KCM19/PCM19

Antwerp, Belgium

HITACHI
Inspire the Next

Taking the Complexities Out of
Machine Learning – Update!

"Hey Ray!" and more…

Mark Hall Ph.D
Chief Machine Learning Architect
November 2019

Ken Wood
VP of Hitachi Vantara Labs
November 2019

@KenWoodOnTech

"What's Cooking at Hitachi Vantara Labs"

HITACHI
Inspire the Next

- Hitachi Vantara Labs, *formerly "Pentaho Labs"*
- "**R**esearch, **d**evelopment & **I**nnovation" for Pentaho related subjects
- Expanding the use cases that Pentaho can solve
- Making Machine/Deep Learning easier to use
- Increasing Data Science Productivity
- Advancing Data Science for Pentaho Users
- *The journey continues…*

A Different Way of Doing Machine Learning

HITACHI
Inspire the Next

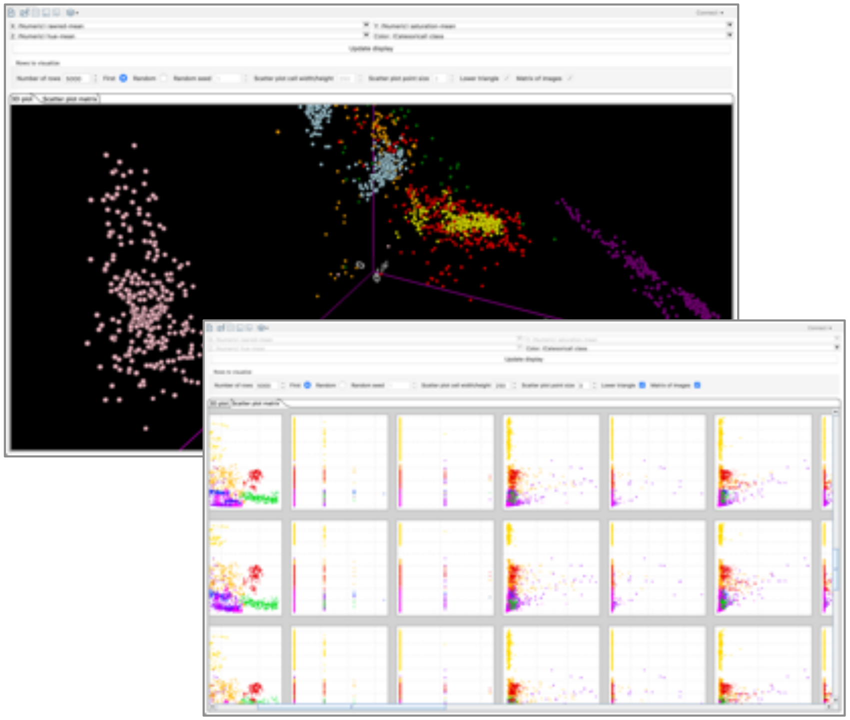
- HV Labs has created 3 methods of doing ML with Pentaho
  - Original Weka integration with Pentaho Data Integration
  - Bring-Your-Own-Code: R and Python executor steps
  - Make Machine Learning easier to use: “no coding” with PMI
    - Continued expansion

- “Plugin Machine Intelligence”
  - 2014 introduced the Data Science Pack
  - PMI is Data Science pack – “*Volume 2?*”
  - New version coming!

Quick Poll!

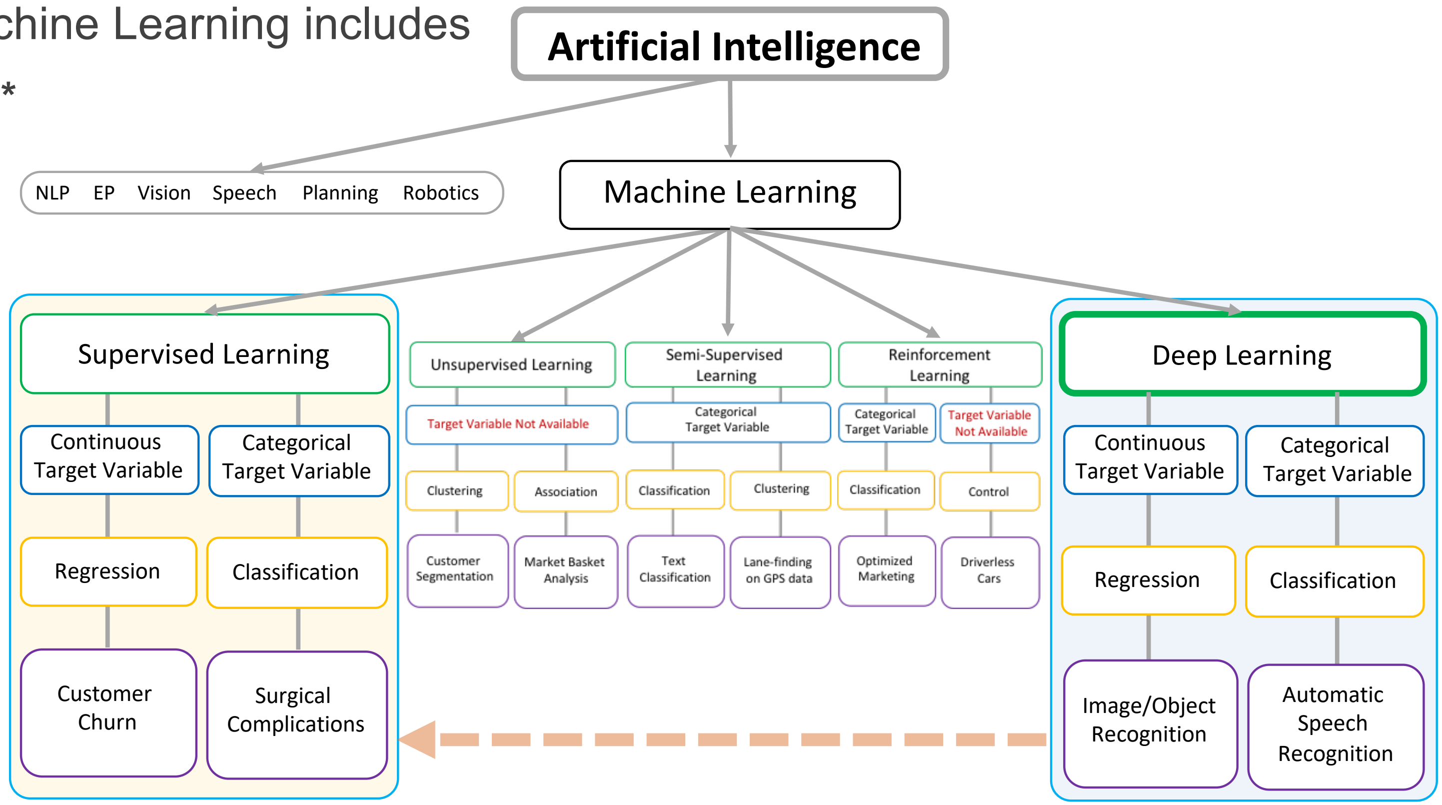
The *Intersection* of 6 Execution Engines

- The first phase focused on commonality and Supervised ML
  - Scikit-learn from python
  - MLR from R
  - MLlib from Apache Spark
  - Weka
  - DL4j – deep learning
  - ***Keras/Tensorflow* –** deep learning
- This is no longer the case
- PMI is a framework built to be extensible



Artificial Intelligence Has Many Subdomains

- The Subdomain of Machine Learning includes
  - **Supervised Learning***
  - Unsupervised Learning
  - Semi-supervised Learning
  - Reinforcement Learning
  - **Deep Learning***
- Classical Machine Learning vs Deep Learning

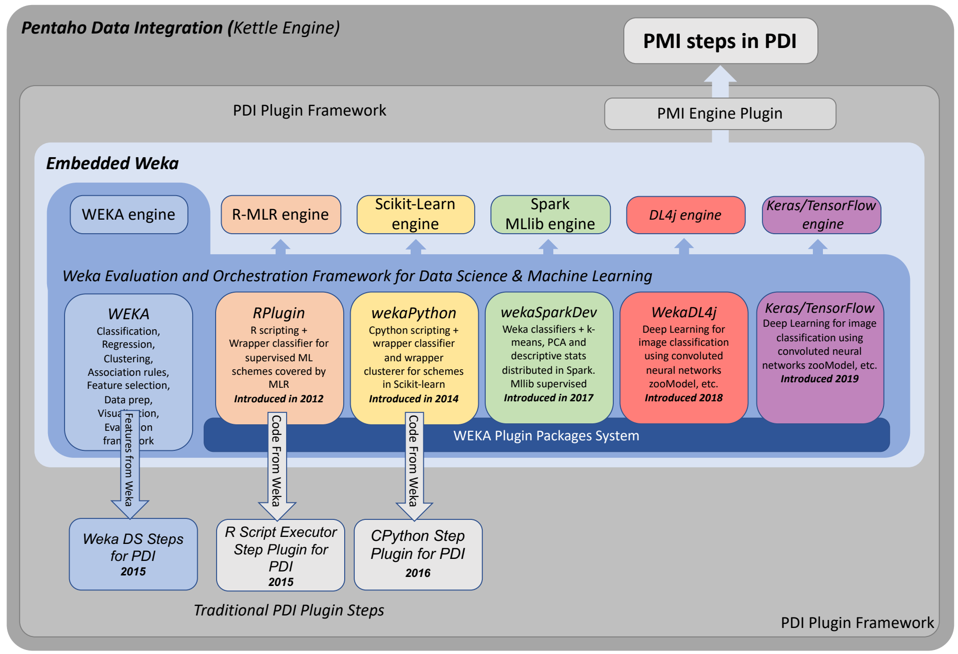


** Implemented in PMI*

The Plugin Machine Intelligence Framework

HITACHI
Inspire the Next

- All 6 execution engines (libraries) are based on this supervised framework
  - **Scikit-learn** from python
  - **MLR** from R
  - **MLlib** from Apache Spark
  - **Weka** from Weka
  - **DL4j -** Deep Learning for java
  - **Keras/Tensorflow** from Python



- PMI is a framework built to be extensible
  - "a plugin of plugins" – more algorithms
  - "a framework of frameworks" – more ML types

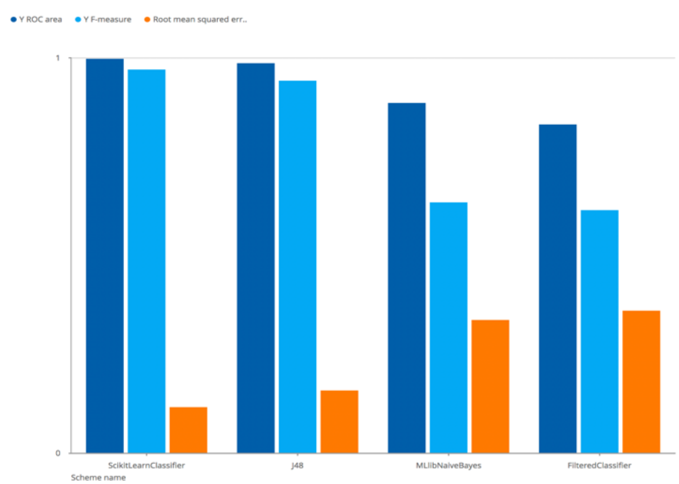
When Machine Learning is Easy To Use …

HITACHI
Inspire the Next

- Domain Experts can use ML
- Data Engineers can use ML
- Increase productivity of Data Scientists
  - DS can focus on the hard problems not the mundane tasks
- Fail Fast, Fail Cheap, Fail Productively
  - Machine Learning Model Exploration
- Complexity has now moved from ML to Data Preparation
  - Putting data in the proper form to ask the right question

Uniform Performance Metrics

- Whatever the algorithm or engine combination, the accuracy measurements can be compared uniformly
- We use this for Machine Learning Model Management and more

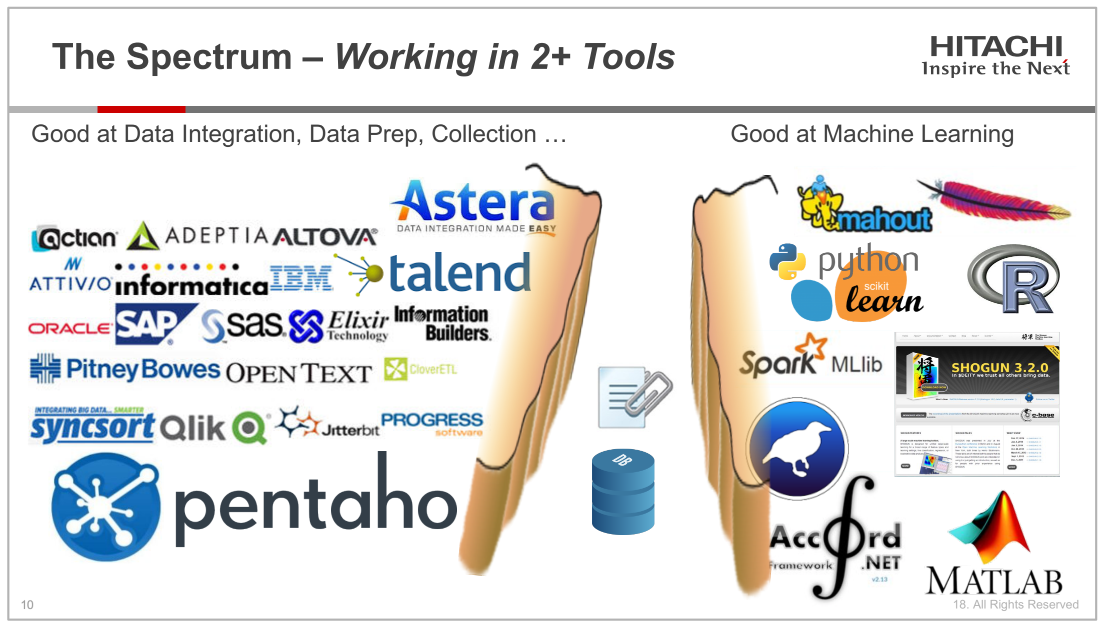


- `Scheme name`
- `Scheme options`
- `Evaluation mode`
- `Unclassified instances`
- `Correctly classified instances`
- `Incorrectly classified instances`
- `Percent correct`
- `Percent incorrect`
- `Mean absolute error`
- `Root mean squared error`
- `Relative absolute error`
- `Root relative squared error`
- `Total number of instances`
- `Kappa statistics`
- *`class`* `TP_rate`
- *`class`* `FP_rate`
- *`class`* `Precision`
- *`class`* `Recall`
- *`class`* `F-measure`
- *`class`* `MCC`
- *`class`* `ROC area`
- *`class`* `PRC area`

Need of New Industry Term? When Two Worlds Merge

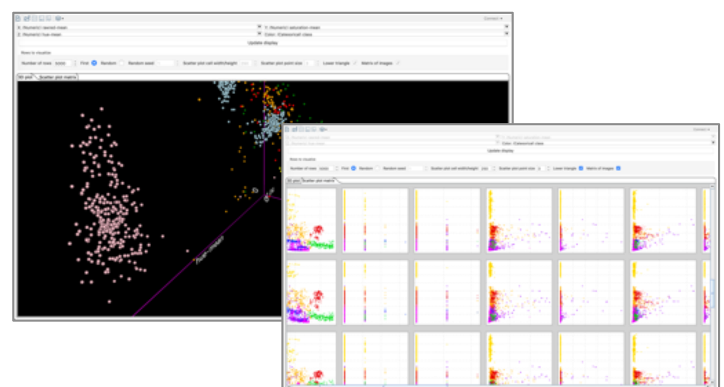
HITACHI
Inspire the Next

- Extract Transform for Machine Learning – ETML - ETMI
- Data Integration for Machine Learning - DIML - DIMI
  - PDIML
  - PDIMI
  - PDML
  - PDMI
- You get the idea!
- Add IoT and this really changes things



Hitachi Vantara Labs Update – PMI v1.5

- New PMI v1.5 Release for PDI being tested
  - New Features – more GPU utilization
    - Deep Learning with Keras/TensorFlow
    - Transfer learning
    - **eXtreme Boosting Classifier & Regressor**
    - Spark MLib 2.4
    - Scatter Matrix and 3D visuals for data exploration



- Demonstrated at NEXT19
- Target release, before the end of 2019

What Could be Coming – *Vision, not Roadmap*

HITACHI
Inspire the Next

- Items we are looking in at the future in no particular order
  - Unsupervised Machine Learning
    - batch clustering
    - Streaming anomaly detection for ML/edge/IoT
  - Model Server
    - ML Model Life Cycle Management
    - Ensembles
  - eXplainable AI – XAI
    - Knowledge Discovery

PMI Concepts to Consider in the Future

HITACHI
Inspire the Next

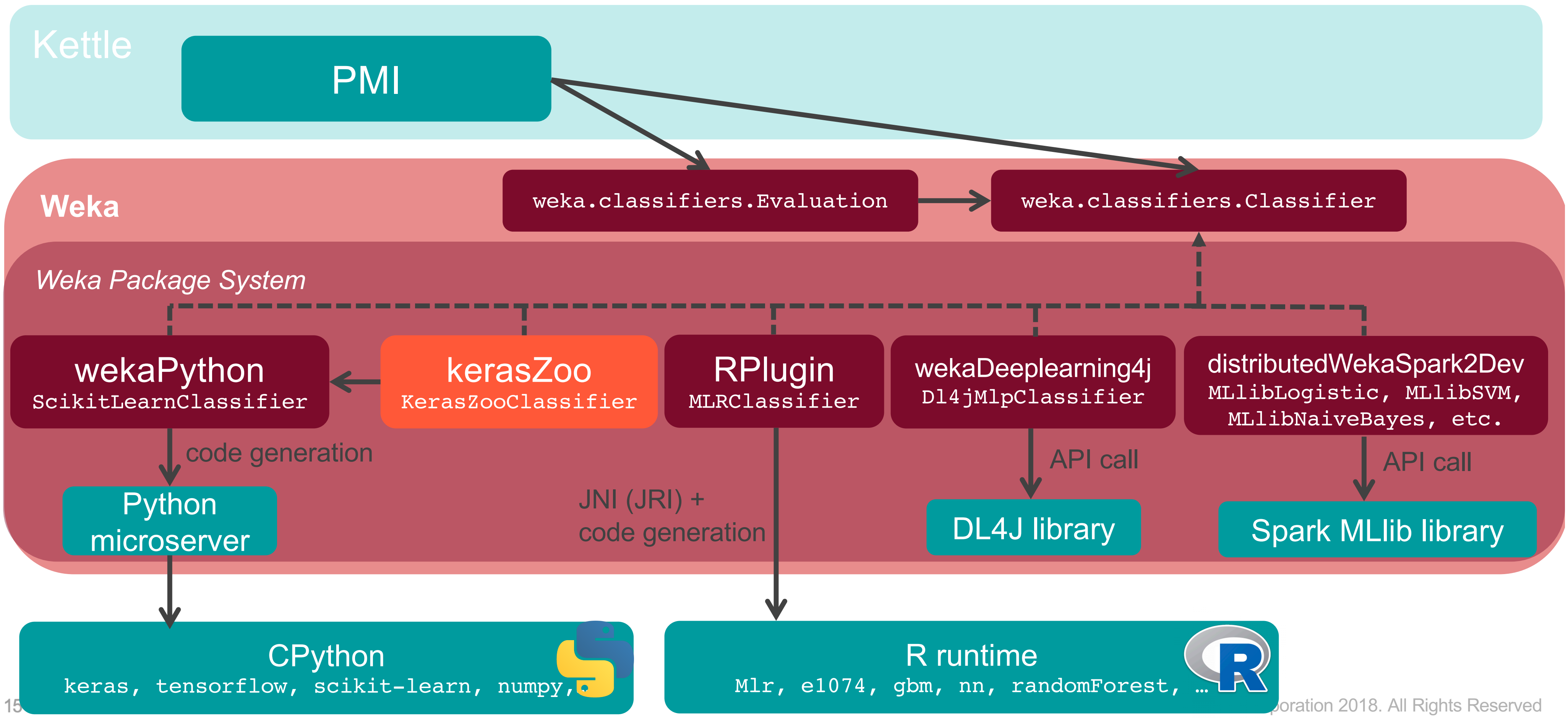
- Hidden results from Machine Learning models
- Create your own Deep Learning models vs. zooModel
- Broad Spectrum Deep Learning Models
  - What I like to call "Junk Drawer Models"
  - Decision tree of DL models versus monolithic models
- Ensemble of models to improve model predictions
  - Voting models and quorum
- More Artificial Intelligence for more Machine Intelligence

HITACHI
Inspire the Next

PMI 1.5: Keras/TF Integration

Technical Deep Dive

Weka provides the interoperability

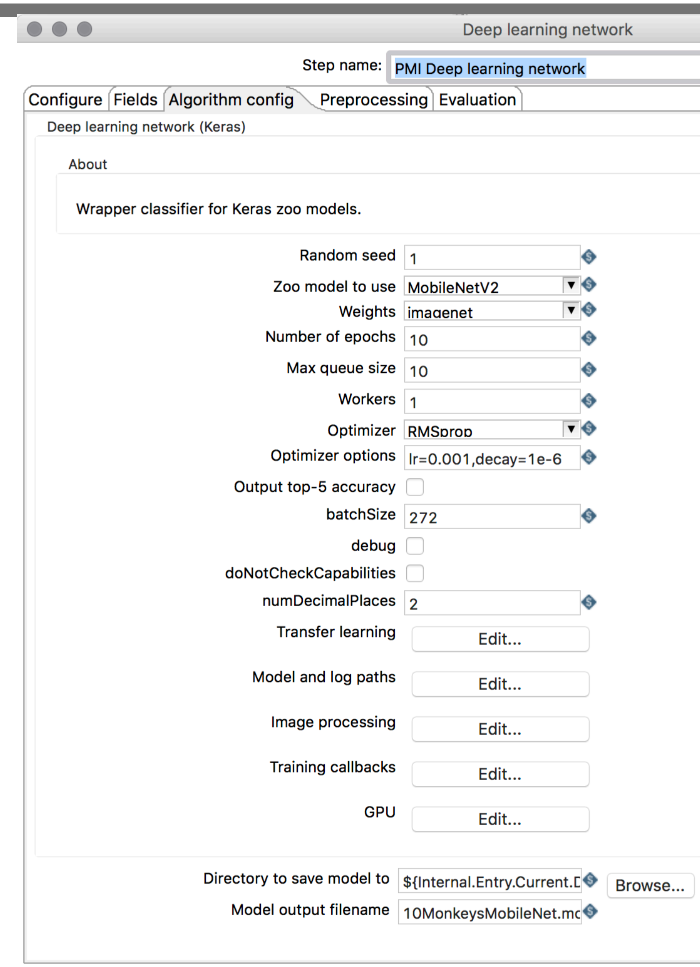


Weka provides the interoperability

- Good for PMI
  - Single API to talk to
  - Leverage Weka's stable evaluation routines
  - PMI plugin "engines" basically layer metadata on classifiers from Weka packages
- Good for Weka
  - New integration/interoperability gets realized in OS Weka first
- Also new in PMI 1.5: **xgboost** integration via `ScikitLearnClassifier` in Weka
  - Not actually offered int scikit-learn, but xgboost has a scikit-learn API…

PMI Keras/TF features

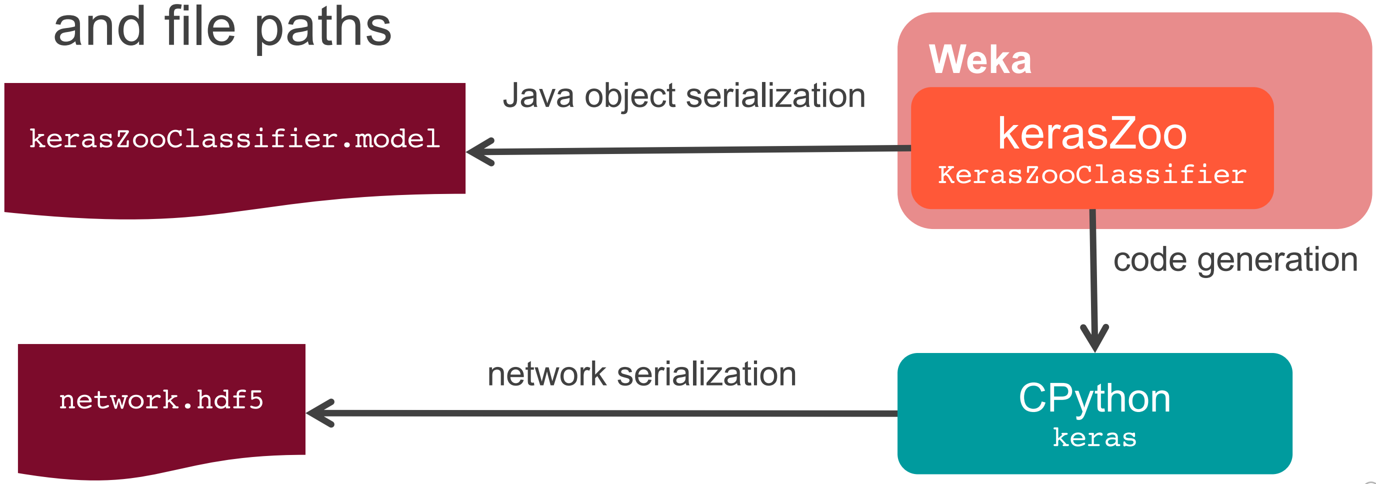
- Zoo models (Keras applications) to begin with
- Train network from scratch, or start with *imagenet* pre-trained weights
- Transfer learning by freezing layers and training new top-level dense layers
- Zoo model-specific image preprocessing
- Training callbacks for epoch metrics, learning rate modification, model checkpoints etc.
- (Multi) GPU options



Code generation

HITACHI
Inspire the Next

- Generated code, layer lists and epoch stats dumped to Spoon log
- Trained network graph saved to file system in hdf5 format
  - Reload and continue training
- Serialized Weka wrapper classifier maintains hyperparameter settings and file paths



Training code (10 Monkeys dataset)

```
- PMI Deep learning network.0 - from keras.applications.mobilenet_v2 import preprocess_input
- PMI Deep learning network.0 - import keras.backend as K
- PMI Deep learning network.0 - from keras import utils
- PMI Deep learning network.0 - from keras.models import load_model
- PMI Deep learning network.0 - from keras.callbacks import Callback, CSVLogger, ReduceLROnPlateau, LearningRateScheduler, ModelCheckpoint
- PMI Deep learning network.0 -
- PMI Deep learning network.0 - K.clear_session()
- PMI Deep learning network.0 -
- PMI Deep learning network.0 - datagen= ImageDataGenerator(preprocessing_function=preprocess_input,rescale=None,samplewise_center=False,sampl
- PMI Deep learning network.0 -
- PMI Deep learning network.0 - keras_zoo_1663988340 = applications.MobileNetV2(include_top=False,weights='imagenet',input_shape=(224,224,3))
- PMI Deep learning network.0 -
- PMI Deep learning network.0 - keras_zoo_train_1663988340['filename'] = keras_zoo_train_1663988340['filename'].astype(str)
- PMI Deep learning network.0 - keras_zoo_train_1663988340['class'] = keras_zoo_train_1663988340['class'].astype(str)
- PMI Deep learning network.0 -
- PMI Deep learning network.0 - generator = datagen.flow_from_dataframe(keras_zoo_train_1663988340, directory='/Users/mhall/datasets/image/10-mo
- PMI Deep learning network.0 -

- PMI Deep learning network.0 - for layer in keras_zoo_1663988340.layers:
- PMI Deep learning network.0 -     layer.trainable=False
- PMI Deep learning network.0 -
- PMI Deep learning network.0 - x = keras_zoo_1663988340.output
- PMI Deep learning network.0 - x = GlobalAveragePooling2D()(x)
- PMI Deep learning network.0 -
- PMI Deep learning network.0 - fc_layers = [256]
- PMI Deep learning network.0 - for fc in fc_layers:
- PMI Deep learning network.0 -     x = Dense(fc, activation='relu')(x)
- PMI Deep learning network.0 -     x = Dropout(0.4)(x)
- PMI Deep learning network.0 -
- PMI Deep learning network.0 - preds = Dense(10, activation='softmax')(x)
- PMI Deep learning network.0 -
- PMI Deep learning network.0 - keras_zoo_transfer_1663988340 = Model(inputs=keras_zoo_1663988340.input, outputs=preds)
- PMI Deep learning network.0 -
- PMI Deep learning network.0 - optimizer=optimizers.RMSprop(lr=0.001,decay=1e-6)
- PMI Deep learning network.0 - keras_zoo_transfer_1663988340.compile(optimizer, loss='categorical_crossentropy', metrics=['accuracy'])
- PMI Deep learning network.0 -
```

Data generator and network definition

Transfer learning configuration

Training code

```
- PMI Deep learning network.0 - class ComputeDeltaTime(Callback):
- PMI Deep learning network.0 -     def on_epoch_end(self, epoch, logs):
- PMI Deep learning network.0 -         logs['time'] = datetime.now().time()
- PMI Deep learning network.0 -
- PMI Deep learning network.0 - epoch_time = ComputeDeltaTime()
- PMI Deep learning network.0 -
- PMI Deep learning network.0 - csv_logger = CSVLogger('/Users/mhall/trainingProg.txt',append=False,separator=',')
- PMI Deep learning network.0 - def schedule(epoch):
- PMI Deep learning network.0 -     if epoch < 5:
- PMI Deep learning network.0 -         return 0.001
- PMI Deep learning network.0 -     elif epoch < 10:
- PMI Deep learning network.0 -         return 0.0001
- PMI Deep learning network.0 -     else:
- PMI Deep learning network.0 -         return 0.00005
- PMI Deep learning network.0 -
- PMI Deep learning network.0 - lr_scheduler = LearningRateScheduler(schedule, verbose=1)
- PMI Deep learning network.0 -
- PMI Deep learning network.0 - keras_zoo_transfer_1663988340.fit_generator(generator=generator, steps_per_epoch=ceil(1097.0 / 32), epochs=10, max_queue_size=10, callbacks=[epoch_time,csv_logger,lr_scheduler])
- PMI Deep learning network.0 -
- PMI Deep learning network.0 - for i, layer in enumerate(keras_zoo_transfer_1663988340.layers):
- PMI Deep learning network.0 -     print(i, layer.name, type(layer), 'trainable =', layer.trainable)
- PMI Deep learning network.0 -
- PMI Deep learning network.0 - keras_zoo_transfer_1663988340.save('/Users/mhall/datasets/image/10-monkeys-kaggle/10MonkeysMobileNet.hdf5')
```

Training callbacks definition:
- Custom epoch timestamp
- Learning rate schedule
- CSV logger

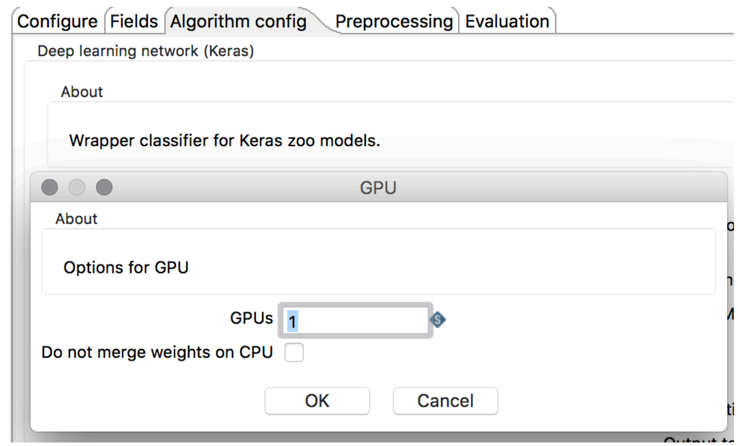
Network save

- Variables (java props, environment, Kettle)
  - Network load/save, images location and epoch logging

GPU

HITACHI
Inspire the Next

- Default setting
  - Training on 1 GPU or CPU if no GPUs visible
- Multi-GPU training/inference
  - `keras.utils.multi_gpu_model`
  - Tensorflow back end only
  - Train on GPU(s); inference on CPU, and vice versa




```
PMI Deep learning network.0 - Checking available gpus:
PMI Deep learning network.0 -
PMI Deep learning network.0 - from keras import backend as K
PMI Deep learning network.0 -
PMI Deep learning network.0 - def _normalize_device_name(name):
PMI Deep learning network.0 -     name = '/' + ':'.join(name.lower().replace('/', '').split(':')[-2:])
PMI Deep learning network.0 -     return name
PMI Deep learning network.0 -
PMI Deep learning network.0 - z = [x.name for x in K.get_session().list_devices()]
PMI Deep learning network.0 - available_devices = [_normalize_device_name(name) for name in z]
PMI Deep learning network.0 - gpus = len([x for x in available_devices if '/gpu:' in x])
PMI Deep learning network.0 - Output from python:
PMI Deep learning network.0 - Number of available GPUs: 0
```

PMI Roadmap (tentative)

- Supervised heterogenous model ensembles
  - Vote, Stacking
- Unsupervised
  - Batch/Incremental clustering
  - Streaming anomaly detection
- DL network graph editor
  - Translators to write Keras, TF, etc.
- Model server

TAIAO

- Collaboration between Universities of Waikato, Auckland and Cantebury, and Beca and NZ MetService
- NZ $13m government funding over seven years
- New ML methods for time series and data streams
  - Big data in real time
  - Environmental focus


Demo Time!
HITACHI
Inspire the Next

Humans vs. the Machine(s)

- MURA dataset v1.1
  - **MU**sculoskeletal **RA**diographs
  - Over 14,000 images

Injury Detection Only

- More public datasets available
  - MRI
  - More x-rays / dental
  - Skin

Will your model perform as well as radiologists in detecting abnormalities in musculoskeletal X-rays?

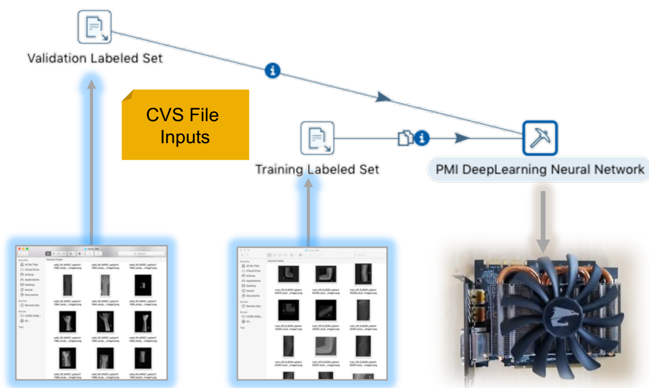
Rank	Date	Model	Kappa
		Best Radiologist Performance *Stanford University* Rajpurkar & Irvin et al., 17	0.778
1	Nov 30, 2018	base-comb2-xuan-v3(ensemble) *jzhang Availink*	**0.843**
2	Nov 06, 2018	base-comb2-xuan(ensemble) *jtzhang Availink*	0.834
3	Oct 06, 2018	muti_type (ensemble model) *SCU_MILAB*	0.833
4	Oct 02, 2018	base-comb4(ensemble) *jtzhang Availink*	0.824
5	Nov 08, 2018	base-comb2-jun2(ensemble)	0.814
5	Nov 07, 2018	base-comb2-ping(ensemble)	0.814
6	Aug 22, 2018	base-comb3(ensemble)	0.805
7	Sep 14, 2018	double_res(ensemble model) *SCU_MILAB*	0.804
8	[illegible] 2018	double-dense-Axy-Axyf512 *ensemble*	0.795
9	[illegible] 24, 2018	he_j	0.775
10	Aug 19, 2018	ianpan (ensemble) *RIH 3D Lab*	0.774
11	Jul 24, 2018	he_j	0.774
12	Jun 17, 2018	gcm (ensemble) *Peking University*	0.773
12	Sep 10, 2018	ty101 *single model*	0.773
13	Aug 31, 2018	he_j	0.764

Training DL Models

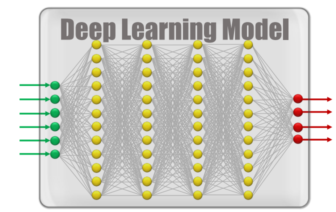
- PDI transformations for building DL Models with PMI

MURA-v1.1 — pi@raspberrypi: ~ — more train_image_ELBOW_pa...

```
MURA-v1.1/train/XR_ELBOW/patient04904/study1_positive/image1.png
MURA-v1.1/train/XR_ELBOW/patient04904/study1_positive/image2.png
MURA-v1.1/train/XR_ELBOW/patient04905/study1_positive/image1.png
MURA-v1.1/train/XR_ELBOW/patient04905/study1_positive/image2.png
MURA-v1.1/train/XR_ELBOW/patient04906/study1_positive/image1.png
MURA-v1.1/train/XR_ELBOW/patient04906/study1_positive/image2.png
MURA-v1.1/train/XR_ELBOW/patient04907/study1_positive/image1.png
MURA-v1.1/train/XR_ELBOW/patient04907/study1_positive/image2.png
MURA-v1.1/train/XR_ELBOW/patient04908/study1_positive/image1.png
MURA-v1.1/train/XR_ELBOW/patient04908/study1_positive/image2.png
MURA-v1.1/train/XR_ELBOW/patient02188/study1_positive/image1.png
MURA-v1.1/train/XR_ELBOW/patient02188/study1_positive/image2.png
MURA-v1.1/train/XR_ELBOW/patient02188/study1_positive/image3.png
MURA-v1.1/train/XR_ELBOW/patient02188/study1_positive/image4.png
MURA-v1.1/train/XR_ELBOW/patient04909/study1_positive/image1.png
MURA-v1.1/train/XR_ELBOW/patient04909/study1_positive/image2.png
MURA-v1.1/train/XR_ELBOW/patient04910/study1_positive/image1.png
MURA-v1.1/train/XR_ELBOW/patient04910/study1_positive/image2.png
MURA-v1.1/train/XR_ELBOW/patient04911/study1_positive/image1.png
MURA-v1.1/train/XR_ELBOW/patient04912/study1_positive/image1.png
MURA-v1.1/train/XR_ELBOW/patient04912/study1_positive/image2.png
MURA-v1.1/train/XR_ELBOW/patient04912/study1_positive/image3.png
MURA-v1.1/train/XR_ELBOW/patient04913/study1_positive/image1.png
MURA-v1.1/train/XR_ELBOW/patient04913/study1_positive/image2.png
MURA-v1.1/train/XR_ELBOW/patient04914/study1_positive/image1.png
MURA-v1.1/train/XR_ELBOW/patient04914/study1_positive/image2.png
MURA-v1.1/train/XR_ELBOW/patient04915/study1_positive/image1.png
:
```



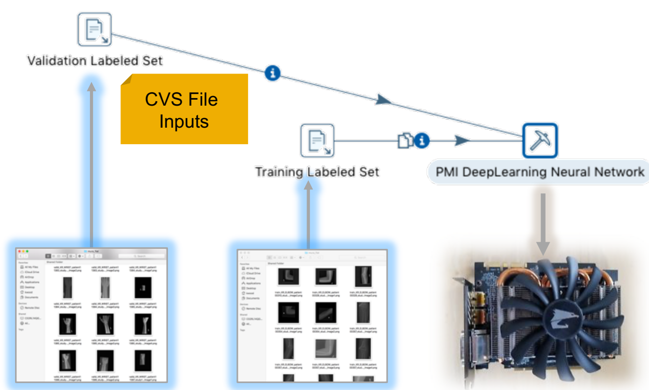
Body Part Identification



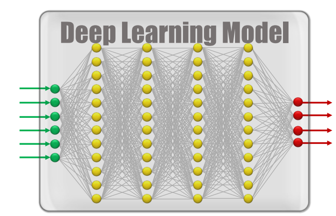
Classes:
Shoulder, Humerus, Elbow, Forearm, Hand

MURA-v1.1 — pi@raspberrypi: ~ — more valid_labeled_studies.c...

```
:MURA-v1.1/valid/XR_WRIST/patient11381/study1_negative/,0
MURA-v1.1/valid/XR_WRIST/patient11382/study1_negative/,0
MURA-v1.1/valid/XR_WRIST/patient11383/study1_negative/,0
MURA-v1.1/valid/XR_WRIST/patient11384/study1_negative/,0
MURA-v1.1/valid/XR_WRIST/patient11385/study1_negative/,0
MURA-v1.1/valid/XR_WRIST/patient11386/study1_negative/,0
MURA-v1.1/valid/XR_WRIST/patient11387/study1_negative/,0
MURA-v1.1/valid/XR_WRIST/patient11388/study1_negative/,0
MURA-v1.1/valid/XR_WRIST/patient11389/study1_negative/,0
MURA-v1.1/valid/XR_WRIST/patient11390/study1_negative/,0
MURA-v1.1/valid/XR_WRIST/patient11391/study1_negative/,0
MURA-v1.1/valid/XR_FOREARM/patient11392/study1_positive/,1
MURA-v1.1/valid/XR_FOREARM/patient11393/study1_positive/,1
MURA-v1.1/valid/XR_FOREARM/patient11394/study1_positive/,1
MURA-v1.1/valid/XR_FOREARM/patient11395/study1_positive/,1
MURA-v1.1/valid/XR_FOREARM/patient11396/study1_positive/,1
MURA-v1.1/valid/XR_FOREARM/patient11397/study1_positive/,1
MURA-v1.1/valid/XR_FOREARM/patient11398/study1_positive/,1
MURA-v1.1/valid/XR_FOREARM/patient11328/study1_positive/,1
MURA-v1.1/valid/XR_FOREARM/patient11399/study1_positive/,1
MURA-v1.1/valid/XR_FOREARM/patient11399/study2_negative/,0
MURA-v1.1/valid/XR_FOREARM/patient11400/study1_positive/,1
MURA-v1.1/valid/XR_FOREARM/patient11248/study1_positive/,1
MURA-v1.1/valid/XR_FOREARM/patient11401/study1_positive/,1
MURA-v1.1/valid/XR_FOREARM/patient11402/study1_positive/,1
MURA-v1.1/valid/XR_FOREARM/patient11403/study1_positive/,1
MURA-v1.1/valid/XR_FOREARM/patient11404/study1_positive/,1
:
```



Injury Detection

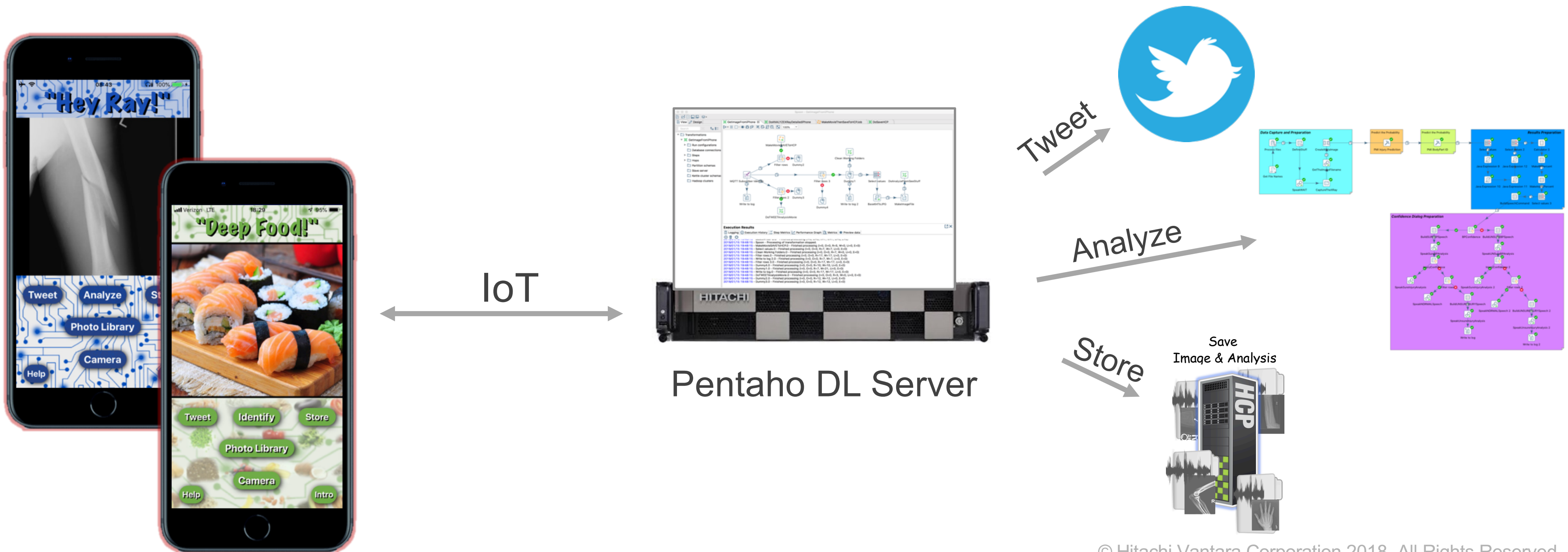


Classes:
Negative, Positive

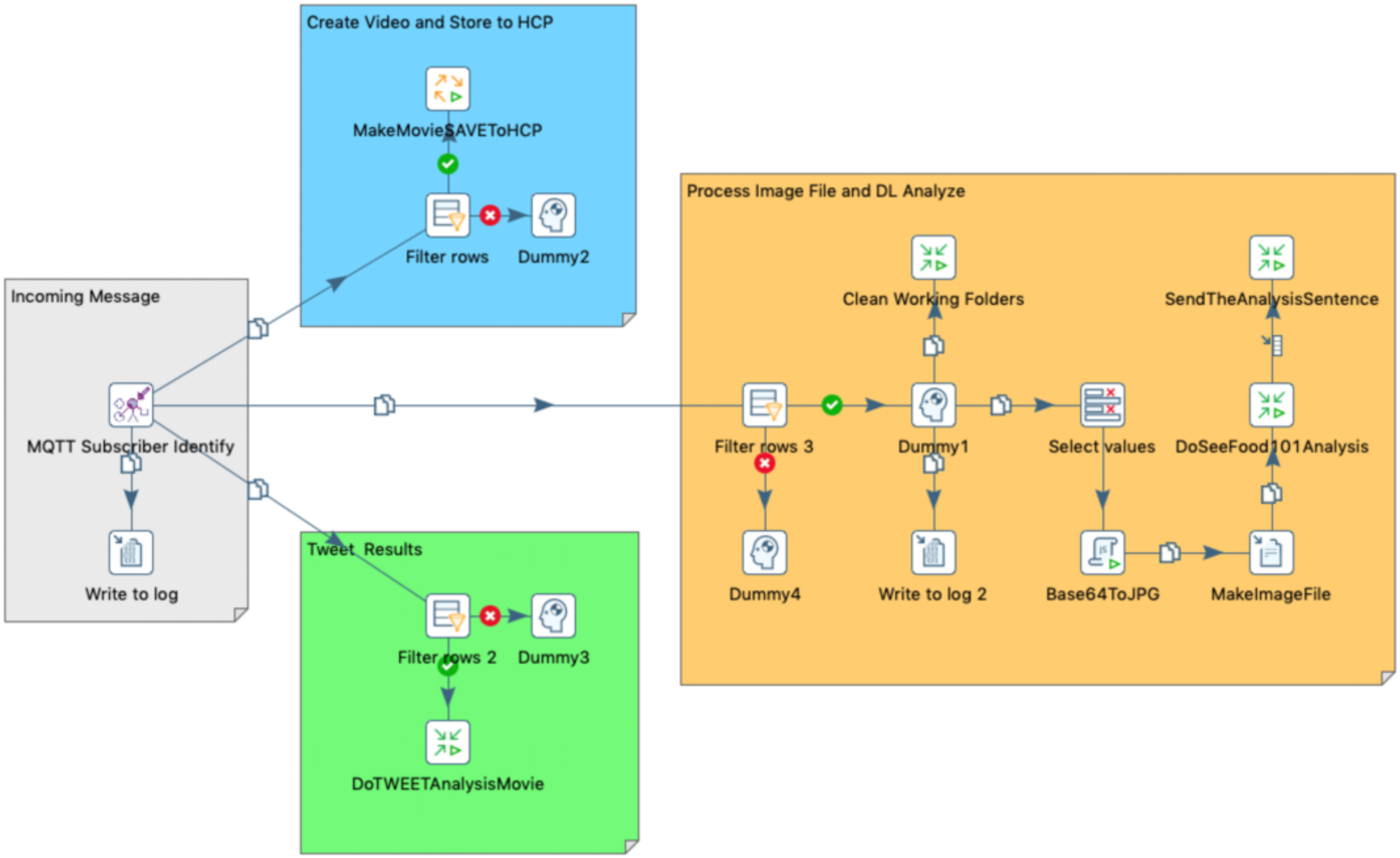
Two Deep Learning Apps Using Pentaho

HITACHI
Inspire the Next

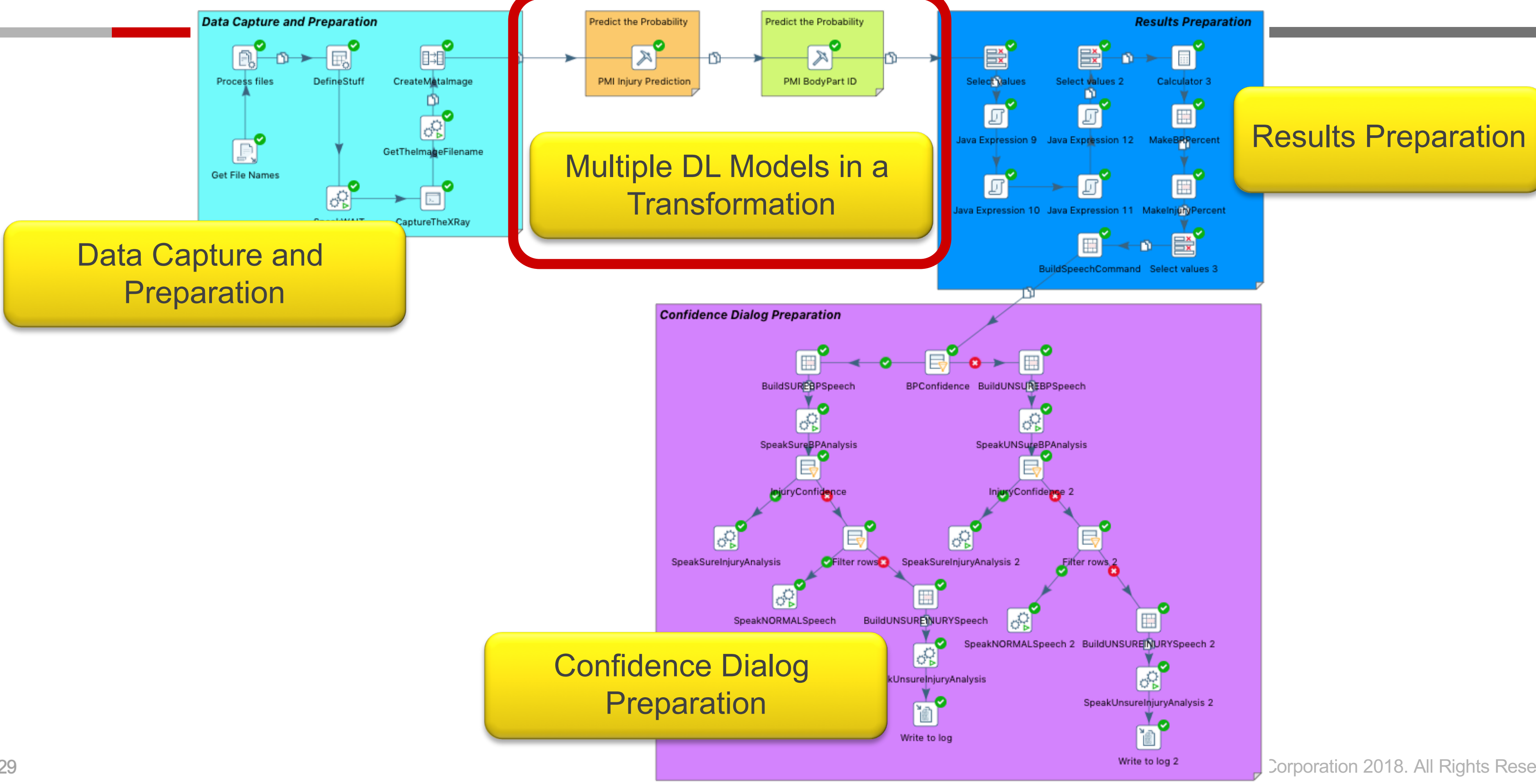
- What's the different between these two Apps?
- Primarily, the Deep Learning Model(s)



Interface to iPhone and Function Routing



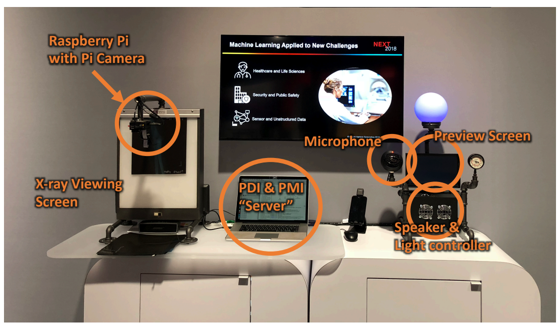
Machine Learning Pipeline – X-Ray Analysis



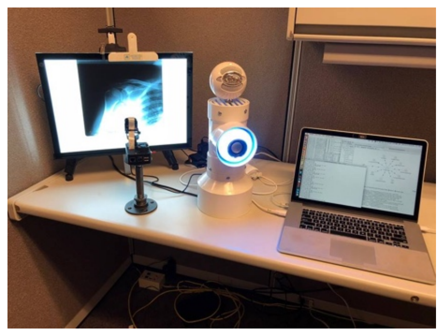
"Hey Ray!" Evolved

HITACHI
Inspire the Next

- Converting "Hey Ray!" to a smart phone with server



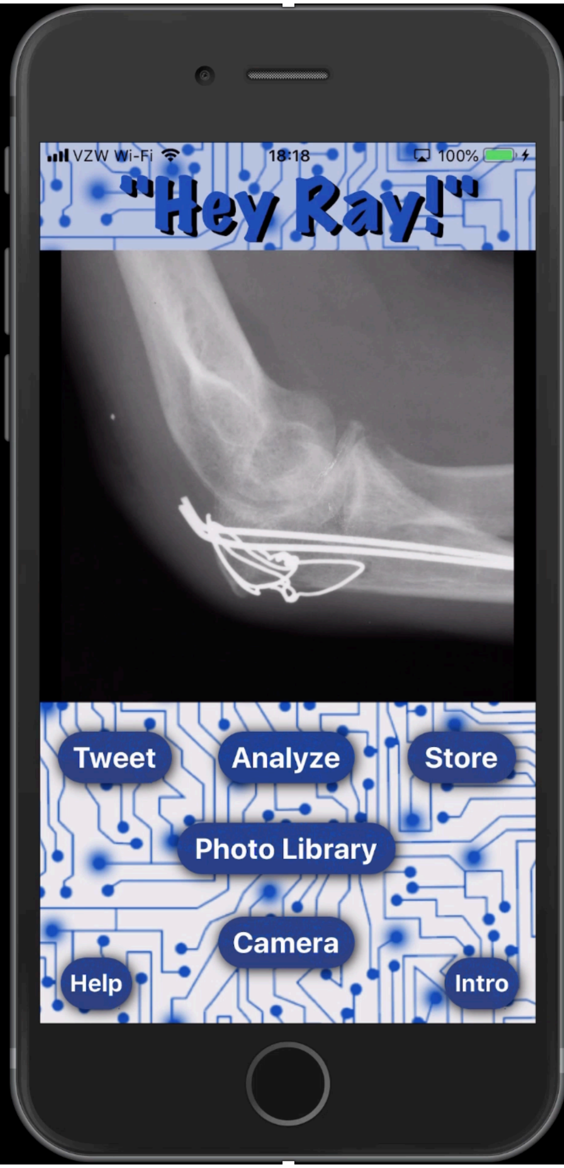
Original



First Attempt at
Compacting Configuration

Current Version

”Hey Ray!”

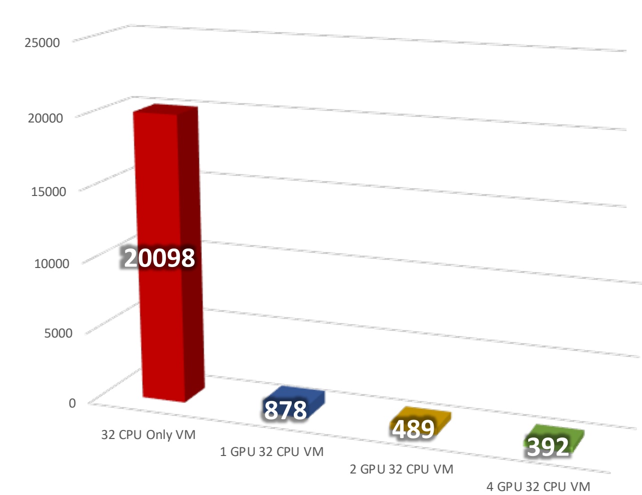

VZW Wi-Fi
18:18
100%
"Hey Ray!"
Tweet
Analyze
Store
Photo Library
Camera
Help
Intro

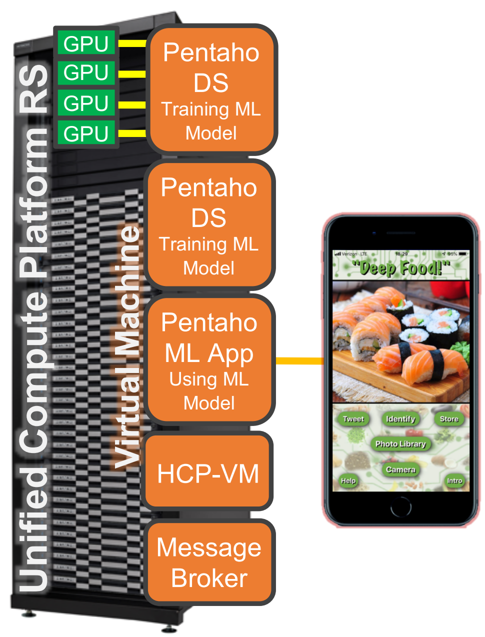
PMI with TensorFlow and Multi-GPU Support

HITACHI
Inspire the Next

- Food-101 Dataset
  - 101 food classes
  - 1000 images per class
  - 101,000 total images

Time to Create a Xception Deep Learning Model

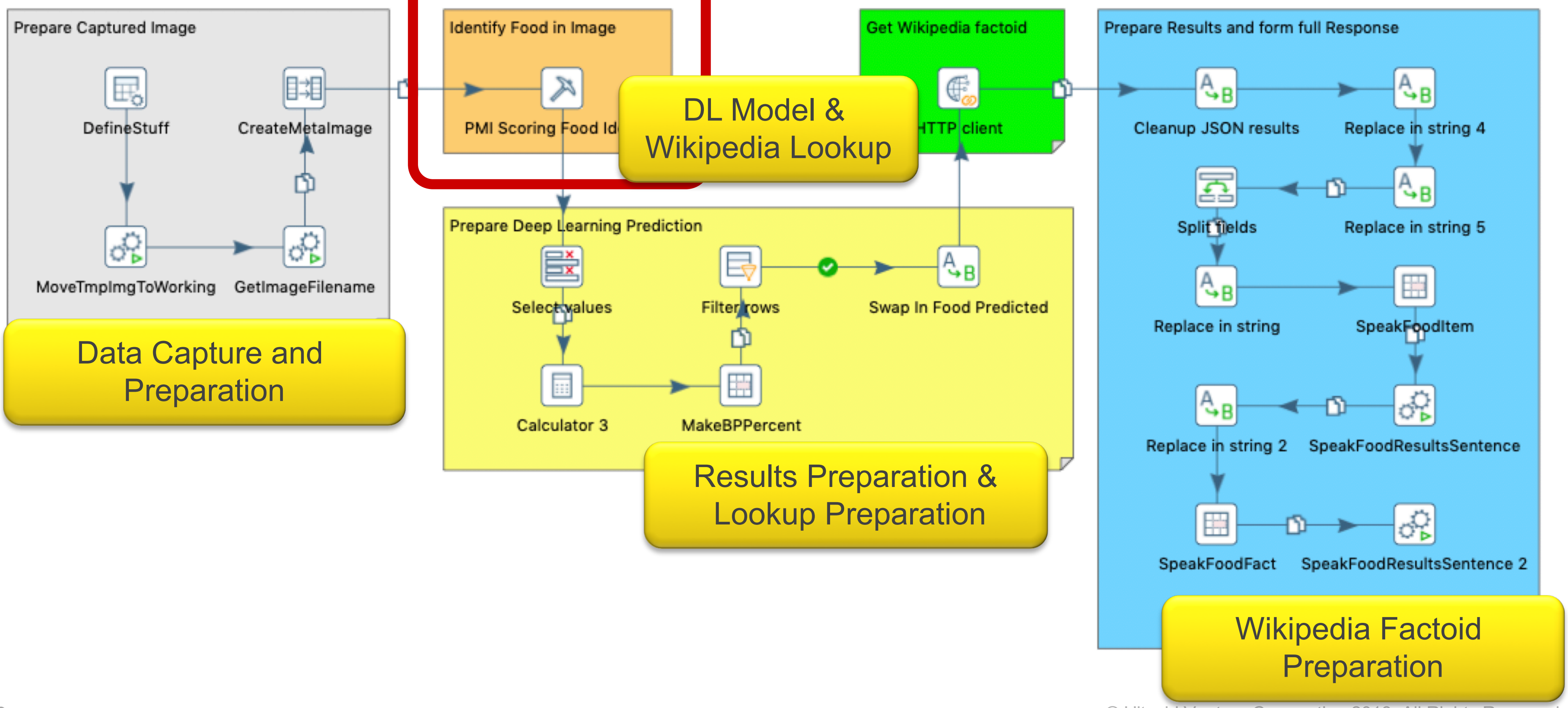




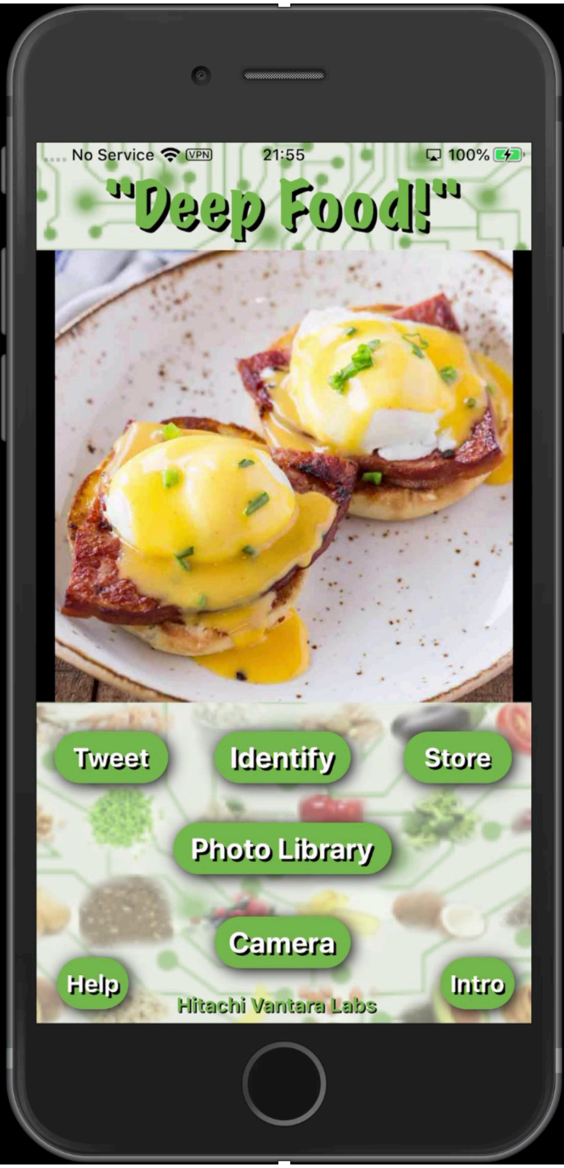
apple_pie	eggs_benedict	onion_rings
baby_back_ribs	escargots	oysters
baklava	falafel	pad_thai
beef_carpaccio	filet_mignon	paella
beef_tartare	fish_and_chips	pancakes
beet_salad	foie_gras	panna_cotta
beignets	french_fries	peking_duck
bibimbap	french_onion_soup	pho
bread_pudding	french_toast	pizza
breakfast_burrito	fried_calamari	pork_chop
bruschetta	fried_rice	poutine
caesar_salad	frozen_yogurt	prime_rib
cannoli	garlic_bread	pulled_pork_sandwich
caprese_salad	gnocchi	ramen
carrot_cake	greek_salad	ravioli
ceviche	grilled_cheese_sandwich	red_velvet_cake
cheesecake	grilled_salmon	risotto
cheese_plate	guacamole	samosa
chicken_curry	gyoza	sashimi
chicken_quesadilla	hamburger	scallops
chicken_wings	hot_and_sour_soup	seaweed_salad
chocolate_cake	hot_dog	shrimp_and_grits
chocolate_mousse	huevos_rancheros	spaghetti_bolognese
churros	hummus	spaghetti_carbonara
clam_chowder	ice_cream	spring_rolls
club_sandwich	lasagna	steak
crab_cakes	lobster_bisque	strawberry_shortcake
creme_brulee	lobster_roll_sandwich	sushi
croque_madame	macaroni_and_cheese	tacos
cup_cakes	macarons	takoyaki
deviled_eggs	miso_soup	tiramisu
donuts	mussels	tuna_tartare
dumplings	nachos	waffles
edamame	omelette	

Machine Learning Pipeline – Food Identification

HITACHI
Inspire the Next



“Deep Food!”


No Service
21:55
100%
"Deep Food!"
Tweet
Identify
Store
Photo Library
Camera
Help
Intro
Hitachi Vantara Labs

New Data Visualization

HITACHI
Inspire the Next

Examine preview data

Rows of step: CSV file input (1000 rows)

#	region-centroid-col	region-centroid-row	short-line-density-5	short-line-density-2	vedge-mean	vegde-sd	hedge-mean	hedge-sd	intensity-mean	rawred-mean	rawblue-mean	rawgreen-mean	exred-mean	exblue-mean	exgreen-n
1	218	178	0.1	0	0.8	0.5	1.1	0.5	59.6	52.4	75.2	51.2	-21.6	46.8	-2
2	113	130	0	0	0.3	0.3	0.3	0.4	0.9	0	2.6	0.1	-2.7	5	-
3	202	41	0	0	0.9	0.8	1.1	1	123	111.9	139.8	117.4	-33.4	50.2	-1
4	32	173	0	0	1.7	1.8	9	6.7	43.6	39.6	52.9	38.3	-12.1	27.9	-1
5	61	197	0	0	1.4	1.5	2.6	1.9	49.6	44.2	61.6	43	-16.1	35.9	-1
6	149	185	0	0	1.6	1.1	3.1	1.9	49.3	45.3	59.6	43.1	-12	30.7	-1
7	197	229	0	0	1.4	1.6	1.2	0.6	17.7	14.1	17.9	21.2	-10.9	0.4	1
8	29	111	0	0	0.4	0.2	0.6	0.2	5.4	6.9	6.3	3	4.4	2.8	-
9	1	81	0	0	12.2	267.5	9.2	205.4	21.3	14	30.6	19.4	-22	27.7	-
10	69	85	0.1	0	3.1	8.2	3.9	9.4	21.4	20.4	28.1	15.8	-3	20	
11	152	83	0	0	4.4	1.3	0.9	0.7	26.5	23.3	33.2	23	-9.6	20.1	-1
12	248	153	0	0	0.3	0.1	0.1	0	0.4	0	1.1	0	-1.1	2.2	-
13	137	141	0	0	0.1	0.1	0.1	0.1	0	0	0.1	0	-0.1	0.2	-
14	86	197	0.1	0.1	1.6	1.5	1.3	1.1	63.2	56.2	77.8	55.7	-21	43.7	-2
15	220	220	0.1	0	2.3	1.1	2.3	4.2	6.4	5.7	5.3	8.3	-2.3	-3.3	
16	207	115	0	0	1.1	0.3	0.2	0	1.2	0.4	2.9	0.3	-2.3	5	-
17	6	51	0	0	1.7	2	1.6	2.1	19.6	18.8	25.7	14.3	-2.4	18.2	-1
18	203	182	0	0	3.7	2.6	3.6	1.7	54.9	49.4	68.1	47.2	-16.4	39.6	-2
19	243	120	0	0	4.4	4.4	1.6	1.8	47.9	44.8	56.3	42.4	-9.2	25.4	-1
20	146	97	0	0	10.1	4.6	0.6	0.6	48	46	54.3	43.6	-5.9	19.1	-1
21	184	145	0	0	0.7	0.6	0.2	0.3	0.6	0.3	1.2	0.1	-0.7	2	-
22	178	128	0	0	0.4	0.3	0.9	0.2	5.9	7.9	6.4	3.4	5.9	1.6	-
23	132	134	0	0	2.7	3.2	1.5	1.5	6.2	2.2	11.4	4.9	-11.9	15.8	-
24	83	28	0	0	0.4	0.1	0.9	0.5	113	99.4	131.1	108.3	-40.6	54.4	-1
25	126	237	0	0	0.9	0.2	1	0.5	5.8	4.2	4.2	8.9	-4.7	-4.7	
26	225	58	0	0	0.3	0.4	0.4	0.3	8.3	5.6	14.1	5.3	-8.3	17.3	
27	14	120	0	0	0.3	0.1	0.4	0.3	1.6	0	3.9	0.9	-4.8	6.9	-
28	5	210	0	0.1	2.2	1.7	4.4	2.6	51.3	45.4	64.3	44.1	-17.6	39.1	-2
29	79	62	0	0	0.6	0.4	0.9	0.6	110.2	100.7	127.1	102.8	-28.6	50.8	-2
30	18	83	0	0.1	6.2	10	12.9	28.8	14.4	9	22.7	11.4	-16.1	24.9	-
31	214	246	0	0	2.8	1.9	3	2	17.5	14.2	15.4	22.9	-9.9	-6.2	1
32	94	140	0	0	0.2	0	0.3	0	3	1.8	6.1	1.2	-3.8	9.2	-
33	54	142	0	0	0.7	0.8	2	2.1	1.8	0.9	3.7	0.9	-2.8	5.6	-
34	107	146	0	0	1.9	1	2.1	1.3	21.4	16.8	29.7	17.7	-13.8	24.9	-1
35	93	236	0.1	0	1.8	2.7	2	0.9	12.5	9.6	10.8	17.2	-8.9	-5.2	1
36	245	249	0	0	2	1.3	1.5	0.6	14.7	10.8	15.3	18.1	-11.9	1.8	1
37	48	173	0	0	1.2	1.1	1.4	1.6	19.2	16.4	16.4	24.8	-8.3	-8.3	1
38	239	122	0	0	0.3	0.1	0.3	0.2	5.6	7	6.7	3	4.3	3.3	-
39	184	145	0	0	0.9	0.5	1.8	0.9	3.4	2.3	5.8	2.1	-3.2	7.1	-
40	109	146	0	0	0.1	0.1	0.1	0.1	0	0	0.1	0	-0.1	0.2	-
41	111	246	0	0	3.2	2.1	5.8	4.1	25.9	21.1	22.4	34.1	-14.3	-10.3	2
42	155	40	0	0	1.9	2.1	0.1	0.1	1.3	1.2	1.6	1	-0.1	0.9	-
43	192	157	0	0	1.1	0.5	0.8	0.2	18.3	13.9	17.2	23.9	-13.3	-3.3	1
44	209	249	0	0	2.4	3.6	2.8	4.2	10.5	7.6	8.1	15.9	-8.9	-7.2	1
45	118	125	0	0	0.3	0.3	0.9	0.3	1.1	0	3.1	0.3	-3.4	5.9	-
46	43	152	0	0	1.9	1.7	1.2	0.8	1.5	1	2.9	0.7	-1.6	4.1	-

Close Show Log

Thermal Imaging and Deep Learning

HITACHI
Inspire the Next