

ApacheCon Big Data Europe  
Nov 2016, Seville, Spain

# SMART MANUFACTURING

With Apache Spark Streaming and Deep Learning

Prajod Vettiyattil, Wipro

@prajods

<https://in.linkedin.com/in/prajod>

# AGENDA

- Introduction
- Smart Manufacturing
- Applications
- Deep learning solution
- CNN
- Deeplearning4j (DL4J)
- Apache Spark
- Deploying machine learning

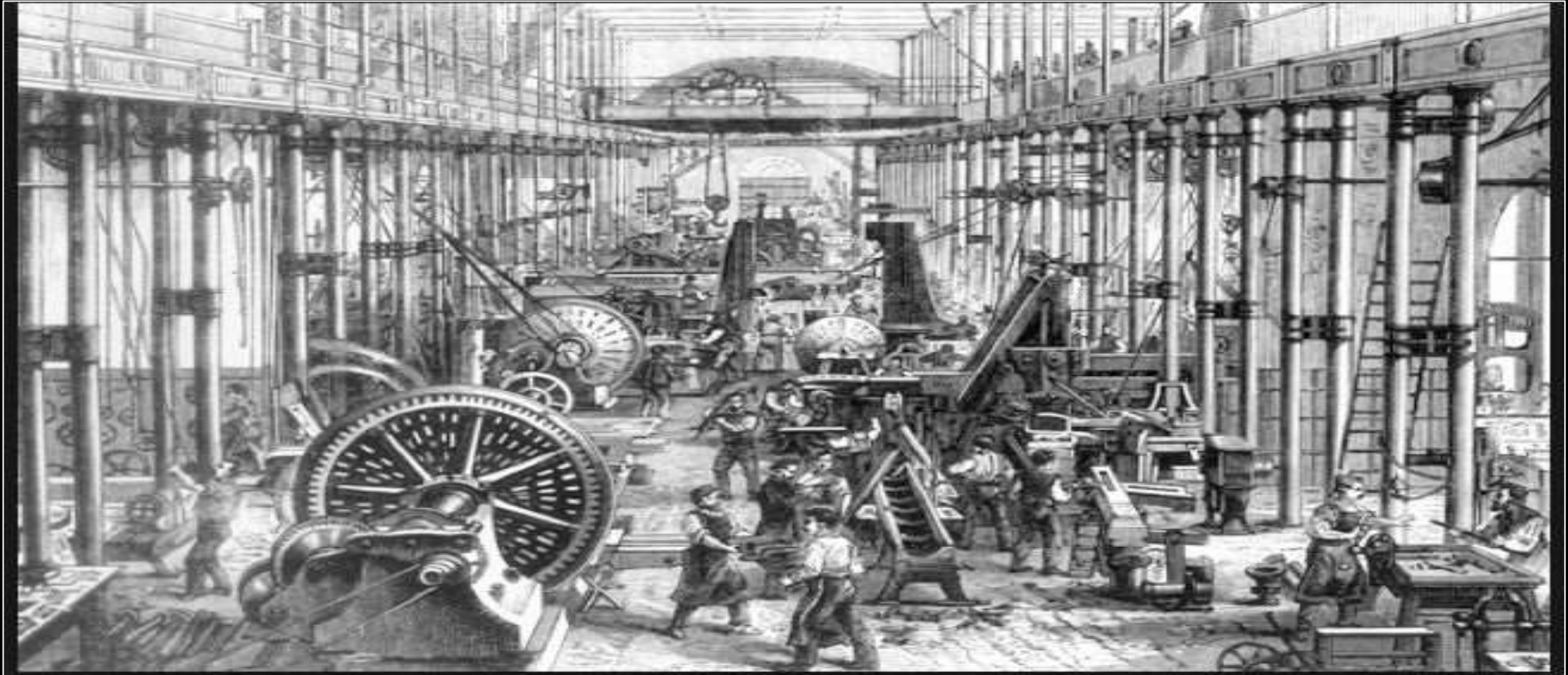
# INTRODUCTION

- Industrialization
- Automation
- Scale
- Commoditization
- Accessibility





# MEN AND MACHINES



# KEY POINT

Machines can do what humans can't

# SMART MANUFACTURING

- Holistic view of entire process
  - Sourcing to product sale
- Efficiency
- Eco friendliness
- Predict over react
- Autonomous machines
- Learning and AI
- Use data and analytics

# APPLICATIONS

- Detecting overheating
  - Equipment
  - Finished products
- Glass defect detection

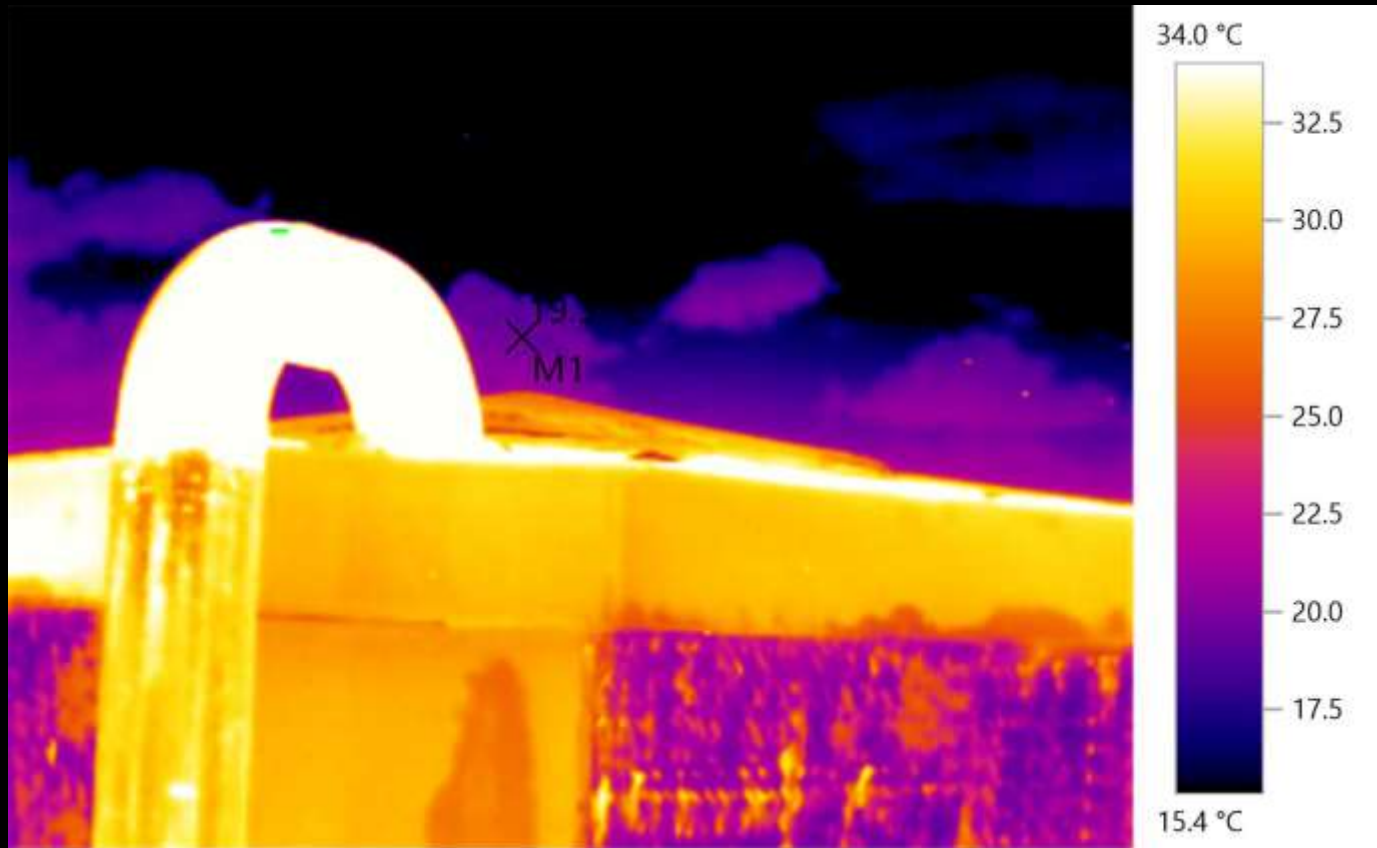
# OVERHEATING

- Detect temperatures above defined boundaries

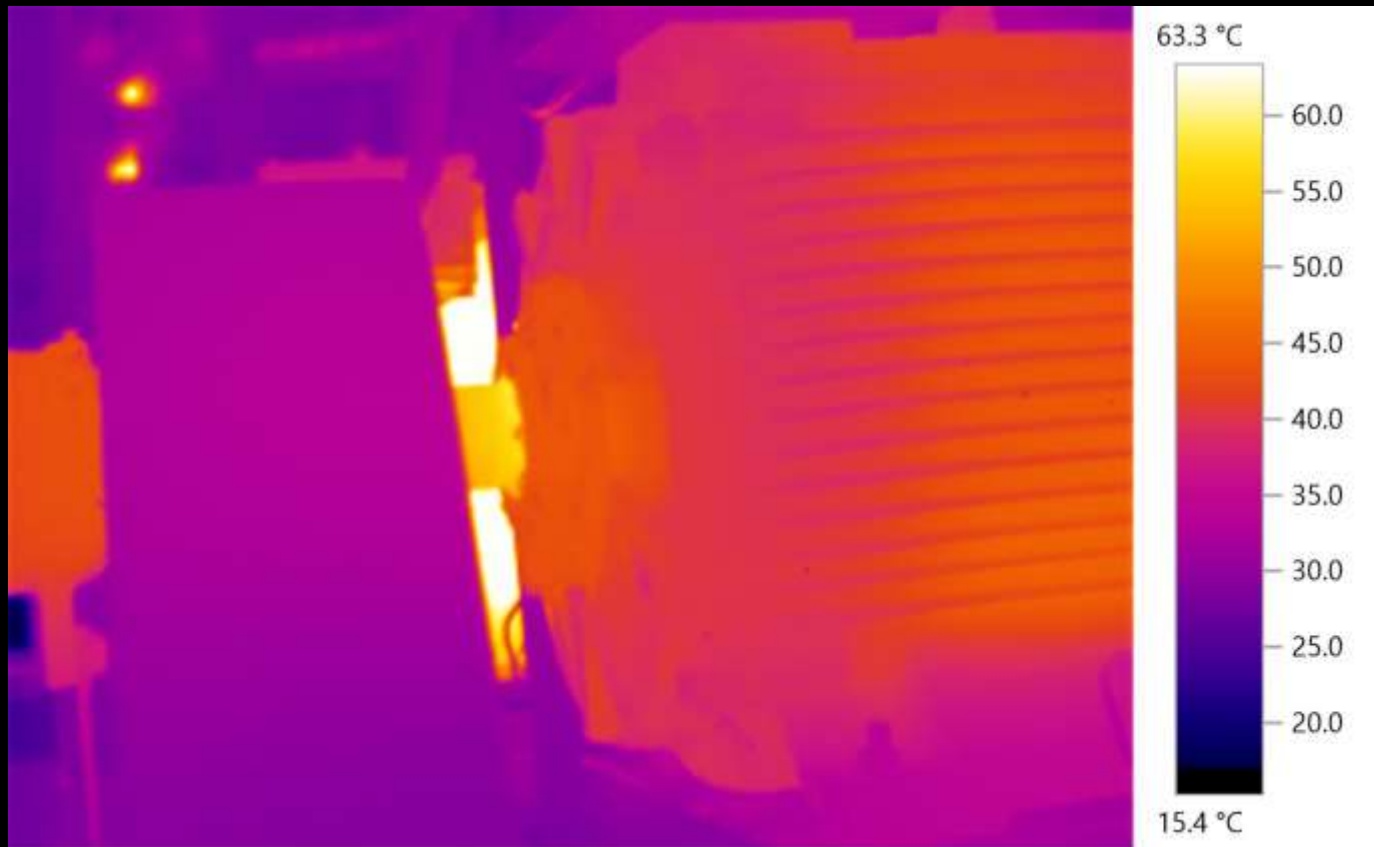




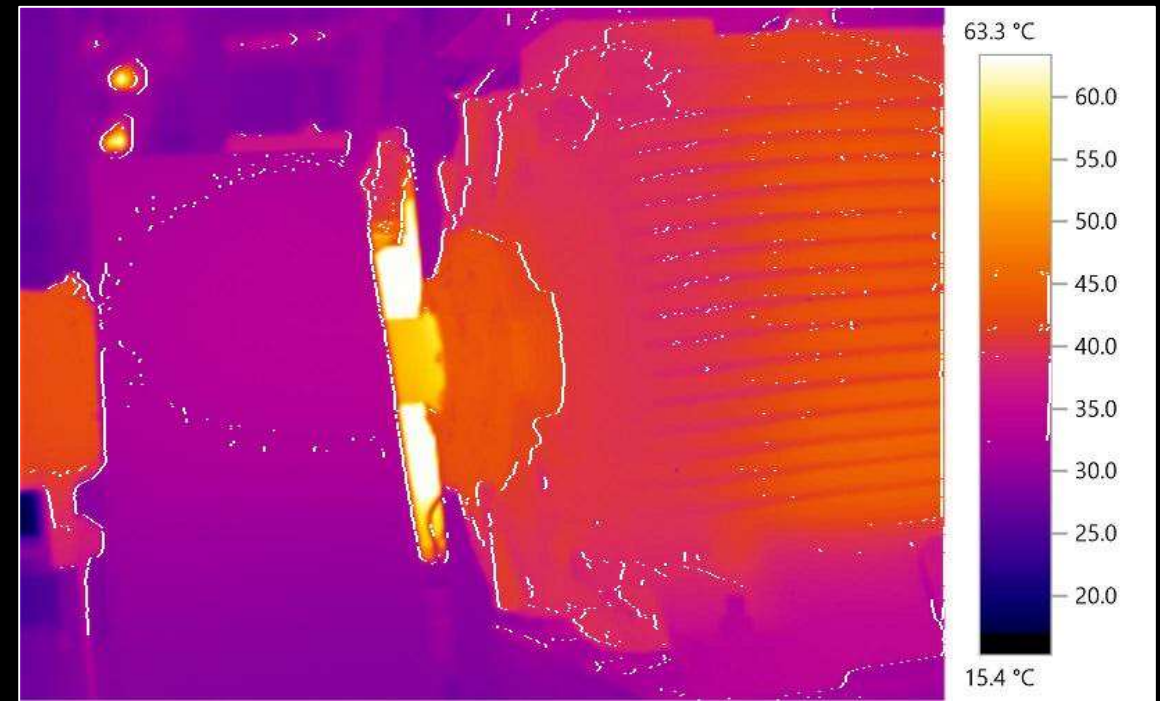
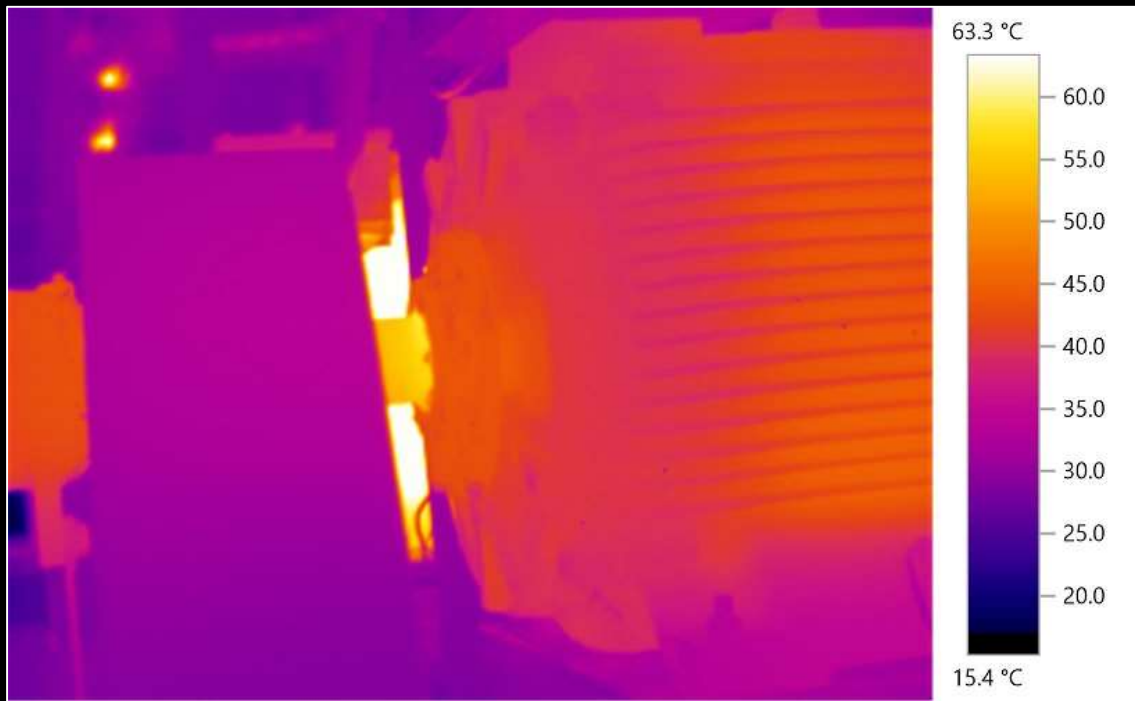
# COOLING TOWER



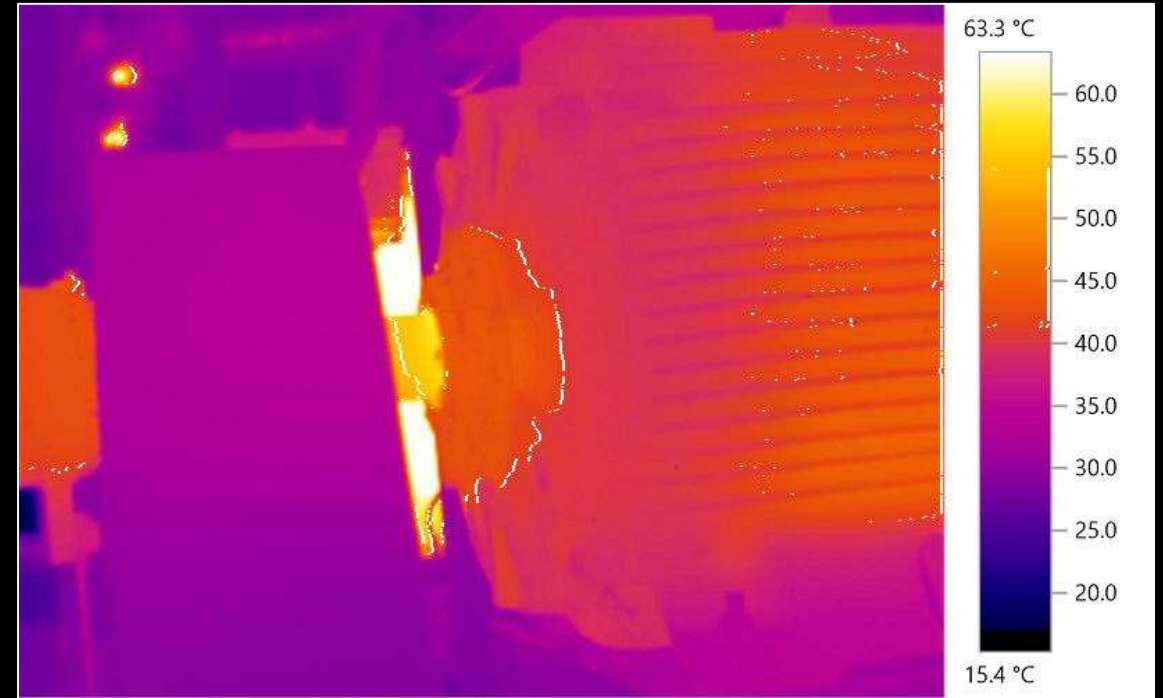
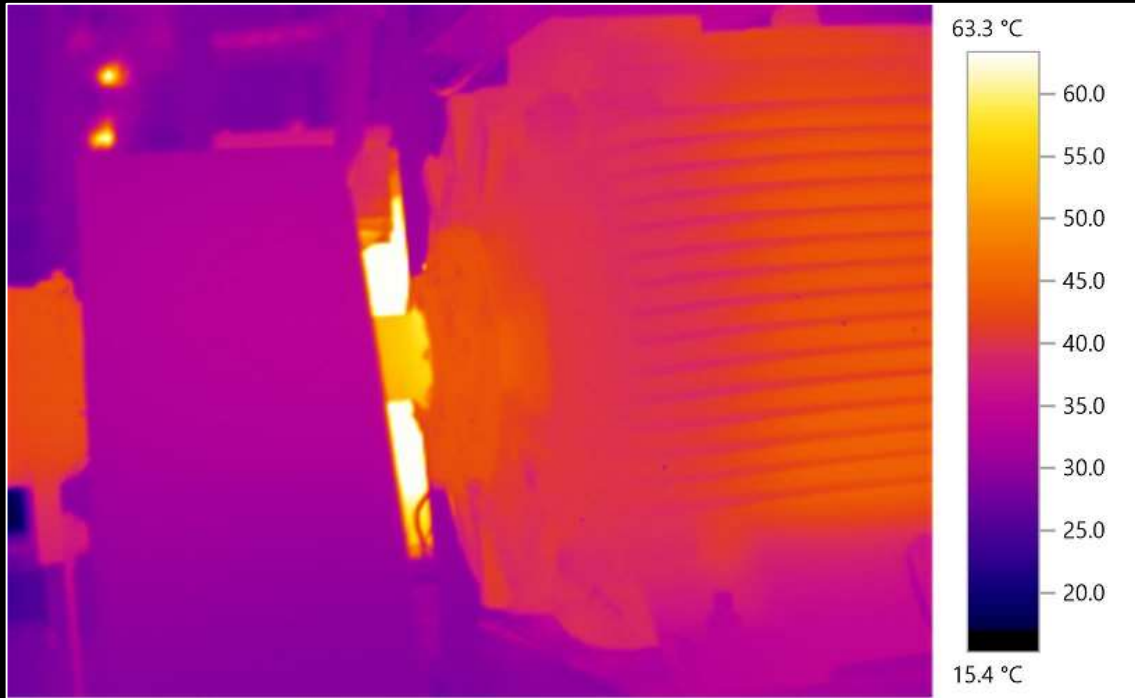
# ELECTRIC MOTOR



# ELECTRIC MOTOR: OVERHEATING



# OVERHEATING: LOWER RESOLUTION





# ELECTRIC MOTOR: NORMAL VIEW







# GLASS DEFECT

# ZOOMING IN



# UNDER A DIFFERENT LIGHT



# AUTOMATING DETECTION

- Machine learning
- Multivariate Analysis
  - Multiple categories of defects
- Deep Learning

# DEEP LEARNING: WHAT

- Neural Network based Machine Learning
- Neural Network
  - 1 Input layer
  - 1 or more hidden layers
  - 1 Output layer
- Basic unit of NN
  - Neuron or Perceptron
- Combine perceptrons in many ways using multiple parameters
- Model parameters
- Hyperparameters

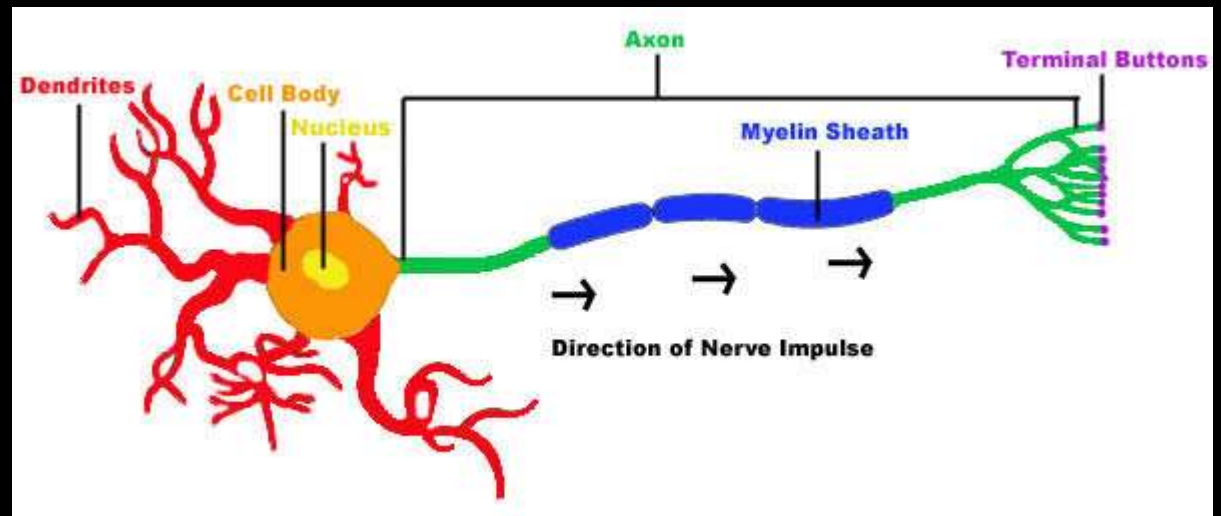


Image from: [ucalgary.ca](http://ucalgary.ca)



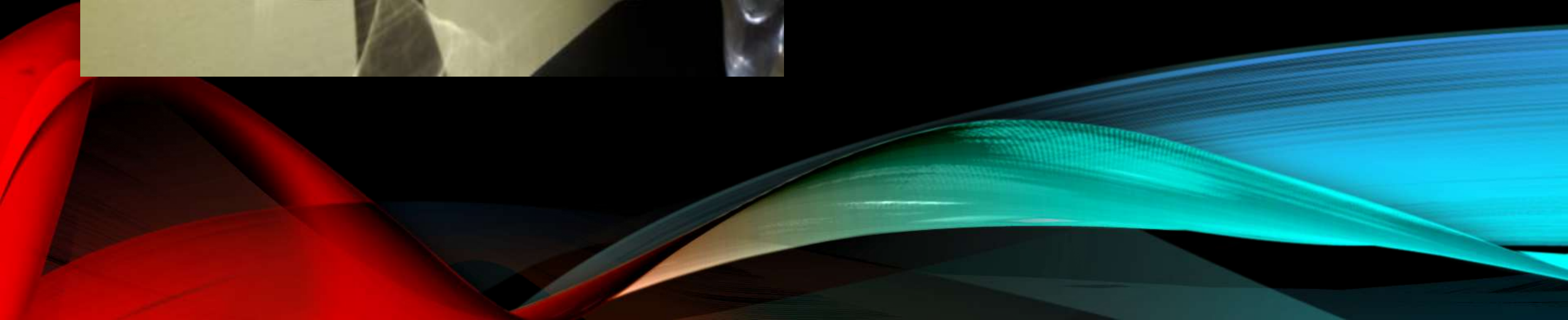
# DEEP LEARNING: WHY

- Can discern complex input patterns
- Higher accuracy than most other ML methods
  - for image analysis
- Ideally suited for spatial data analysis
- Higher time needed for training
- Parallel execution
- Higher level architectures for evolving needs
  - CNN, RNN, LSTM





# CONVOLUTIONAL NEURAL NETWORKS



# CONVOLUTIONAL NEURAL NETWORK

- Convolutional Neural Network (CNN)
- Convolve: mix two functions
  - Matrix multiplication and addition
  - Dot product
- Pass many filters over an image
- Image features
- Images are 3D features
- Discern the important features/signals
- Filter = kernel

# CNN: APPLICATION

- Self driving cars
- Robotics
- Drones
- Industrial automation
- Physical security
- Medical labs
- Wherever images or videos are used

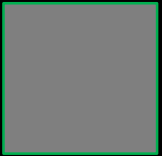
# IMAGE FOR CONVOLUTION



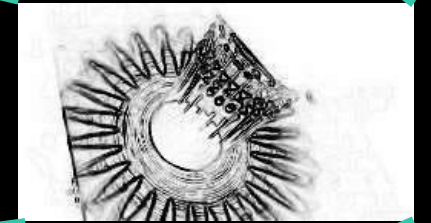
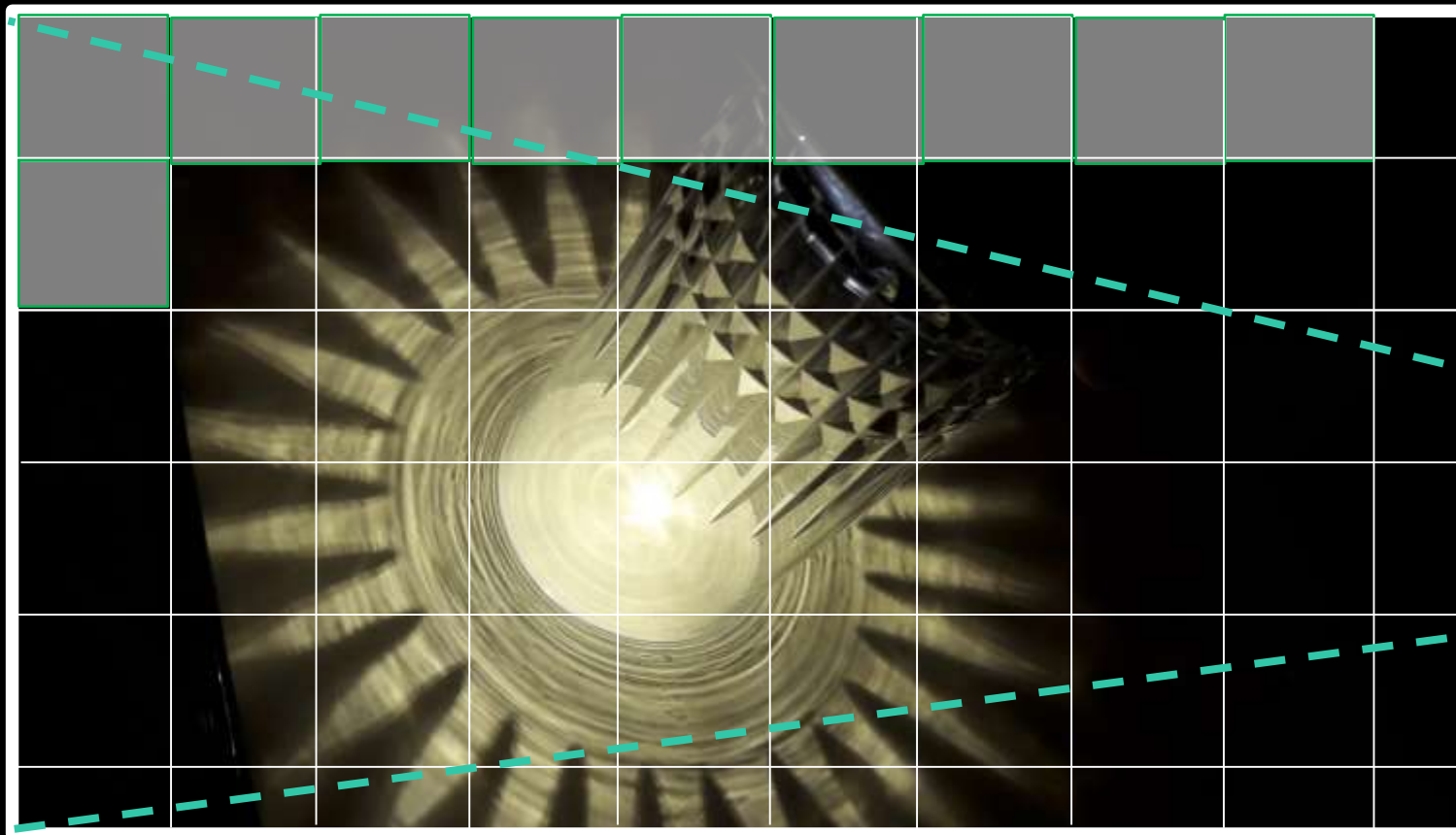


# CONVOLUTION

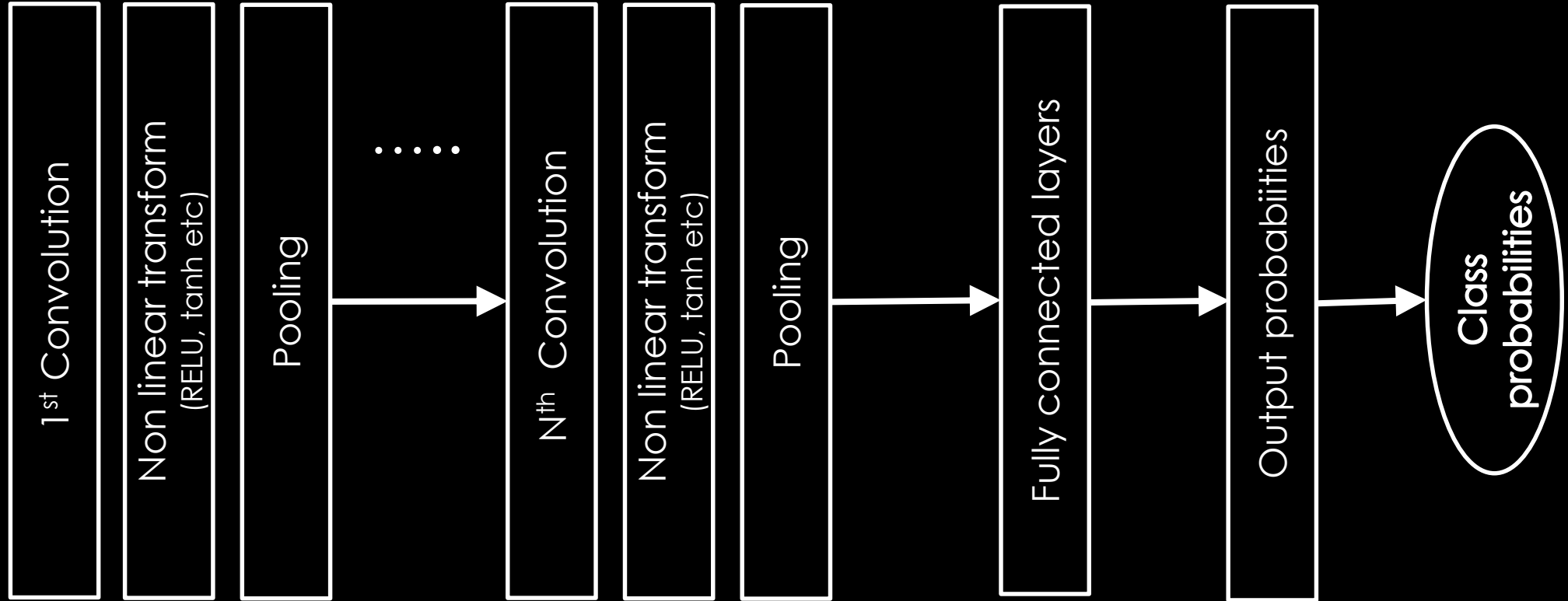
Filter matrix



Image



# CNN ARCHITECTURE



Feature extraction + dimension reduction

Classification

# CNN PARAMETERS

- Model parameters
  - Filter values
  - Weights
  - Biases

Learned by NN

- Hyperparameters
  - Activation function
  - # layers
  - # nodes in each layer
  - Filter size and count
  - Regularization
  - Convolution layers
  - Maxpooling layers

Experience of designer

# CNN: SAMPLE CODE

- ```
MultiLayerConfiguration conf = new NeuralNetConfiguration.Builder()  
    .seed(seed).iterations(iterations).regularization(false).l2(0.003).activation("relu")  
    .learningRate(0.0007).weightInit(WeightInit.XAVIER)  
    .optimizationAlgo(OptimizationAlgorithm.STOCHASTIC_GRADIENT_DESCENT)  
    .updater(Updater.RMSPROP).momentum(0.9).list()  
    .layer(0, convInit("cnn1", channels, 50 , new int[]{5, 5}, new int[]{1, 1}, new int[]{0, 0}, 0))  
    .layer(1, maxPool("maxpool1", new int[]{2,2}))  
    ....  
    .layer(4, new DenseLayer.Builder().nOut(500).build())  
    .layer(5, new OutputLayer.Builder(LossFunctions.LossFunction.NEGATIVELOGLIKELIHOOD)  
        .nOut(numLabels).activation("softmax").build())  
    .backprop(true).pretrain(false).cnnInputSize(height, width, channels).build();
```

# DEEPLARNING4J (DL4J)

- Deep learning library
- Open source
- Apache 2.0 license
- Java based
- Distributed execution
- Runs on Spark and Hadoop



# DEEPLARNING4J: FEATURES

- ND4J
  - N Dimensional Arrays/Tensors
  - Like numpy
- Canova/DataVec
  - Data extraction, vectorization
- Arbitrer
  - Evaluate and tune models
- CPU or GPU
  - OpenBLAS, Intel MKL, Nvidia CUDA

# DEEPLARNING4J: SUPPORT

- Community support
  - <https://gitter.im/deeplearning4j/deeplearning4j>
- Commercial support
  - [www.skymind.io](http://www.skymind.io)

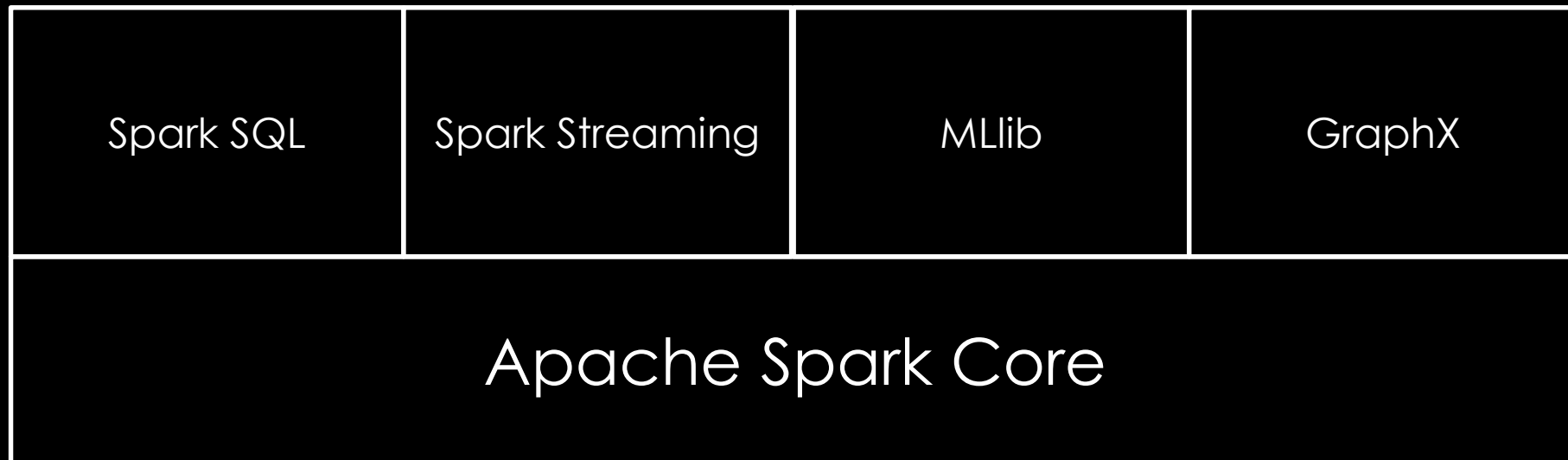
# APACHE SPARK

- Incubated in 2009 at Berkeley University
- 100s of contributors
- Yahoo, Intel, UC Berkeley, ..., 50+ orgs
- 10 to 100 times faster than Hadoop MR
  - <https://databricks.com/blog/2014/10/10/spark-petabyte-sort.html>

# APACHE SPARK

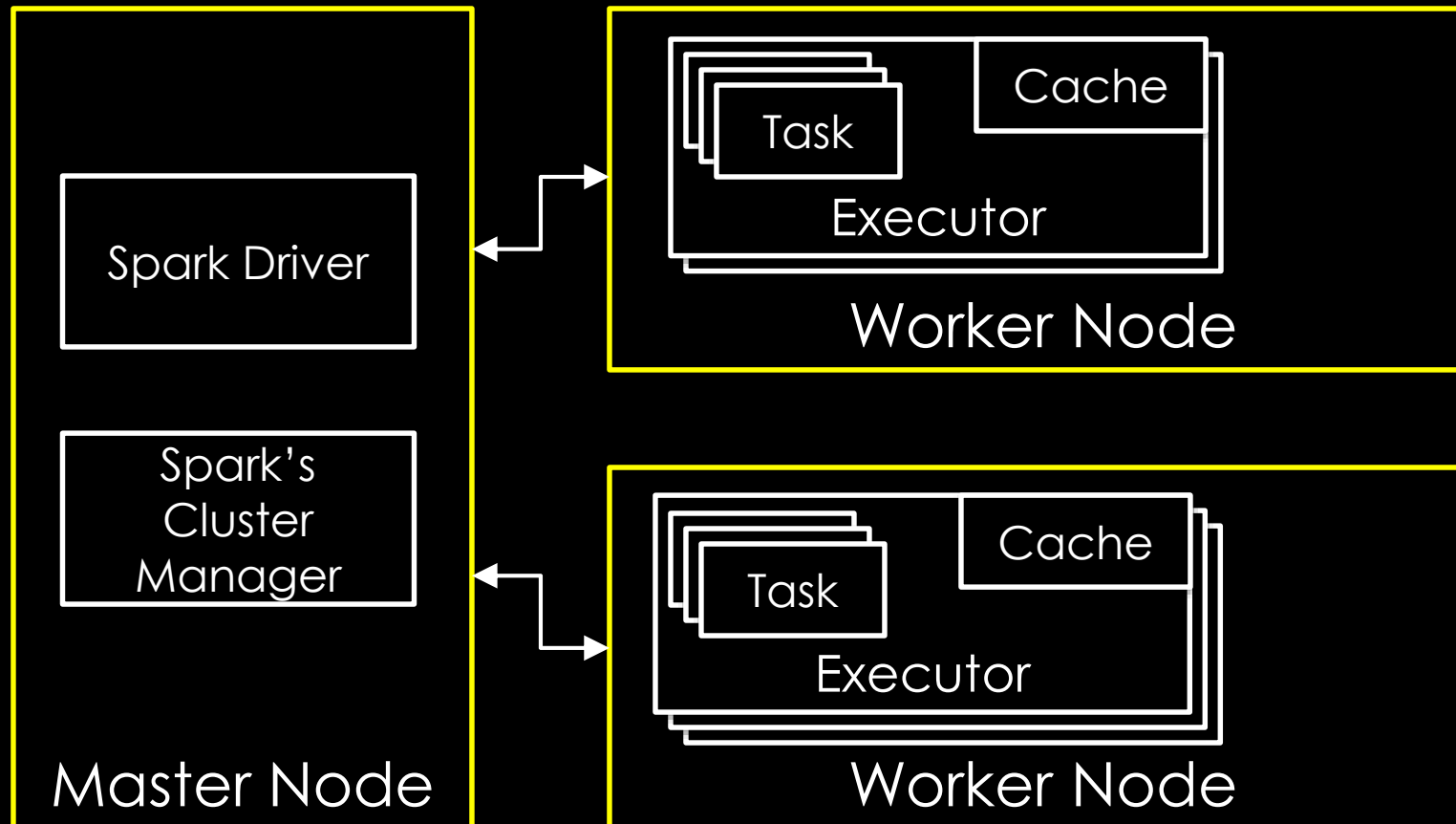
- Complements Hadoop
  - Replaces Hadoop MR
  - Adds
    - In memory processing
    - Stream processing
    - Interactive queries
- YARN or Mesos for clustering
- Java, Scala, Python, R

# SPARK: LOGICAL VIEW

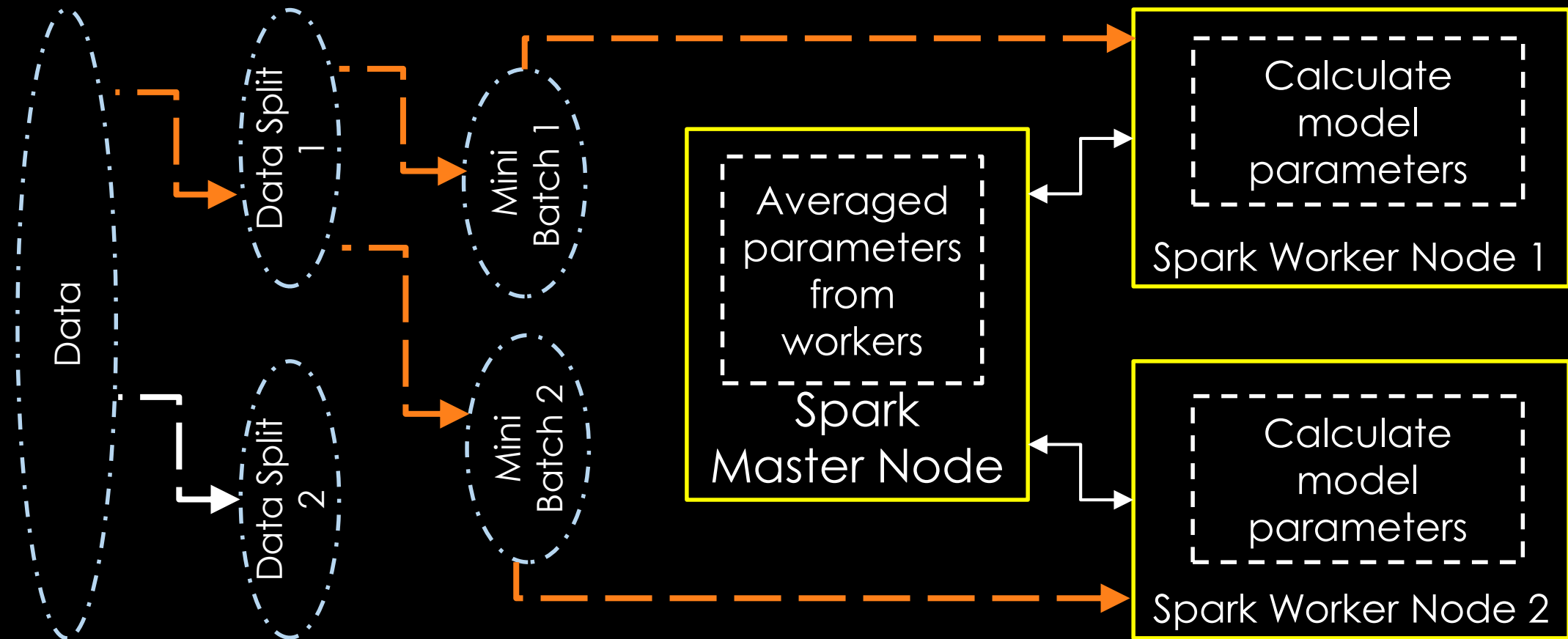




# SPARK: DEPLOYMENT VIEW



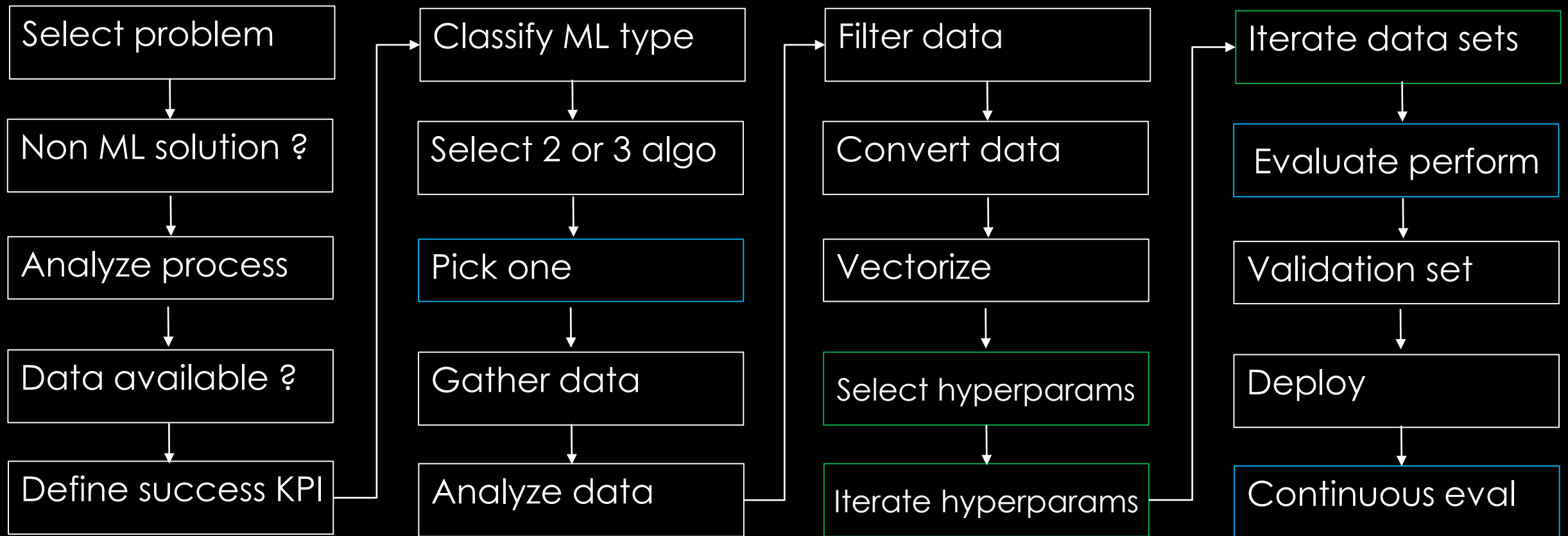
# PARALLEL TRAINING: SPARK + DL4J



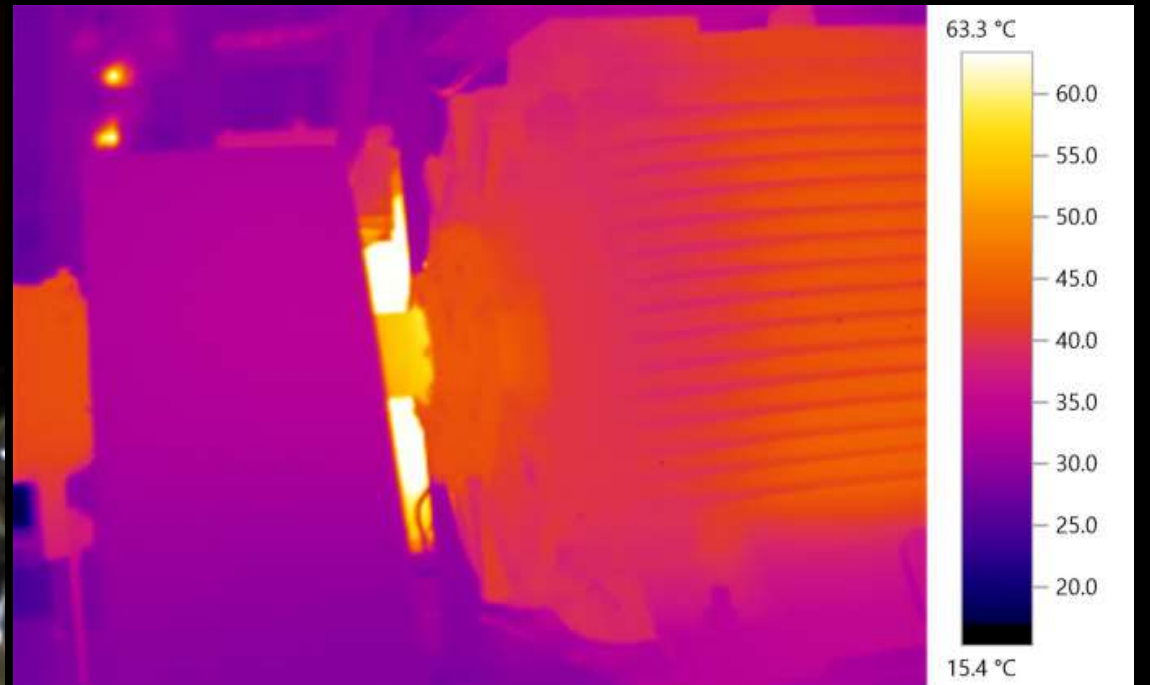
# THE MACHINE LEARNING PROCESS



# A LONG JOURNEY



# SELECT PROBLEM





# GATHER AND FORMAT DATA



# PROGRAM

# ITERATE WITH HYPERPARAMETERS AND DATA

# CONTINUOUS EVALUATION

# SUMMARY

- Automation and commoditization
- Humans and machines
- Beyond human capacity
- Application to product quality inspection
- Deep learning and images: higher accuracy
- CNN
- DL4J + Spark
- The Machine Learning process





# REFERENCES

- [https://en.wikipedia.org/wiki/Smart\\_manufacturing](https://en.wikipedia.org/wiki/Smart_manufacturing)
- <http://karpathy.github.io/neuralnets/>
- <http://cs231n.github.io/convolutional-networks/>
- <https://deeplearning4j.org/convolutionalnets.html>
- <https://ujjwalkarn.me/2016/08/11/intuitive-explanation-convnets/>
- [https://en.wikipedia.org/wiki/Glass\\_production](https://en.wikipedia.org/wiki/Glass_production)
- <http://www.qorpak.com/pages/lab-glass-defects-inspect-packaging>
- [https://en.wikipedia.org/wiki/Automated\\_analyser](https://en.wikipedia.org/wiki/Automated_analyser)

# QUESTIONS

Prajod Vettiyattil, Wipro  
@prajods  
<https://in.linkedin.com/in/prajod>

