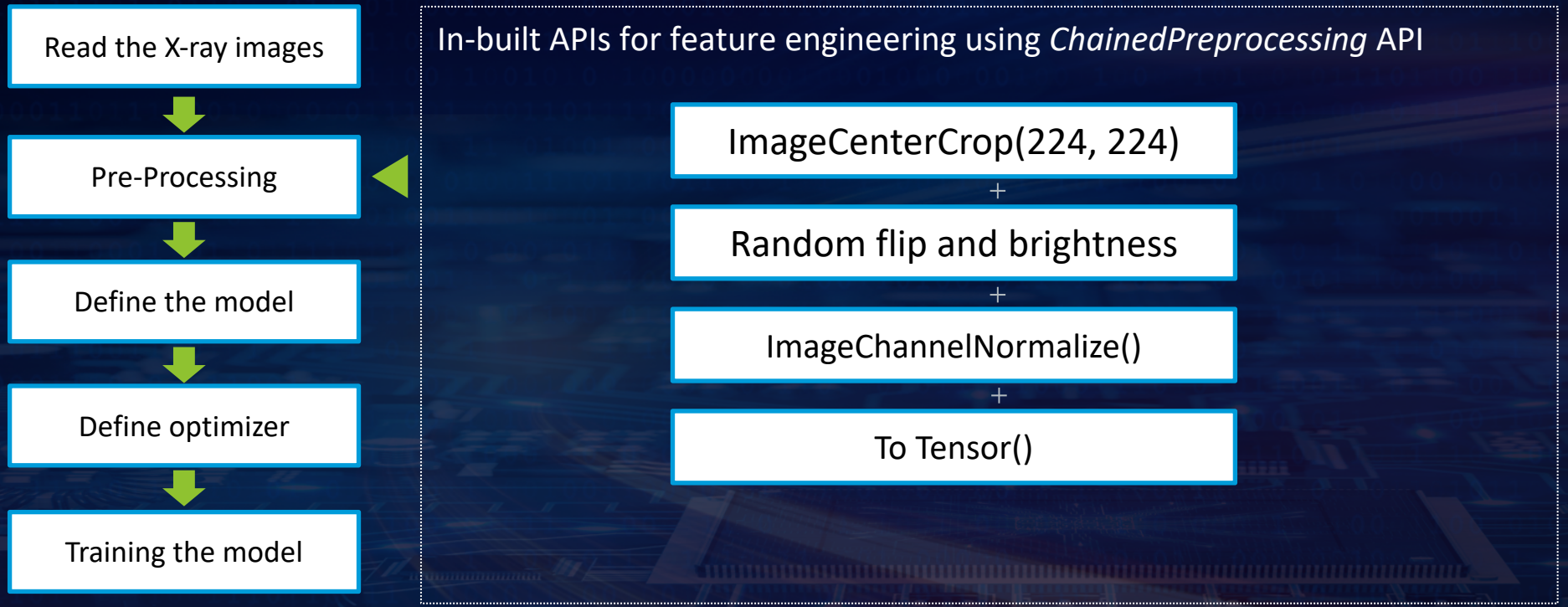

- Initialize *NNContext* and load Xray images into *DataFrames* using *NNImageReader*

```
from zoo.pipeline.nnframes import NNImageReader
imageDF = NNImageReader.readImages(image_path, sc,
                    resizeH=256, resizeW=256, image_codec=1)
```

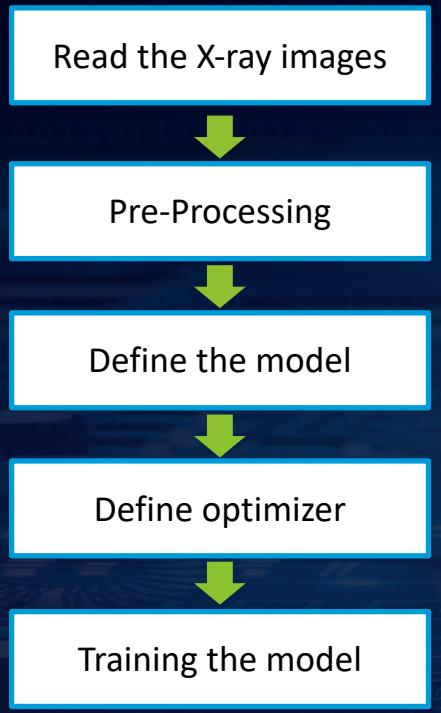
- Process loaded X-ray images and add labels (another DataFrame) using Spark transformations

```
trainingDF = imageDF.join(labelDF, on="Image_Index",
                                  how="inner")
```

# Feature Engineering – Image Pre-processing

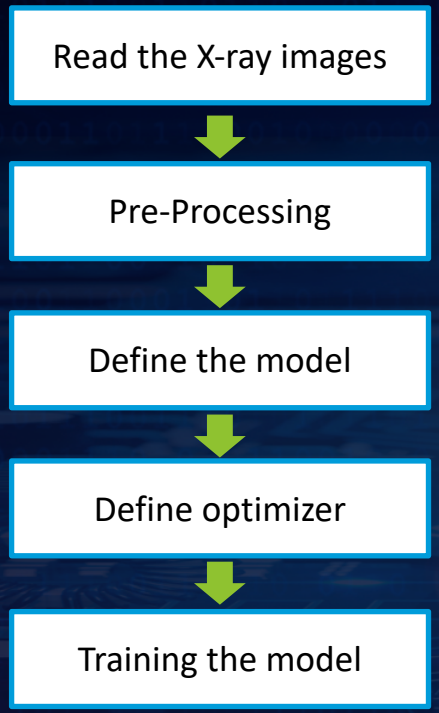


# Defining the model with Transfer Learning APIs



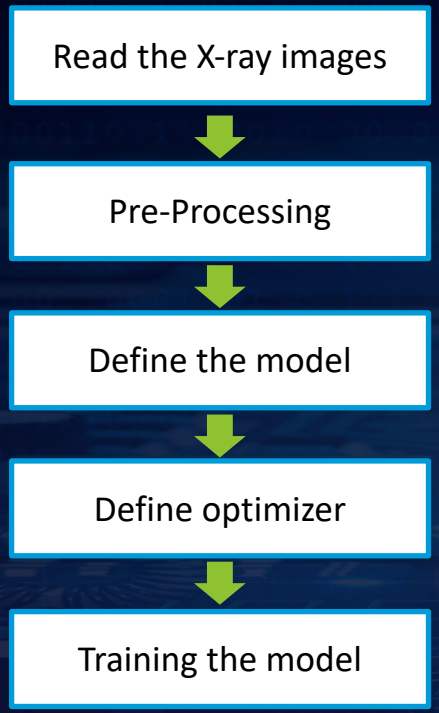

- Load a pre-trained model using *Net.load_bigdl*. The model is trained with ImageNet dataset
  - ResNet 50
  - DenseNet
  - Inception
- Remove the final *softmax* layer of ResNet-50
- Add new input (for resized x-ray images) and output layer (to predict the 14 diseases). Activation function is *Sigmoid*
- Avoid overfitting
  - Regularization
  - Dropout

# Define the Optimizer




- Evaluated two optimizers: SGD and Adam Optimizer
- Learning rate scheduler is implemented in two phases:
  - Warmup + Plateau schedule
  - Warmup: Gradually increase the learning rate for 5 epochs
  - Plateau: Plateau("**Loss**", factor=0.1, patience=1, mode="**min**", epsilon=0.01, cooldown=0, min_lr=1e-15)

# Train the model using ML Pipelines




- Analytics Zoo API *NNEstimator* to build the model
- *.fit()* produces a neural network model which is a Transformer
- You can now run *.predict()* on the model for inference
- AUC-RoC is used to measure the accuracy of the model. Spark ML pipeline API *BinaryClassificationEvaluator* to determine the AUC-ROC for each disease.

```
estimator = NNEstimator(xray_model, BinaryCrossEntropy(), transformer)
            .setBatchSize(batch_size).setMaxEpoch(num_epoch)
            .setFeaturesCol("image").setCachingSample(False).
            .setValidation(EveryEpoch(), validationDF, [AUC()], batch_size)
            .setOptimMethod(optim_method)
Xray_nnmodel = estimator.fit(trainingDF)
```

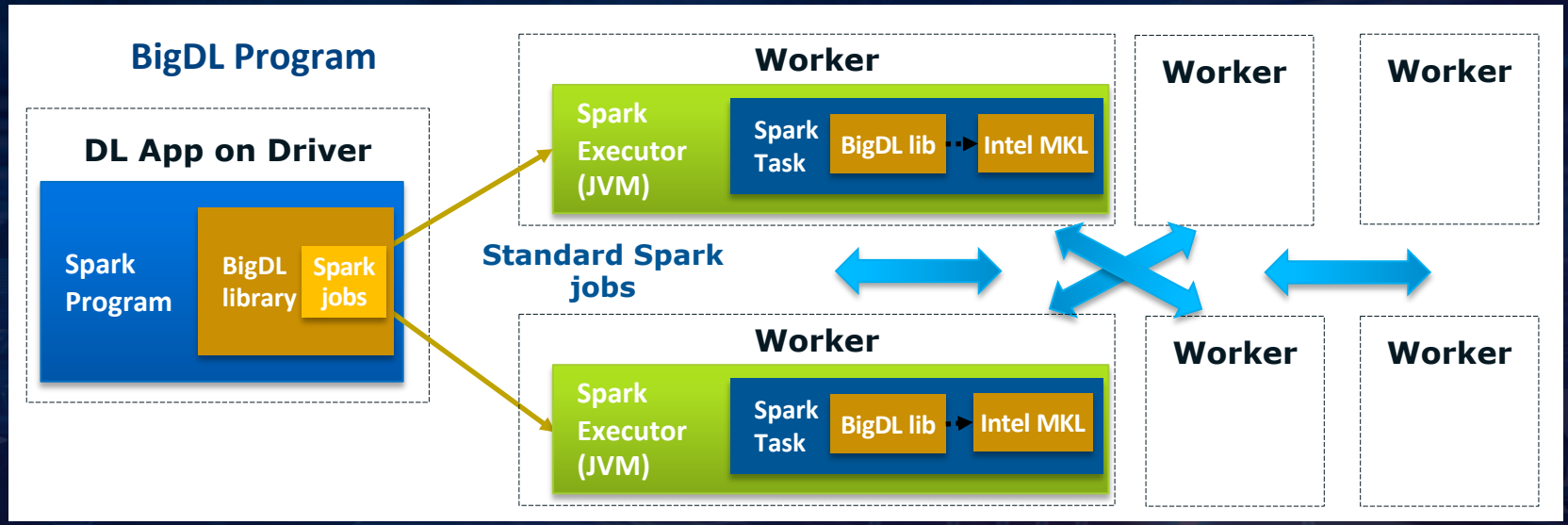
# Distributed training in BigDL

## Standard Spark jobs

- No changes to the Spark or Hadoop clusters needed

## Data parallel

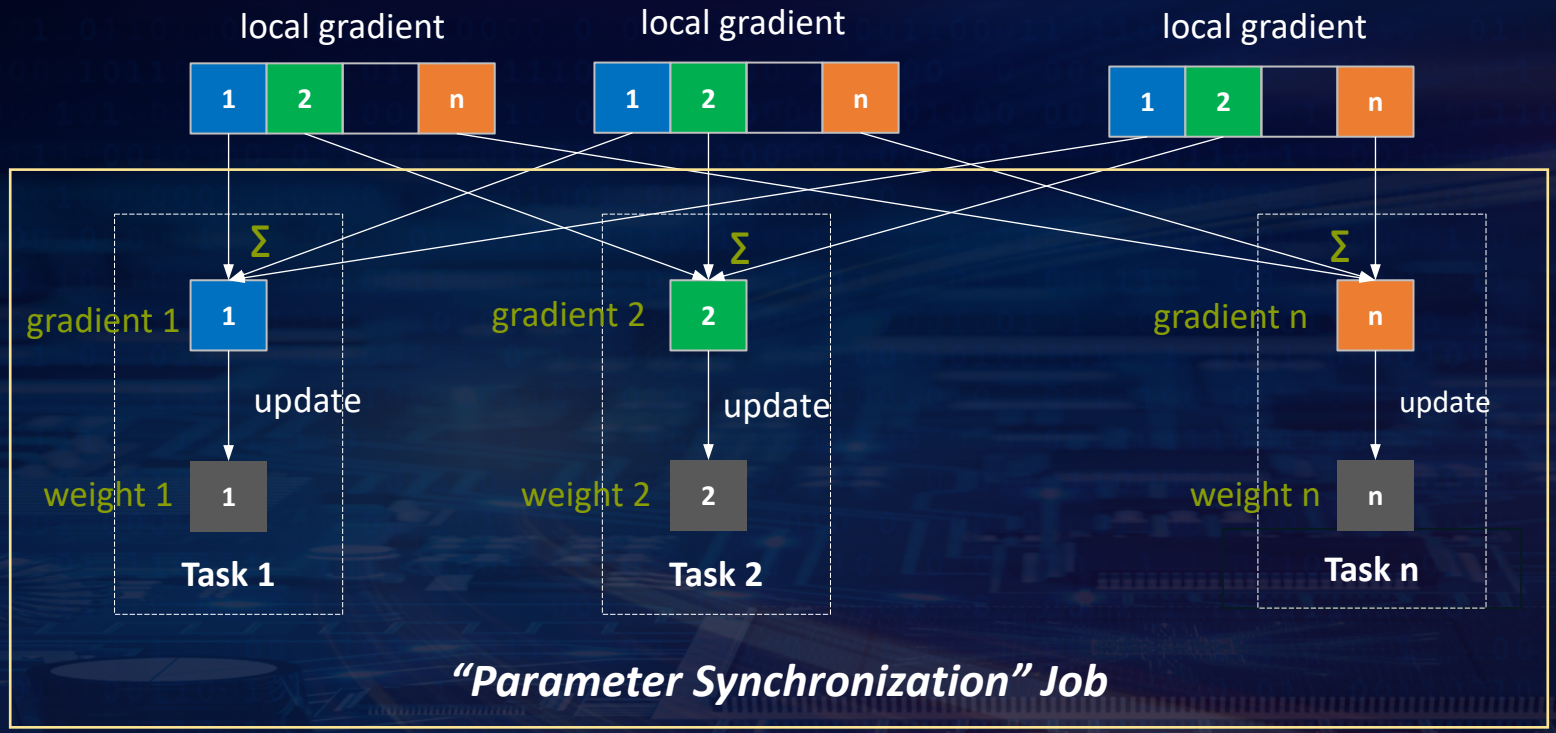
- Each Spark task runs the same model on a subset of the data (batch)



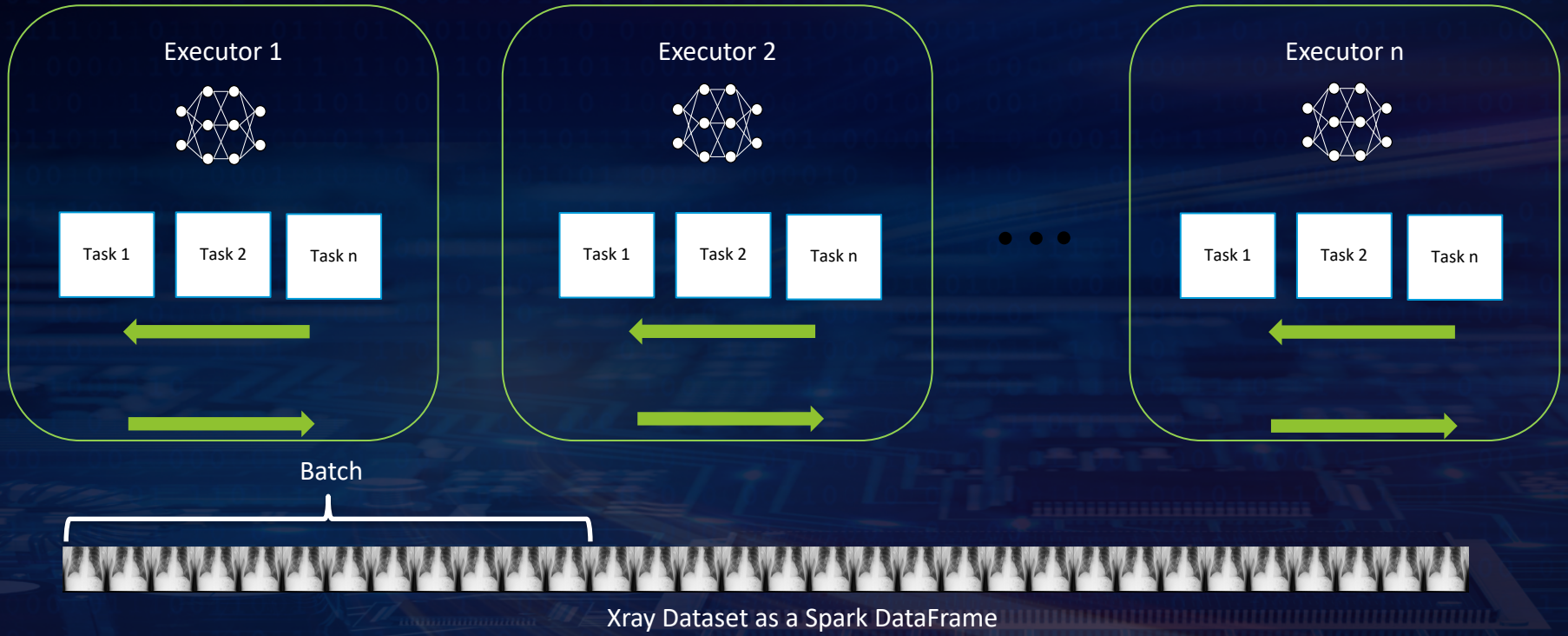

Dai, Jason Jinquan, et al. Bigdl: A distributed deep learning framework for big data. *Proceedings of the ACM Symposium on Cloud Computing*. 2019.

# Distributed training in BigDL

**Allreduce**



# Distributed training in BigDL



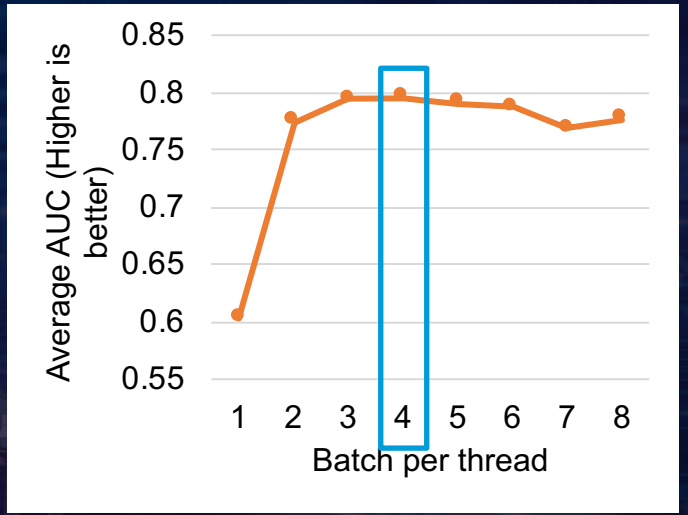
# Results and observations

## Spark Parameter recommendations

- Number of executors ⇒ Number of compute nodes
- Number of executor cores ⇒ Number of physical cores minus two
- Executor memory ⇒ Number of cores per executor * (Memory required for each core + data partition)

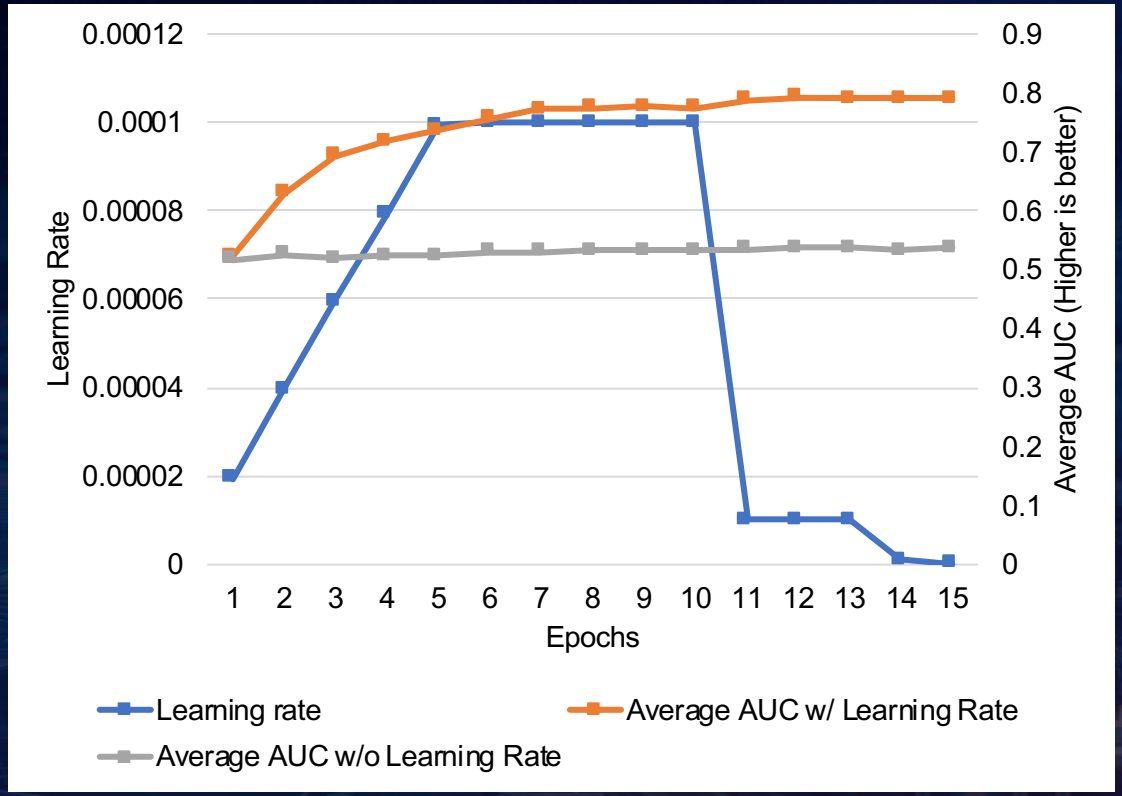
## Hyper-parameter

- Batch Size
  - Increase batch size increase parallelism
  - Impact accuracy



# Results and observations

## Impact on Learning Rate Scheduler

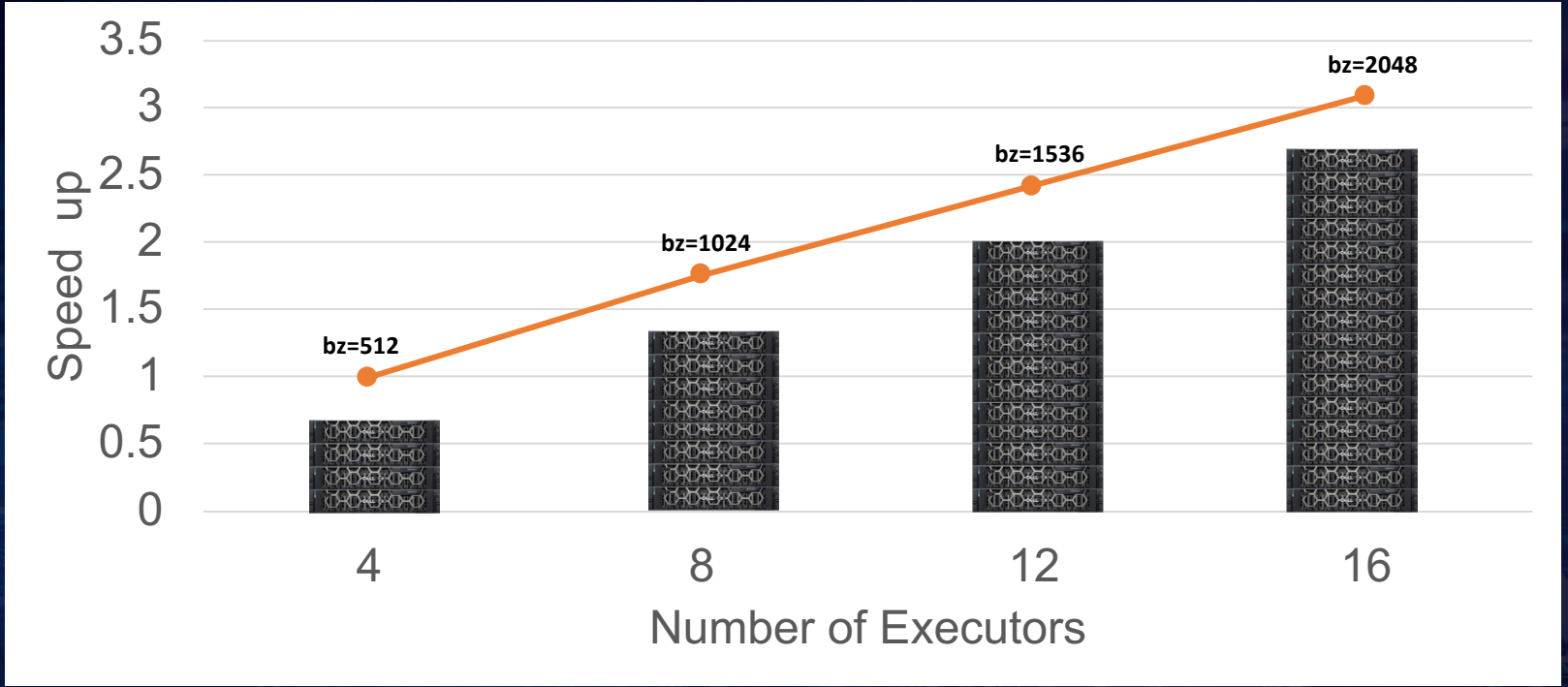



- **Optimizer:**
  - **Adam** (with Learning Rate Scheduler) outperforms SGD
- LR : Learning rate scheduler helps is covering the model much faster
  - Warmup: Gradually increase the learning rate for 5 epochs
  - Cool down: Hold the learning rate for a few epochs before cooling down till desired accuracy is achieved

# Results and observations

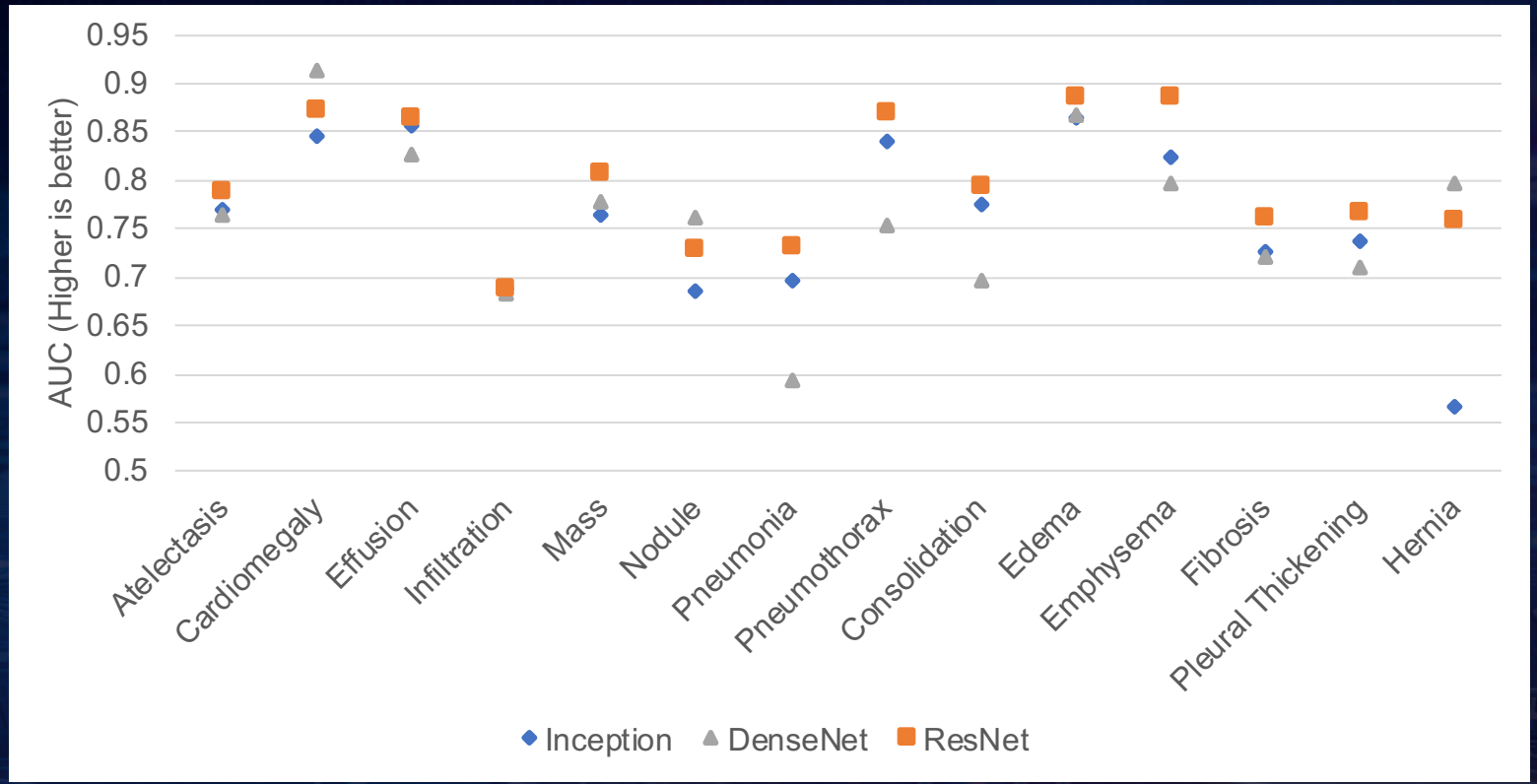
**Scalability**

3x speed up from 4 nodes to 16 nodes.
~ 2.5 hours to train the model



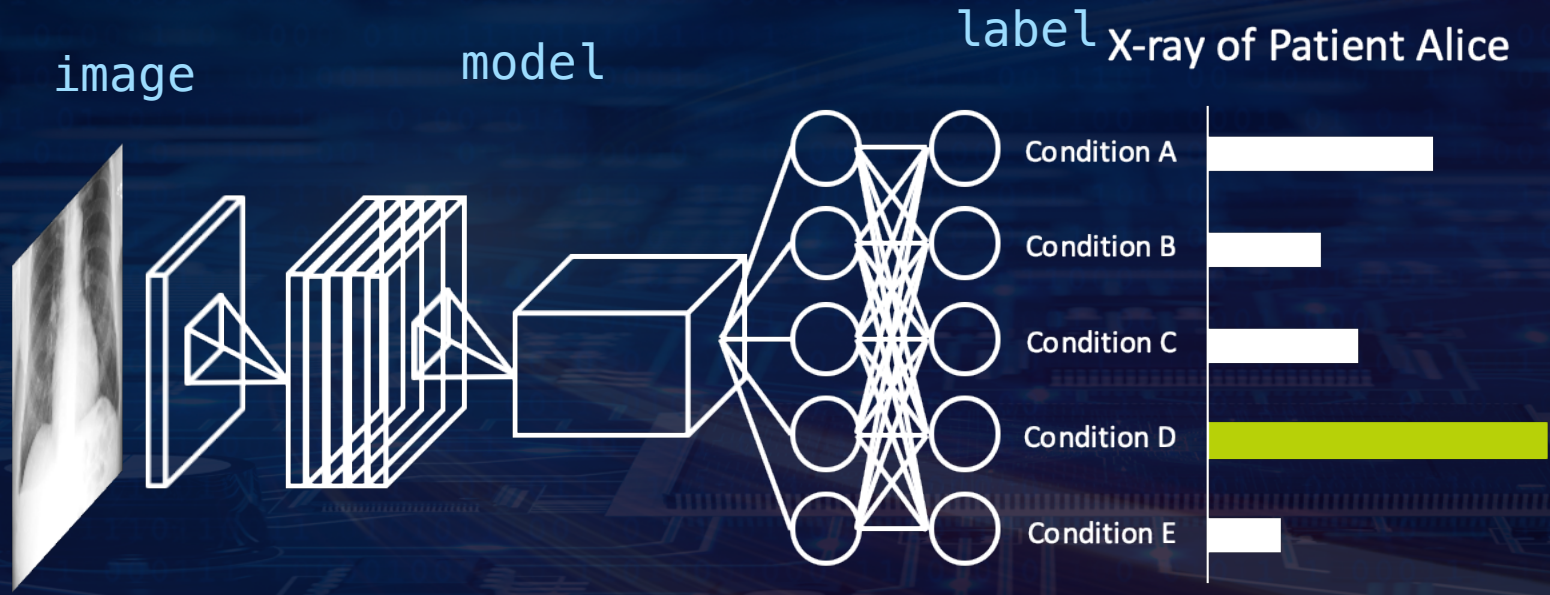
# Results and observations

## Base model comparison

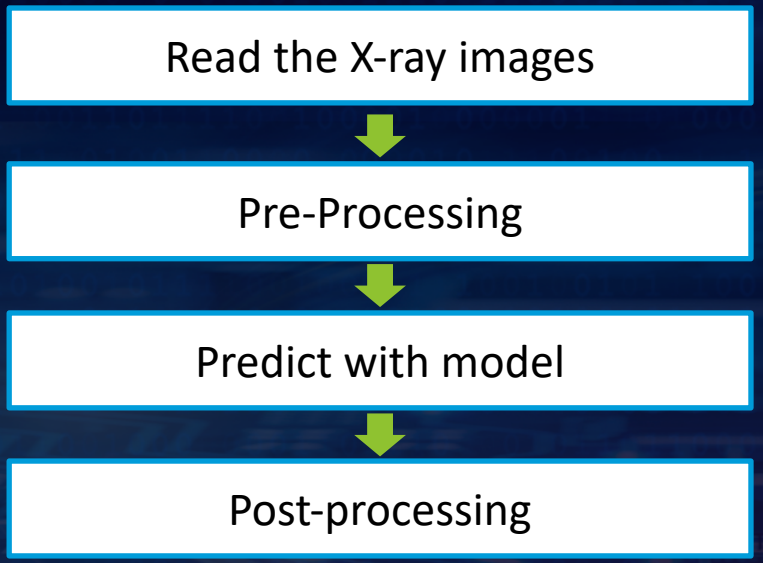


# Inference: Predict with Neural networks

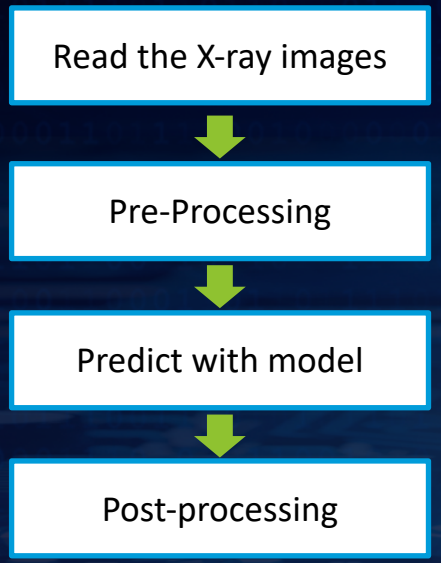
```
def predict(model, images):
    # Using model to predict
    return label
```



# Model: Predict with X-ray Model



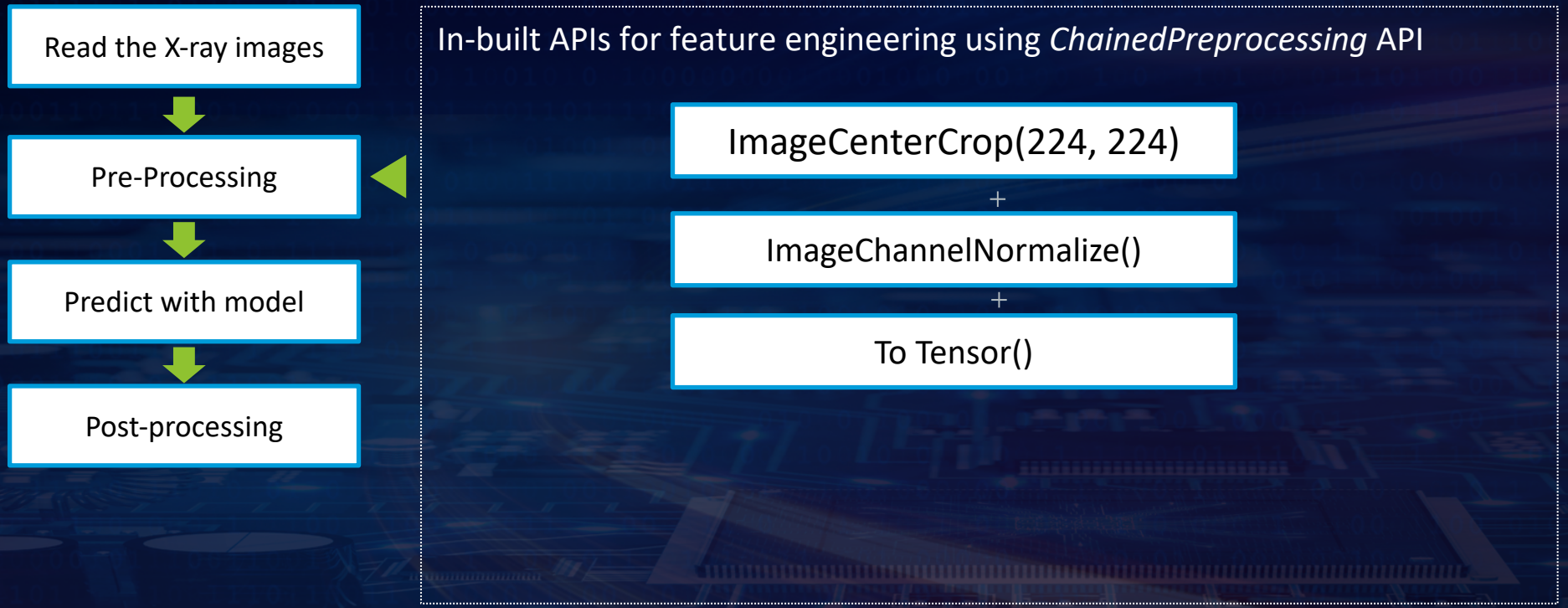
# Read the X-ray images as Spark DataFrames



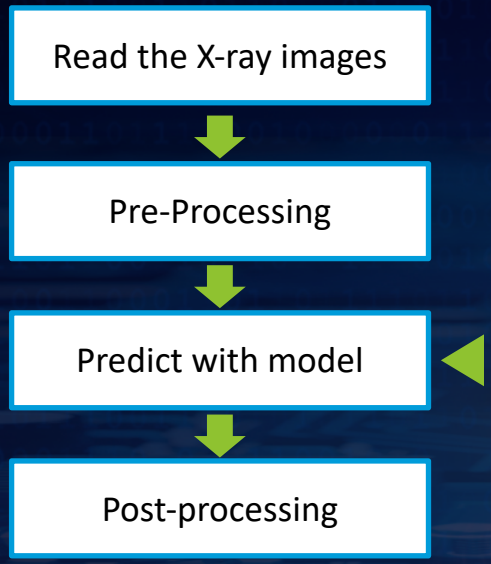

- Initialize *NNContext* and load Xray images into *DataFrames* using *NNImageReader*

```
from zoo.pipeline.nnframes import NNImageReader
imageDF = NNImageReader.readImages(image_path, sc,
                  resizeH=256, resizeW=256, image_codec=1)
```

# Read the X-ray images as Spark DataFrames



# Read the X-ray images as Spark DataFrames



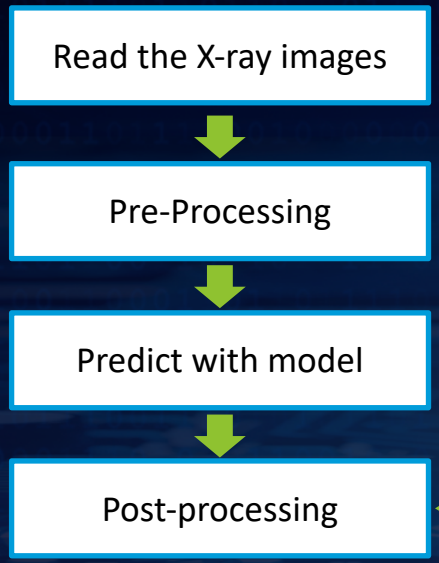

- Load trained model

```
trained_model = load_bigdl(model_path,bin_path)
```

- Predict with model

```
predictionDF = predict(trained_model, image_path)
```

# Read the X-ray images as Spark DataFrames

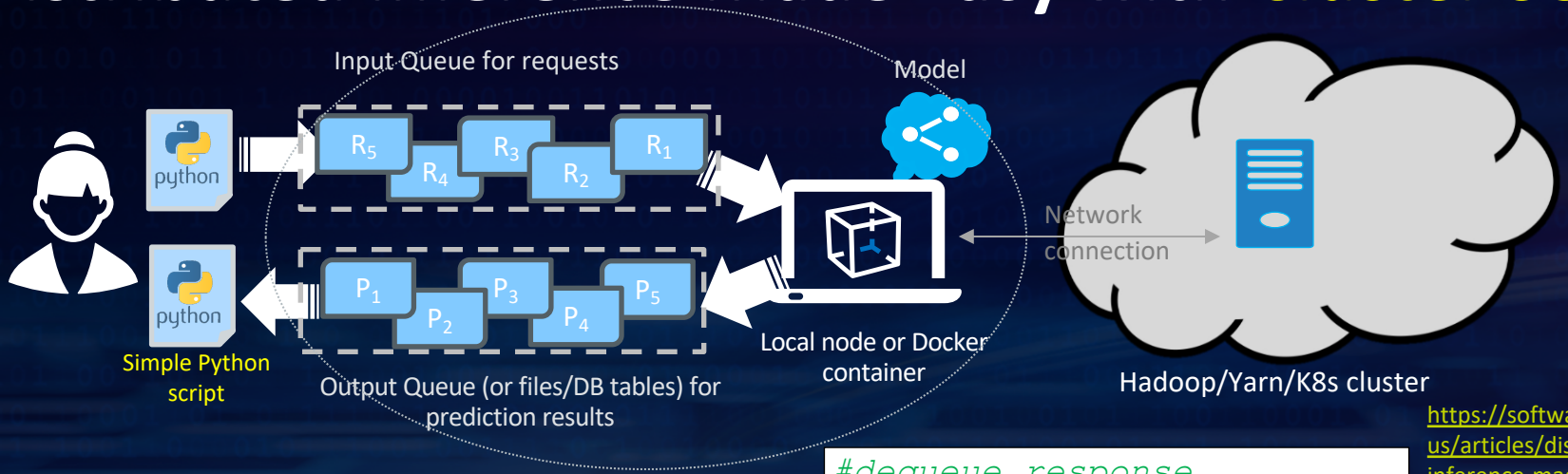



- Post-processing result (print, Top-1 or Top-5)

```
print_prediction_output(predictionDF)
```

```
Finding_Labels      - Prediction          - Label
--------------      - ----------          - -----
Atelectasis         - 0.0513895861804     - 0.0
Cardiomegaly        - 0.999990344048      - 1.0
Effusion            - 0.772566437721      - 1.0
Infiltration        - 0.104134827852      - 0.0
Mass                - 0.00671234633774    - 0.0
Nodule              - 0.00529360491782    - 0.0
Pneumonia           - 0.00492261582986    - 0.0
Pneumothorax        - 0.00217330595478    - 0.0
Consolidation       - 0.0124941058457     - 0.0
Edema               - 0.00712430523708    - 0.0
Emphysema           - 0.00228400458582    - 0.0
Fibrosis            - 0.0059804264456     - 0.0
Pleural_Thickening  - 0.00832628458738    - 0.0
Hernia              - 0.00239278026856    - 0.0
```

# Distributed Inference Made Easy with Cluster Serving




https://software.intel.com/en-us/articles/distributed-inference-made-easy-with-analytics-zoo-cluster-serving

```
#enqueue request
input = InputQueue()
img = cv2.imread(path)
img = cv2.resize(img, (224, 224))
input.enqueue_image(id, img)
```

```
#dequeue response
output = OutputQueue()
result = output.dequeue()
for k in result.keys():
  print(k + ": " + \
     json.loads(result[k]))
```

- √ **Users freed from complex distributed inference solutions**
- √ **Distributed, real-time inference automatically managed Analytics Zoo**
  - *TensorFlow, PyTorch, Caffe, BigDL, OpenVINO, …*
  - *Spark Streaming, Flink, …*

Analytics Zoo API in blue

# Others

- Datasets
  - Image classification with more classes
  - Object Detection & Image segmentation (Annotated Dataset with bounding box or segmentation)

- Models
  - High resolution input (between 224*224~1024*1024)
  - Other pre-trained models
  - Fine-tuning with more layers
- Other frameworks in Anlytics-Zoo (PyTorch and TensorFlow)
- Hyper-parameter tuning & AutoML (New feature in Analytics-Zoo)

# Reference


- AI-assisted Radiology Using Distributed Deep Learning on Apache Spark and Analytics Zoo
- https://github.com/dell-ai-engineering/BigDL-ImageProcessing-Examples
- Using Deep Learning on Apache Spark to diagnose thoracic pathology from chest X-rays

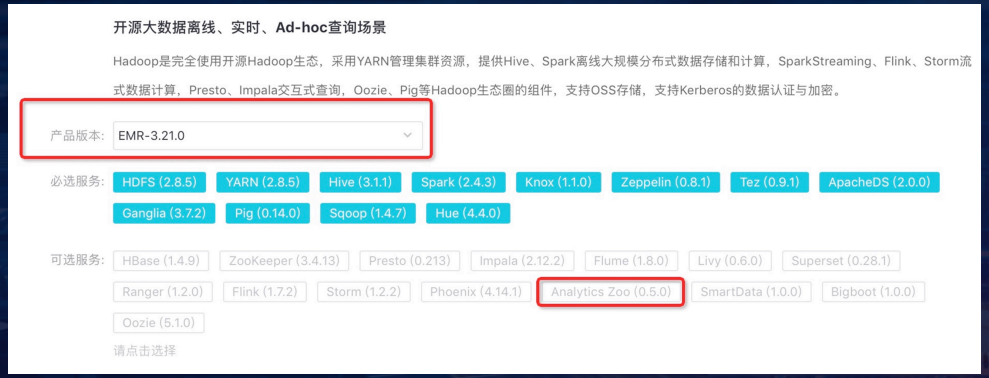
- ChestX-ray8: Hospital-scale Chest X-ray Database and Benchmarks on Weakly-Supervised Classification and Localization of Common Thorax Diseases
- CheXNet: Radiologist-Level Pneumonia Detection on Chest X-Rays with Deep Learning
- Comparison of Deep Learning Approaches for Multi-Label Chest X-Ray Classification

- COVID-Net: A Tailored Deep Convolutional Neural Network Design for Detection of COVID-19 Cases from Chest X-Ray Images
- COVID-CT-Dataset: A CT Scan Dataset about COVID-19

# Analytics Zoo on Ali E-MR

ANALYTICS ZOO + Alibaba Cloud aliyun.com

Analytics Zoo is already out-of-box on Ali EMR：




* Version upgrade for Analytics Zoo is on-going.

For more information and support, contact Wesley：

Email: wesley.du@intel.com
DingTalk:

# Apache Spark

# What is Analytics Zoo

BigDL

Distributed, High-Performance

Deep Learning Framework

for Apache Spark

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

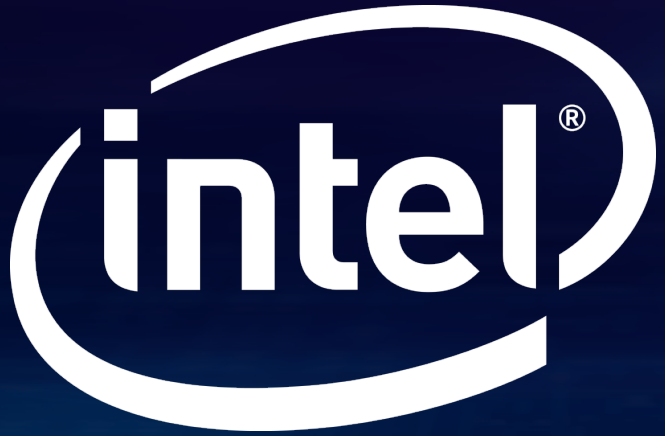
ANALYTICS ZOO

Unified Analytics + AI Platform

Distributed TensorFlow, Keras, PyTorch and BigDL on Apache Spark

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

**Accelerating Data Analytics + AI Solutions At Scale**


intel®

# Legal Disclaimers

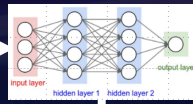
- Intel technologies' features and benefits depend on system configuration and may require enabled hardware, software or service activation. Learn more at intel.com, or from the OEM or retailer.
- No computer system can be absolutely secure.
- Tests document performance of components on a particular 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 **http://www.intel.com/performance**.

# Backup

# Deep Learning

## Training

Training data → Preprocessing → Training → Postprocessing → model




## Inference

Data → Preprocessing → Predict → Postprocessing → Answer

# Motivation

## 新型冠状肺炎带来的新需求和趋势

| 检测方法 | 优点 | 缺点 |
| --- | --- | --- |
| 试剂盒（主要） | 准确，可检测**无症状** | 有损耗，有接触，检测试剂长 |
| 胸透x光（辅助） | 不精确，无法检测**无症状** | 无损耗，无接触，检测时间短 |

### COVID-19 X-Ray Datasets

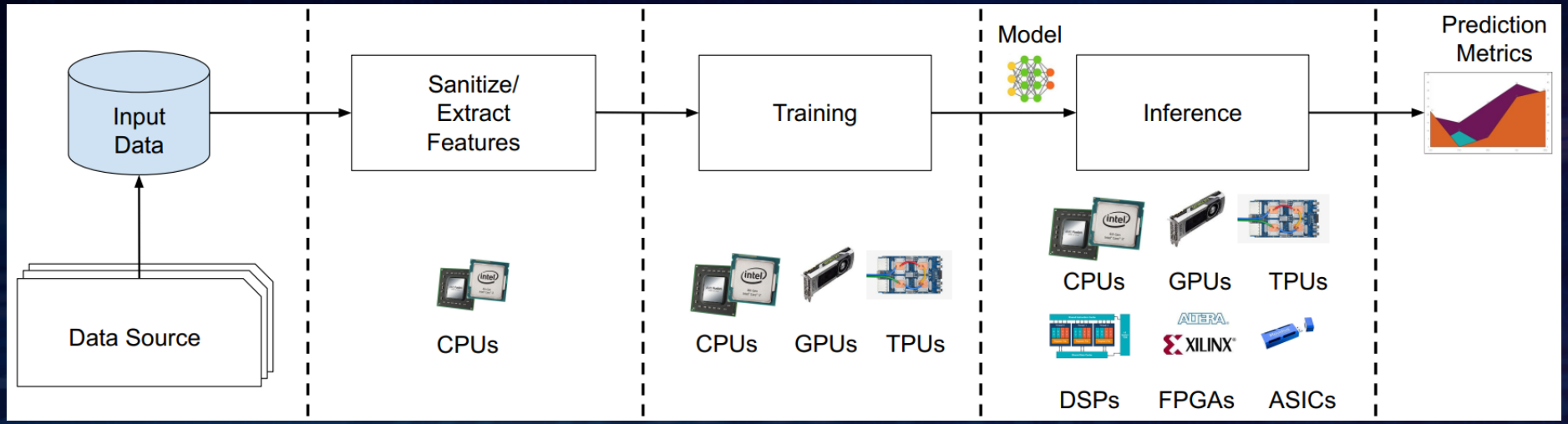
https://github.com/ieee8023/covid-chestxray-dataset
COVID-CT-Dataset: A CT Scan Dataset about COVID-19

### COVID-19 X-Ray based research

https://www.technologyreview.com/2020/03/24/950356/coronavirus-neural-network-can-help-spot-covid-19-in-chest-x-ray-pneumonia/
COVID-Net: A Tailored Deep Convolutional Neural Network Design for Detection of COVID-19 Cases from Chest X-Ray Images

# Predicting diseases in Chest X-rays

# Deep Learning Pipeline



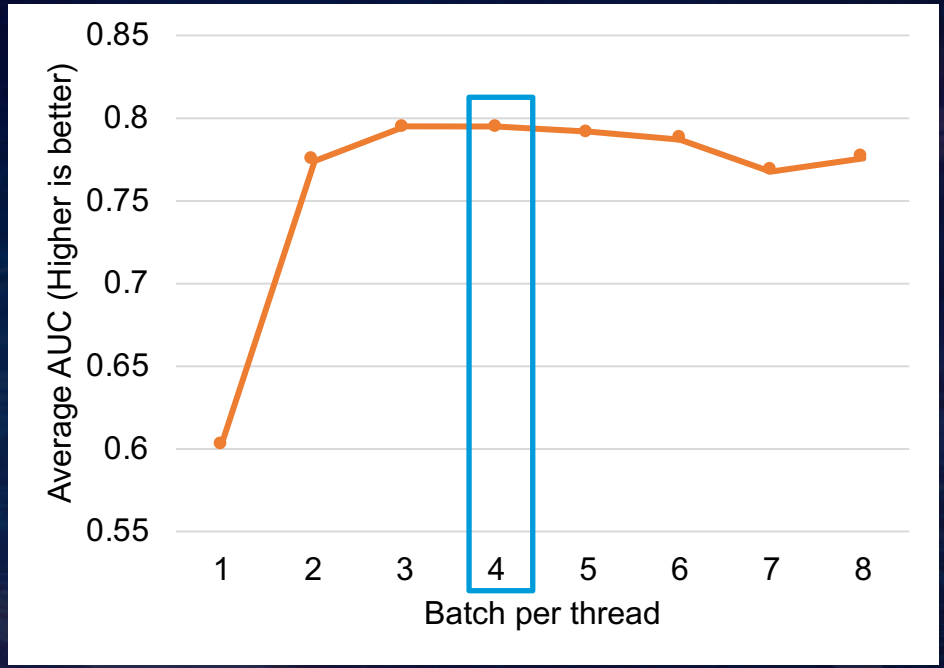

https://mlperf.org/

VJ Reddi, 2019, MLPerf Inference Benchmark

# Results and observations

## Batch size and Scalability

- Two factors:
  - Batches per thread (mini-batch): Number of images in an iteration per worker
  - Global batch size: Batches per thread * Number of workers
- Batch size is a critical parameter for model accuracy as well as for distributed performance
  - Increasing the batch size increases parallelism
  - Increasing the batch size may lead to a loss in generalization performance especially



# Results and observations

## Average AUC-ROC

| Disease | ResNet-50 | DenseNet | Inception |
|---|---|---|---|
| Atelectasis | **0.789** | 0.764 | 0.770 |
| Consolidation | 0.873 | **0.915** | 0.844 |
| Infiltration | **0.865** | 0.826 | 0.855 |
| Pneumothorax | **0.689** | 0.683 | 0.688 |
| Edema | **0.807** | 0.778 | 0.764 |
| Emphysema | 0.729 | **0.762** | 0.684 |
| Fibrosis | **0.731** | 0.592 | 0.696 |
| Effusion | **0.870** | 0.753 | 0.841 |
| Pneumonia | **0.794** | 0.696 | 0.775 |
| Pleural Thickening | **0.886** | 0.867 | 0.865 |
| Cardiomegaly | **0.887** | 0.796 | 0.825 |
| Nodule | **0.762** | 0.722 | 0.725 |
| Mass | **0.768** | 0.709 | 0.738 |
| Hernia | 0.759 | **0.796** | 0.565 |
| Average AUC | **0.801** | 0.761 | 0.760 |