



# IBM PowerAI Vision with Xilinx Alveo U200 Accelerator Card

Presented By

**IBM PowerAI**

Amartey Pearson

Sr. Technical Staff Member - IBM PowerAI Development

[apearson@us.ibm.com](mailto:apearson@us.ibm.com)

Oct 2<sup>nd</sup>, 2018



# Power Systems?

Top 500 ranking [\[edit\]](#)

Top 10 positions of the 51st TOP500 in June 2018<sup>[17]</sup>

Rank ↕	Rmax Rpeak (PFLOPS) ↕	Name ↕	Model ↕	Processor ↕	Interconnect ↕	Vendor ↕	Site country, year ↕	Operating system ↕
1 ▲	122.300 187.659	Summit	Power System AC922	POWER9, Tesla V100	Infiniband EDR	IBM	Oak Ridge National Laboratory 🇺🇸 United States, 2018	Linux (RHEL)
2 ▼	93.015 125.436	Sunway TaihuLight	Sunway MPP	SW26010	Sunway <sup>[18]</sup>	NRCPC	National Supercomputing Center in Wuxi 🇨🇳 China, 2016 <sup>[18]</sup>	Linux (Raise)
3 ▲	71.610 119.194	Sierra	Power System S922LC	POWER9, Tesla V100	Infiniband EDR	IBM	Lawrence Livermore National Laboratory 🇺🇸 United States, 2018	Linux (RHEL)



## Summit by the numbers

200: Petaflops (quadrillion calculations per second)

9,216: Number of IBM POWER9 CPUs.

4,608: Number of nodes.



*“With the IBM Power Systems AC922 server, **IBM has already demonstrated that we have the best platform for enterprise AI training.***

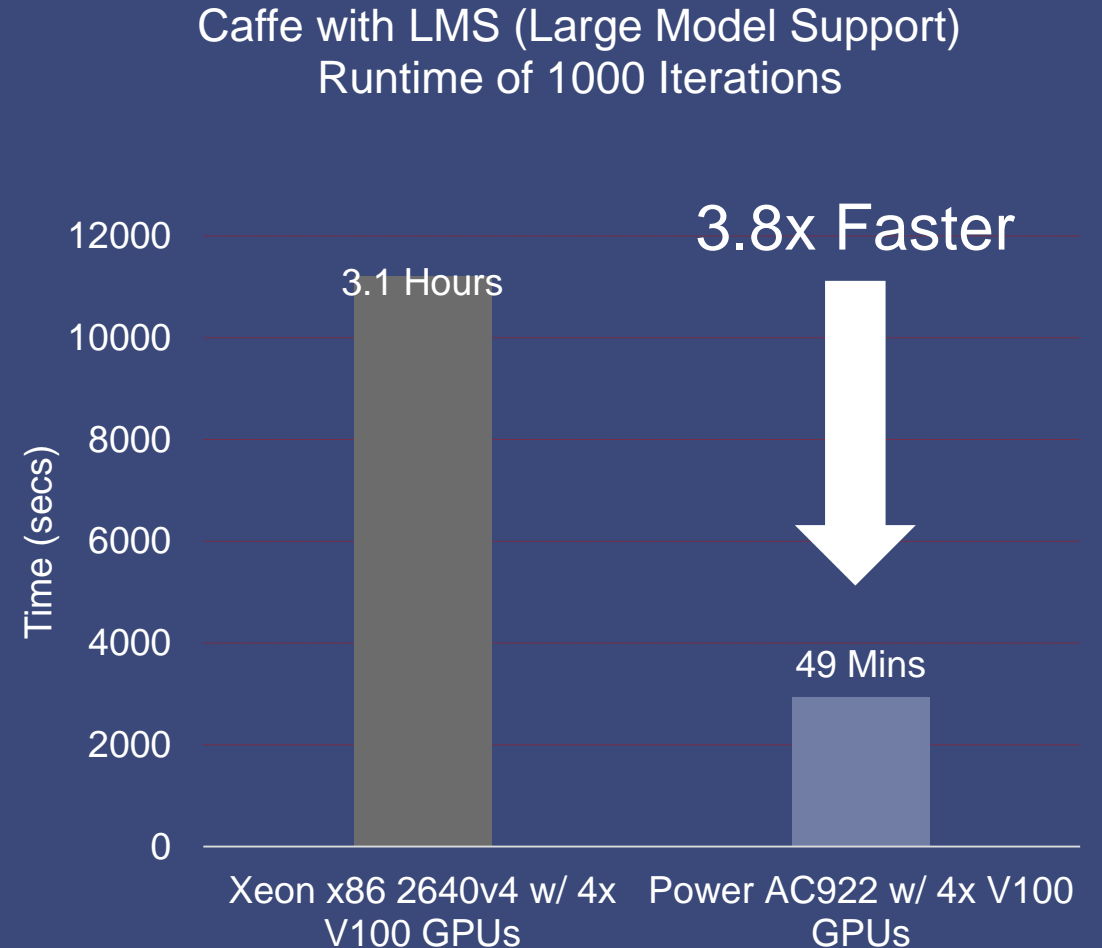
*IBM sees inference as a key component of a complete, end-to-end AI platform, and POWER9's leadership I/O bandwidth for data movement makes it an **ideal pairing with Xilinx's new Alveo U200 accelerator card to bring inference to the enterprise.**”*

- Steve Sibley, Vice President of IBM Cognitive Systems.

# Large AI Models Train ~4 Times Faster

**POWER9 Servers with NVLink to  
GPUs  
vs  
x86 Servers with PCIe to GPUs**

Detailed Benchmark Information in Back



GoogLeNet model on Enlarged  
ImageNet Dataset (2240x2240)

# PowerAI *Vision*

## Auto-ML for Images & Video

Label

Train

Deploy

# PowerAI

## PowerAI: Open Source ML Frameworks



Large Model Support (LMS)

# PowerAI *Enterprise*

Distributed Deep Learning  
(DDL)

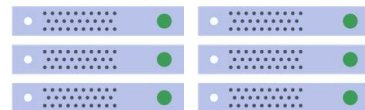
Auto ML

**IBM Spectrum Conductor with Spark**  
Cluster Virtualization,  
Auto Hyper-Parameter Optimization

## Deep Learning Impact (DLI) Module

Data & Model  
Management, ETL,  
Visualize, Advise

# Accelerated Infrastructure



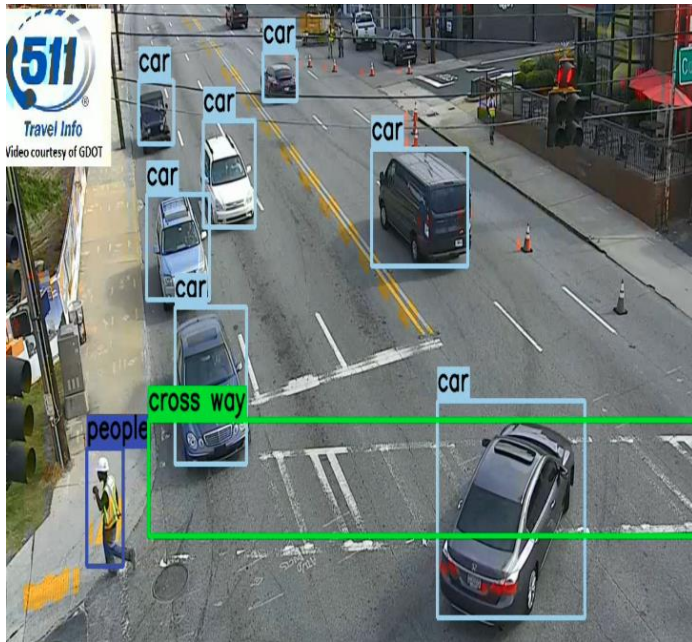
Accelerated Servers



Storage

# PowerAI Vision: "Point-and-Click" AI for Images & Video

## Label Image or Video Data



## Auto-Train AI Model

IBM PowerAI Vision Data Sets Models Deployed Models Administrator Log out

Data set MyDogs

Objects

Trained model / fasterrcnn-model

Trained models are created from prepared data sets. The model can be validated, exported, and deployed for production for Graphics Processing Units (GPU).

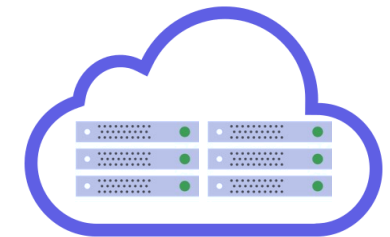
Deploy model Export model

LEARNING RATE	MAX ITERATION	WEIGHT DECAY
0.001	4000	0.0005
MOMENTUM	RATIO	ACCURACY
0.9	0.8	98%

Loss VS Iteration

Legend: Train Loss CLS (blue), Train Loss Bbox (green)

## Package & Deploy AI Model

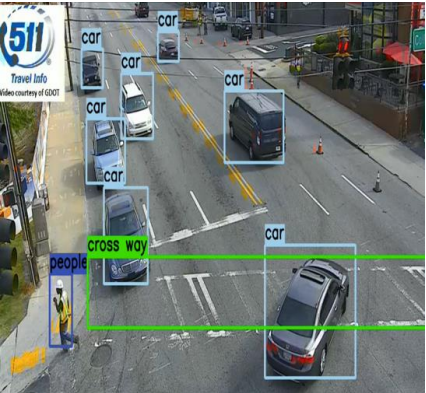


# Core Capabilities

- > **Manage and Label datasets - images and video**
- > **Classification of Images**
  - Identify object in an image and classify to one of the trained categories
- > **Detect Objects**
  - Analyze Images and Videos and detect several trained objects
- > **Auto labeling for rapid processing of raw datasets**
  - Partially trained models auto label raw datasets to create larger datasets
- > **Data Augmentation**
- > **RESTFul APIs to develop custom solutions and extend existing applications**

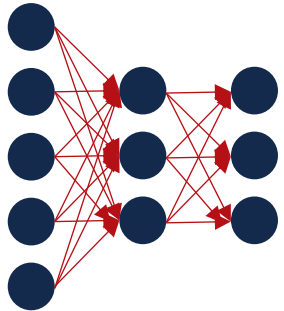
# Semi-Automatic Labeling using PowerAI Vision

Manually Label



Define Labels  
Manually Label Some  
Images / Video Frames

Train DL Model

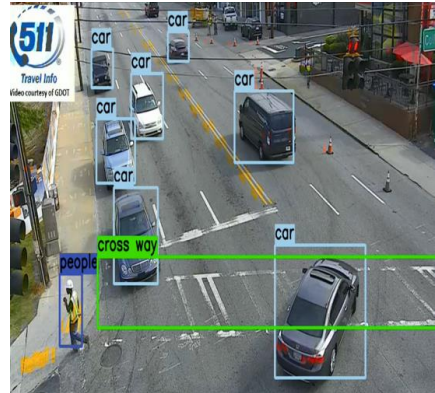


Use Trained DL Model



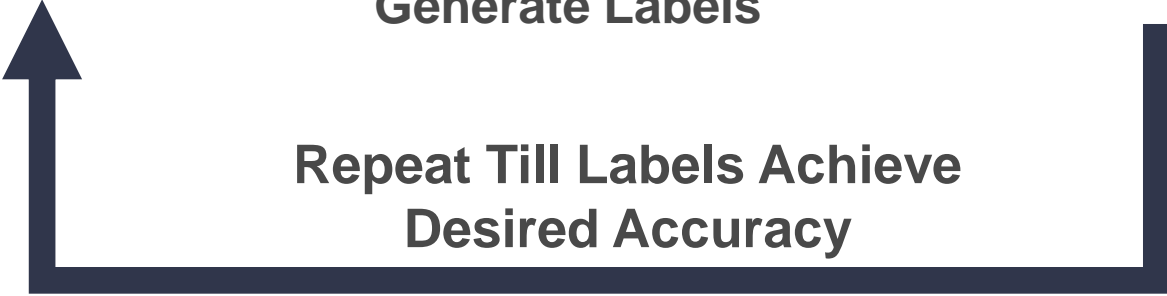
Run Trained DL Model  
on Entire Input Data to  
Generate Labels

Correct Labels  
on Some Data



Manually Correct  
Labels on Some Data

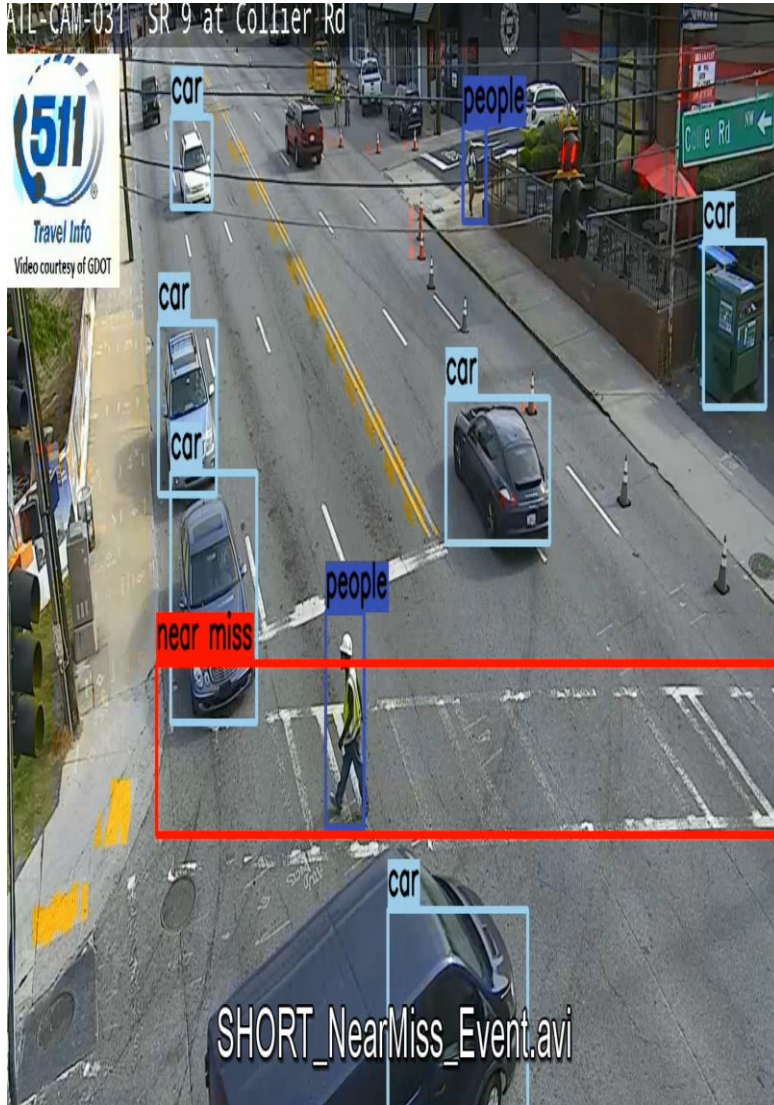
Repeat Till Labels Achieve  
Desired Accuracy





# Computer vision at play

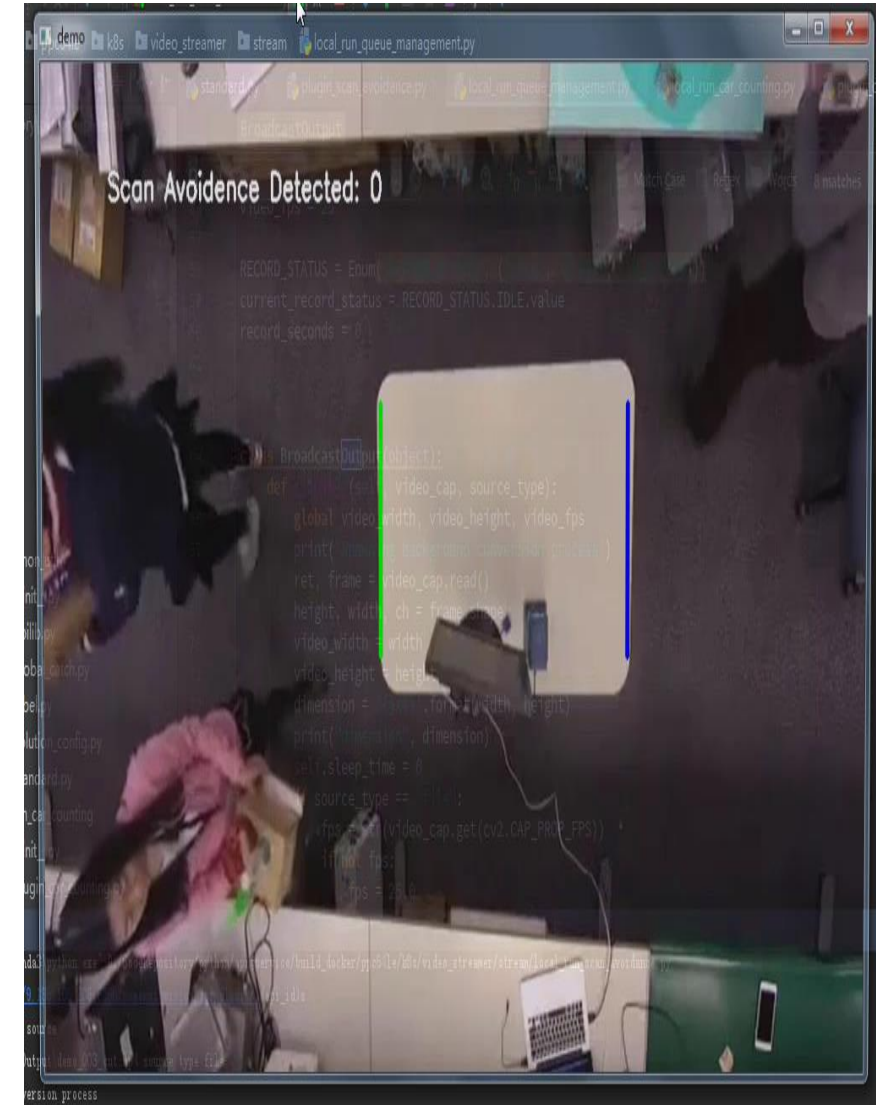
## Safer Cities



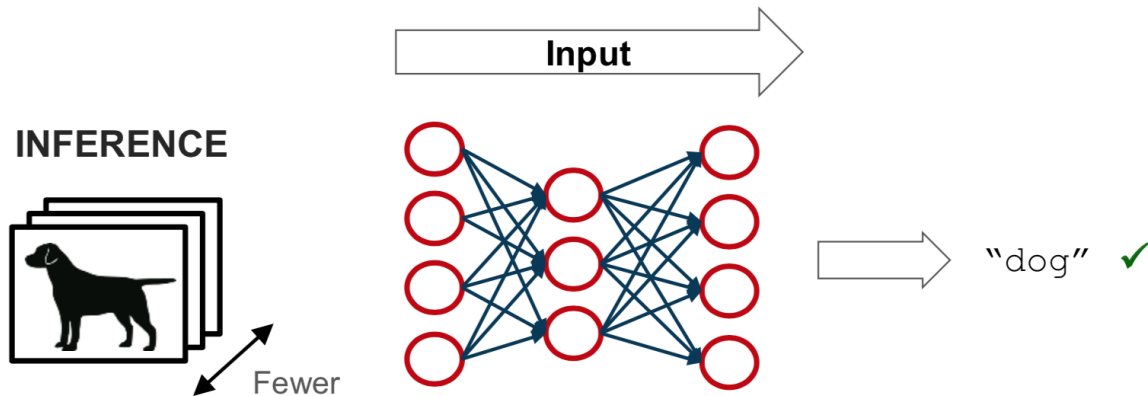
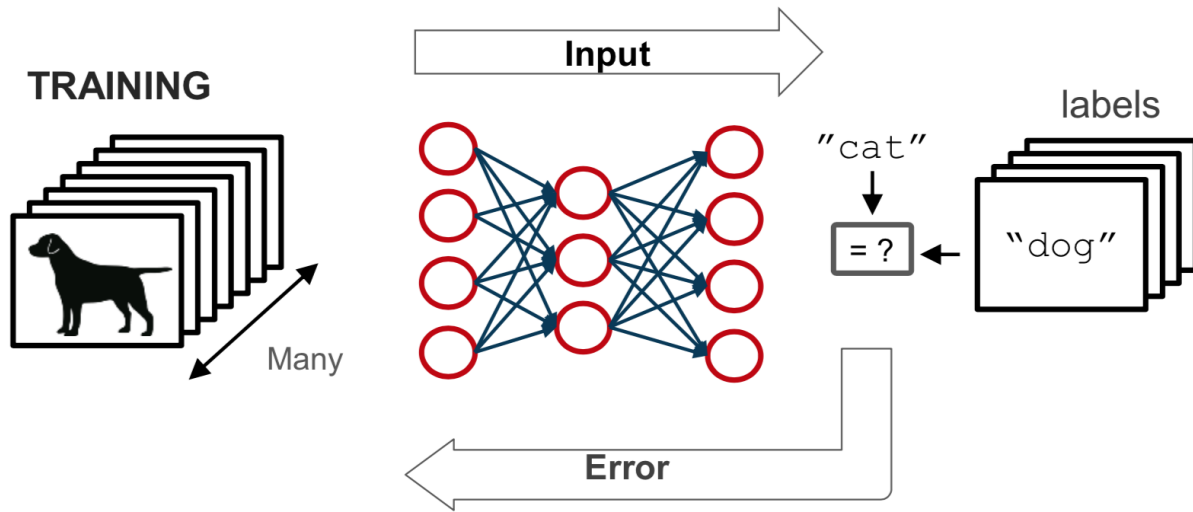
## Asset Management



## Retail



# Training vs Inference



## Inferencing

- Single forward pass
- Some layers can be merged / pruned
- Weights can be quantized (FP32 → INT16/8/4/...)
- Network choice - speed vs accuracy
- Latency / Throughput

# Deploying Trained Models

Data Center: Train model & Compile to Edge

PowerAI Vision



Trained  
DNN model



CPU  
+  
Accelerator

Model Parser & Compiler

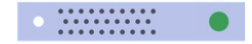
Model Optimization  
(Layer Merge + Quantization)

Output model + weights

Map to Different  
Platforms



Cloud or Edge



FPGAs, CPUs, GPUs



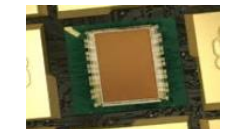
Xilinx Alveo U200



Embedded FPGA



Embedded  
GPU

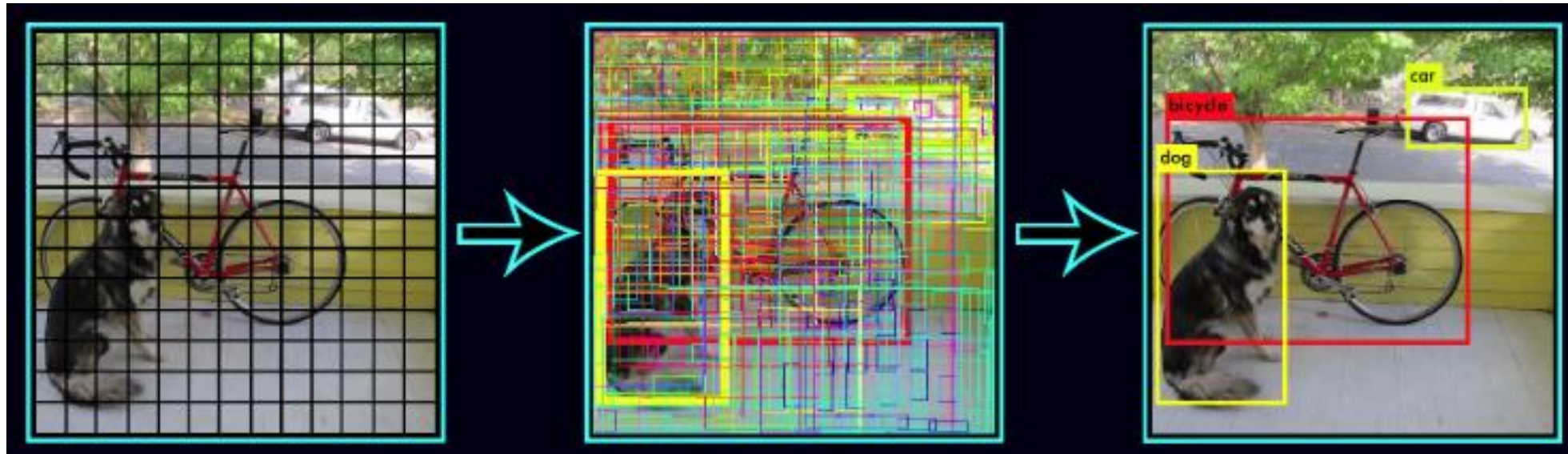


Neural network  
processor

# PowerAI Vision + Alveo U200



# Model Choice - YOLO (You Only Look Once)



- > Inference is fast (single network eval), and takes entire image into context
- > Tiny-yolo is even faster, though accuracy may be compromised
- > <https://pjreddie.com/darknet/yolov2/>
- > <https://youtu.be/VOC3huqHrss>

NOTE: PowerAI Vision supports multiple model types, but we focus here on YOLO

# Import images and label objects


Edit / rightsized

[← Done editing](#)

**CATEGORY** [+ Add New](#)

**OBJECTS** [+ Add New](#)

[papillon](#) [X](#)      [papillon \(1\)](#) [X](#)



The interface shows a gallery of Papillon images on the left, a main image with a bounding box and label in the center, and category/object lists on the right. The main image is a Papillon dog with long, fringed ears, black and white fur, and a white blaze on its face. A blue bounding box is drawn around the dog's head and ears, with the word 'papillon' written in white on a blue background above the box.

# Train labeled model

IBM PowerAI Vision

Data Sets Models Deployed Models

Filter by

- Images
- Videos

> Categories Edit

> Objects

Data set / rightsized

Total files: 338 Matching files: 338 Selected files: 0

Train model

Augment data Auto label Export data set

Assign category

Sort by: Se

Model

Create new model

rightsized\_model

Drag files

Type of training

- Image classification
- Object detection

Pre-processed

Base model

General (other scenarios)

Advanced options

Model selection

- Optimized for accuracy (faster R-CNN)
- Optimized for speed (tiny YOLO V2)

Model hyperparameters Restore defaults

Max iteration [100-100000] 40000 Momentum [0.1-1.0]

Learning rate [0-0.01] Weight decay [0-0.5] Ratio [0.5-1.0] 0.8

Return to data set Train

IBM PowerAI Vision

Data Sets Models Deployed Models

Training / MyDogs\_model

Browsing away from this page will not stop the training. To delete or stop the training early, click the 'stop training' button.

4.7 HOURS LEFT

Initialized 9/28/2018, 11:38:08 AM Stop training

MAX ITERATION 10000 RATIO 0.8

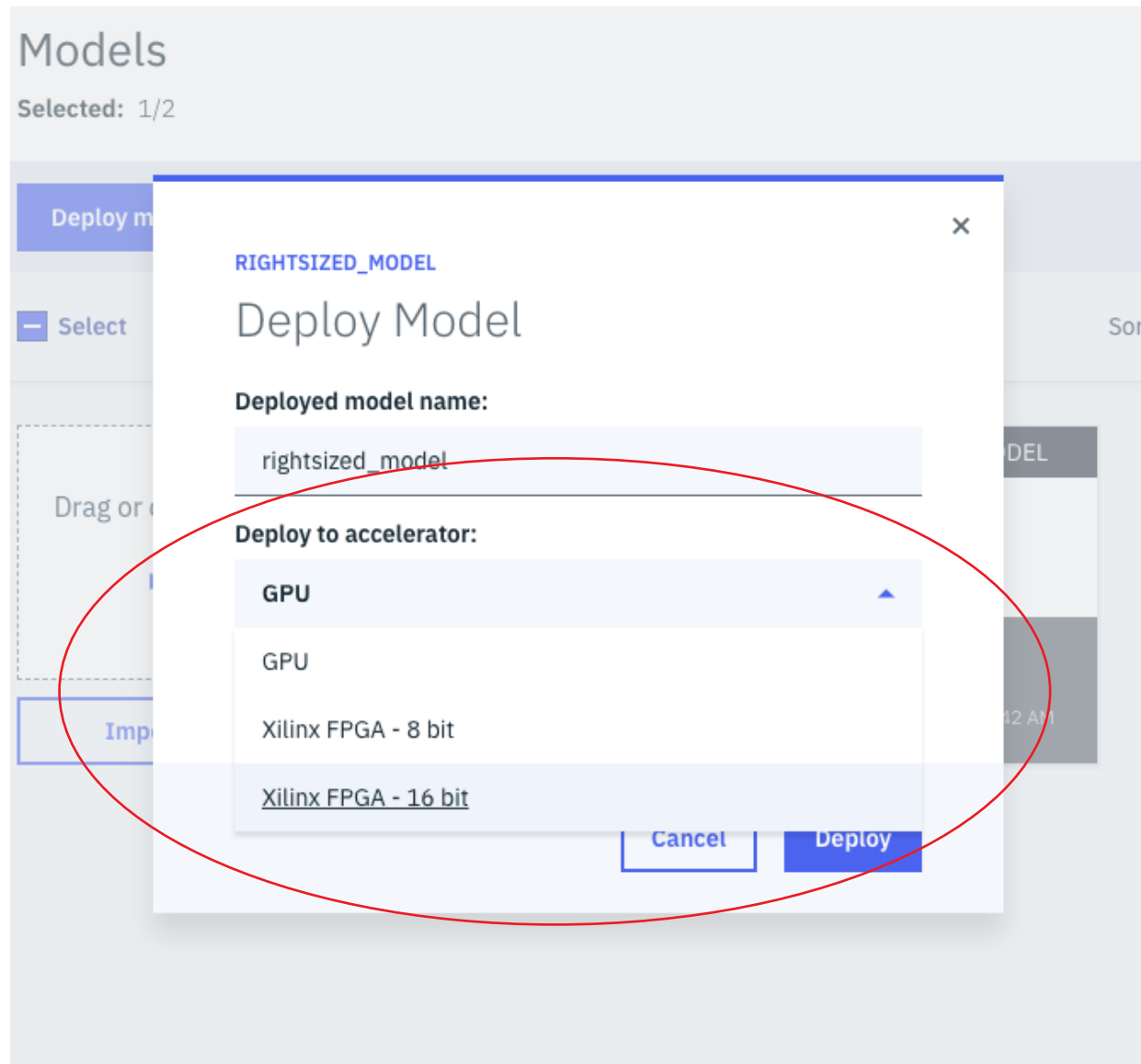
Loss VS Iteration

Train Loss

Loss VS Iteration

Test Loss

# Deploy Trained Model





# Deploy Trained Model

The screenshot shows the IBM PowerAI Vision interface. At the top, there are navigation tabs for 'Data Sets', 'Models', and 'Deployed Models'. The 'Deployed Models' tab is active. On the left, there is a sidebar with 'Data set' (Not found) and 'Objects'. The main content area shows a 'Deployed model / rightsized\_model' page. Below the title, there is a note about the API. A blue box highlights the API endpoint: 'api/dlapis/bf1c53fa-b931-4697-9...'. A red box highlights the 'Optimization' and 'Accelerator type' fields in a larger view. Below that, a table lists model details: Type (Object detection), Optimization (Optimized for speed (tiny YOLO V2)), Accelerator type (XILINX FPGA - 16 Bit), Model (rightsized\_model), Author (admin), and Created (9/28/2018, 9:28:28 AM). A red oval highlights the 'Optimization' and 'Accelerator type' fields in the table.

IBM PowerAI Vision    Data Sets    Models    **Deployed Models**    admin    Log out

← Deployed model / rightsized\_model

You can call the generated application programming interface (API) to run your deployed model. The API is unique to this model, and you cannot edit the API.

API endpoint: `api/dlapis/bf1c53fa-b931-4697-9...`

Type:	Object detection
<b>Optimization:</b>	Optimized for speed (tiny YOLO V2)
<b>Accelerator type:</b>	XILINX FPGA - 16 Bit
Model:	<a href="#">rightsized_model</a>
Author:	admin
Created:	9/28/2018, 9:28:28 AM

# PowerAI Vision - Microservices on Kubernetes

## > Microservices:

- >> UI
- >> Database
- >> Auth
- >> API

## > Training:

- >> Creates a new POD/Container
- >> Allocates GPU
- >> Trains
- >> Exits

## > Inferencing:

- >> Creates a new POD/Container
- >> Allocates Accelerator (GPU or FPGA)
- >> Provides Inferencing API

```
# /opt/powerai-vision/bin/helm.sh status amartey
LAST DEPLOYED: Wed Aug  1 13:51:20 2018
NAMESPACE: default
STATUS: DEPLOYED

RESOURCES:
==> v1/Service
NAME                                CLUSTER-IP    EXTERNAL-IP    PORT(S)          AGE
powerai-vision-amartey-keycloak     10.10.0.151   <none>         8080/TCP,8443/TCP 58d
powerai-vision-amartey-mongodb      10.10.0.93    <none>         27017/TCP         76d
powerai-vision-amartey-portal        10.10.0.201   <none>         9080/TCP          76d
powerai-vision-amartey-postgres     10.10.0.77    <none>         5432/TCP          76d
powerai-vision-amartey-taskanaly    10.10.0.246   <none>         5000/TCP          76d
powerai-vision-amartey-ui           10.10.0.18    <none>         80/TCP            76d

==> v1beta1/Deployment
NAME                                DESIRED    CURRENT    UP-TO-DATE    AVAILABLE    AGE
powerai-vision-amartey-keycloak     1          1          1              1             58d
powerai-vision-amartey-mongodb      1          1          1              1             76d
powerai-vision-amartey-portal        1          1          1              1             76d
powerai-vision-amartey-postgres     1          1          1              1             76d
powerai-vision-amartey-taskanaly    1          1          1              1             76d
powerai-vision-amartey-ui           1          1          1              1             76d

==> v1beta1/Ingress
NAME                                HOSTS        ADDRESS      PORTS    AGE
powerai-vision-amartey-ing *           127.0.0.1   80         76d

==> v1/Secret
NAME                                TYPE        DATA    AGE
powerai-vision-amartey-secrets      Opaque      6        76d

==> v1/ConfigMap
NAME                                DATA    AGE
powerai-vision-amartey-config      52      76d
```

# Scheduling FPGAs on Kubernetes

## > Device Plugin Framework

- >> Daemon on each node reports available FPGAs to master node
- >> Master node tracks allocation of FPGAs to pods that request it
- >> Master schedules pod to node
  - If there's no FPGA available, pod waits in scheduling until one is
- >> Node's daemon tells pod how to connect to FPGA

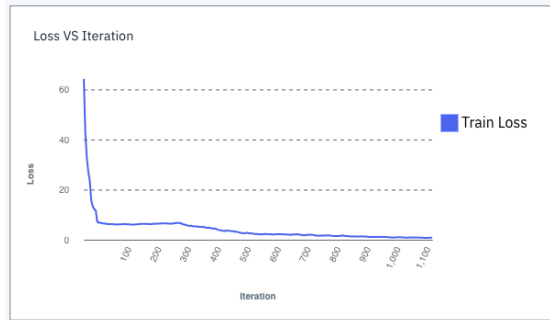
## > Daemon

- >> Node's daemon connects pod to required resources
  - PCI block devices (user and management PFs)
  - Mount paths (drivers)
  - Environment variables
- >> Deployed pod only given access to single allocated device
  - Does not need privileged access

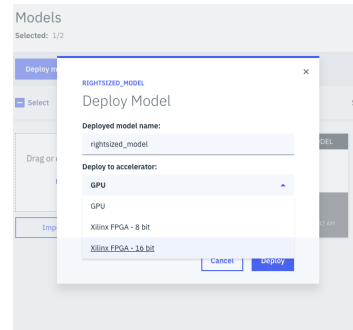


# Integration with Xilinx MLSuite

## Train YOLO



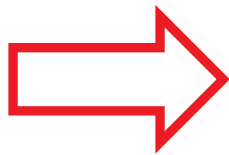
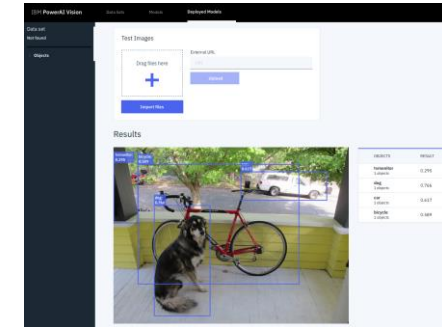
## Deploy



## Schedule Available FPGA



## Run Inference



## Inference Container

{ REST API }

Deploy.py

GPU or FPGA?

xDNN Compile  
& Quantize (if needed)

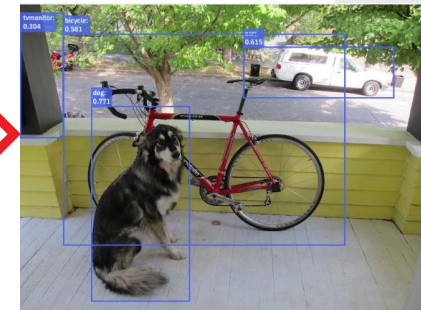
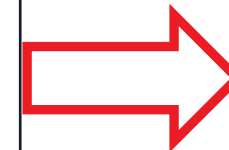
xDNN Runtime

Caffe Prototxt,  
Weights, & Anchors

Sample Training  
Images

```
▼ {webAPIId: "bf1c53fa-b931-4697-9ef7-985137b2e080", ...}
  ▼ classified: [...]
```

- ▶ 0: {confidence: 0.304, ymax: 246, label: "tvmonitor", image\_id: "image.png", ...}
- ▼ 1: {confidence: 0.771, ymax: 549, label: "dog", image\_id: "image.png", confidence: 0.771, image\_id: "image.png", label: "dog", xmax: 314, xmin: 132, ymax: 549, ymin: 188, ...}
- ▶ 2: {confidence: 0.615, ymax: 172, label: "car", image\_id: "image.png", ...}
- ▶ 3: {confidence: 0.581, ymax: 445, label: "bicycle", image\_id: "image.png", imageUrl: "https://github.com/pjreddie/darknet/raw/master/data/dog.jpg", result: "success", webAPIId: "bf1c53fa-b931-4697-9ef7-985137b2e080", ...}



# Container Logs

## Compile & Quantize

```
root      : INFO      INFO:root:Unzipped the snapshot files to /opt/deploy
root      : INFO      INFO:root:
root      : INFO      --- Compiling ---
root      : INFO
root      : INFO
root      : INFO      WARNING: Logging before InitGoogleLogging() is written to STDERR
root      : INFO      kwargs {'verbose': False, 'ddr': 256, 'networkfile':
'/opt/deploy/deploy.prototxt', 'dsp': 56, 'strategy': 'all', 'weights':
'/opt/deploy/model.caffemodel', 'memory': 5, 'generatefile':
'/opt/deploy/model.caffemodel_data/yolo.cmds'}
...
root      : INFO      *****
root      : INFO      * BUILDING NETWORK SCHEDULE
root      : INFO      *****
root      : INFO      Network Schedule ['data', 'conv0', 'relu0', 'pool1', 'conv2', 'relu2',
'pool3', 'conv4', 'relu4', 'conv5', 'relu5', 'conv6', 'relu6', 'pool7', 'conv8', 'relu8',
'conv9', 'relu9', 'conv10', 'relu10', 'pool11', 'conv12', 'relu12', 'conv13', 'relu13',
'conv14', 'relu14', 'conv15', 'relu15', 'conv16', 'relu16', 'pool17', 'conv18', 'relu18',
'conv19', 'relu19', 'conv20', 'relu20', 'conv21', 'relu21', 'conv22', 'relu22', 'conv23',
'relu23', 'conv24', 'relu24', 'conv26', 'relu26', 'pool27', 'concat28', 'conv29', 'relu29',
'conv30']
...
root      : INFO      *****
root      : INFO      * GENERATING OUTPUT FILES
root      : INFO      *****
root      : INFO      XDNN Command file: /opt/deploy/model.caffemodel_data/yolo.cmds
root      : INFO      XDNN JSON Report file: /opt/deploy/model.caffemodel_data/yolo.cmds.json
root      : INFO      OUTPUT REPORT:
root      : INFO      Unsupported Layers: 0
...
root      : INFO      CL_PLATFORM_VENDOR Xilinx
root      : INFO      CL_PLATFORM_NAME Xilinx
root      : INFO      CL_DEVICE_0: 0x24fda600
root      : INFO      CL_DEVICES_FOUND 1, using 0
root      : INFO      loading /opt/xfdsn/overlaybins/1525/overlay_3.xclbin
```

## Inference

```
root      : INFO      INFO:root:GET - imageUrl:
https://github.com/pjreddie/darknet/raw/master/data/dog.jpg.
root      : INFO      INFO:root:Saving original file to
/tmp/caffe_images_uploads/image.png.
root      : INFO      INFO:root:Running FPGA inferencing.
root      : INFO      INFO:root:Preparing Input...
root      : INFO      INFO:root:Running 1 image(s)
root      : INFO      INFO:root:Results for image 0:
/tmp/caffe_images_uploads/image.png
root      : INFO      INFO:root:Found 4 boxes
root      : INFO      INFO:root:Obj 0: tvmonitor
root      : INFO      INFO:root:                score = 0.304418
root      : INFO      INFO:root:                (xlo,ylo) = (0,246)
root      : INFO      INFO:root:                (xhi,yhi) = (79,15)
root      : INFO      INFO:root:Obj 1: dog
root      : INFO      INFO:root:                score = 0.771104
root      : INFO      INFO:root:                (xlo,ylo) = (132,549)
root      : INFO      INFO:root:                (xhi,yhi) = (314,188)
root      : INFO      INFO:root:Obj 2: car
root      : INFO      INFO:root:                score = 0.614885
root      : INFO      INFO:root:                (xlo,ylo) = (415,172)
root      : INFO      INFO:root:                (xhi,yhi) = (694,78)
root      : INFO      INFO:root:Obj 3: bicycle
root      : INFO      INFO:root:                score = 0.581320
root      : INFO      INFO:root:                (xlo,ylo) = (80,445)
root      : INFO      INFO:root:                (xhi,yhi) = (603,54)
root      : INFO      INFO:root:Finished images 1-1 of 1
root      : INFO      INFO:root:  Results: [{'confidence': 0.304, 'ymax': 246,
'label': 'tvmonitor', 'image_id': 'image.png', 'xmax': 79, 'xmin': 0, 'ymin': 15},
{'confidence': 0.771, 'ymax': 549, 'label': 'dog', 'image_id': 'image.png', 'xmax': 314,
'xmin': 132, 'ymin': 188}, {'confidence': 0.615, 'ymax': 172, 'label': 'car',
'image_id': 'image.png', 'xmax': 694, 'xmin': 415, 'ymin': 78}, {'confidence': 0.581,
'ymax': 445, 'label': 'bicycle', 'image_id': 'image.png', 'xmax': 603, 'xmin': 80,
'ymin': 54}]]
```

# Additional Info

- > **Come see us - TODO**
- > **PowerAI Vision**
  - >> <https://www.ibm.com/us-en/marketplace/ibm-powerai-vision>
- > **Accelerated Power Systems:**
  - >> <https://www.ibm.com/it-infrastructure/power/accelerated-computing>
- > **PowerAI Enterprise**
  - >> <https://www.ibm.com/us-en/marketplace/deep-learning-platform>





XILINX  
DEVELOPER  
FORUM

