



A Software-Defined Sensor Network  
*Cyberinfrastructure for Edge Computing*  
[www.sagecontinuum.org](http://www.sagecontinuum.org)



# How to train your AI

Omar Zorob and Seongha Park



Northwestern  
University



Argonne  
NATIONAL LABORATORY



Colorado State  
University



THE UNIVERSITY OF  
CHICAGO

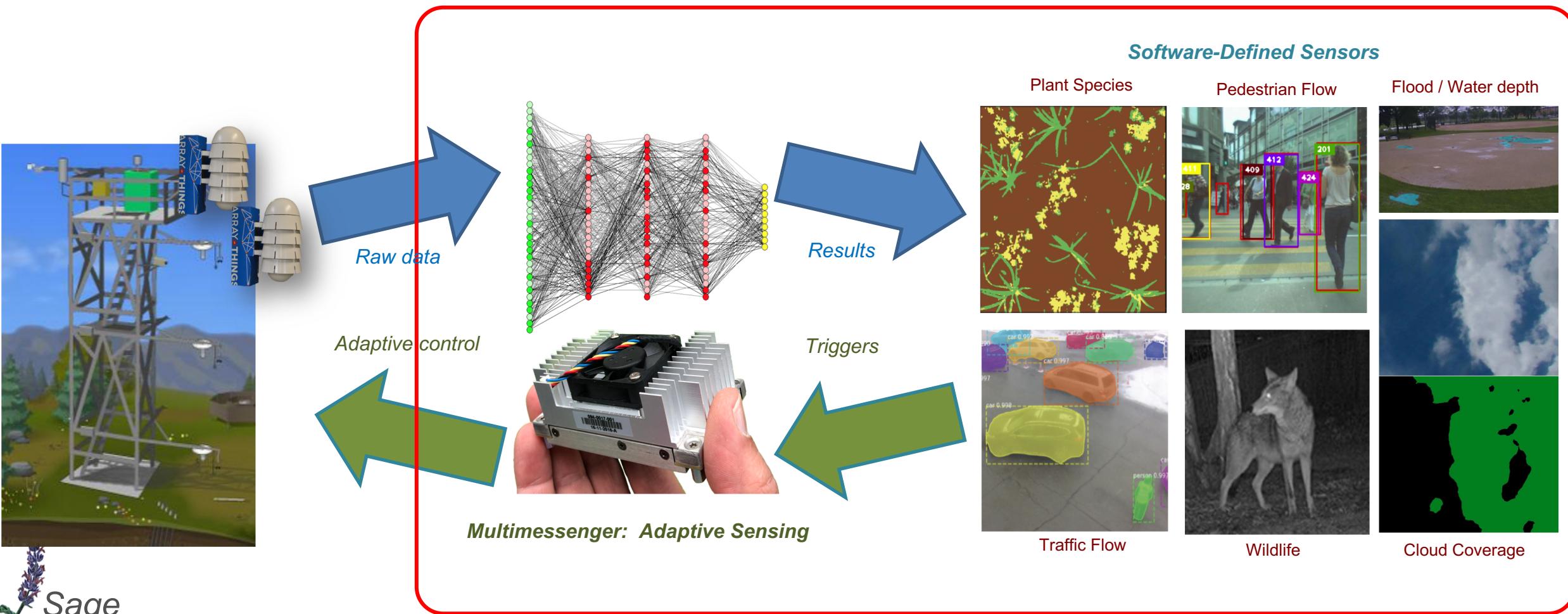


Northern Illinois  
University



# Train Your Own Model for Your Problem

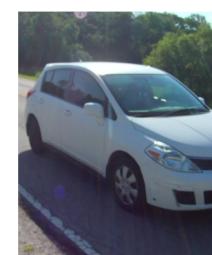
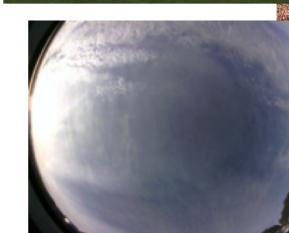
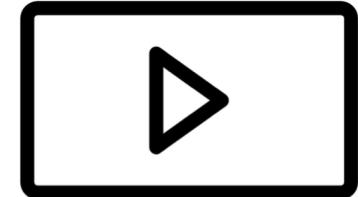
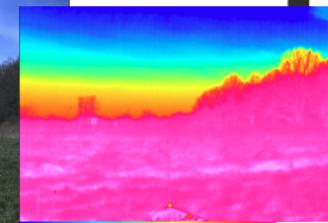
Within the Platform that SAGE is providing



# Training Datasets for Science Problems

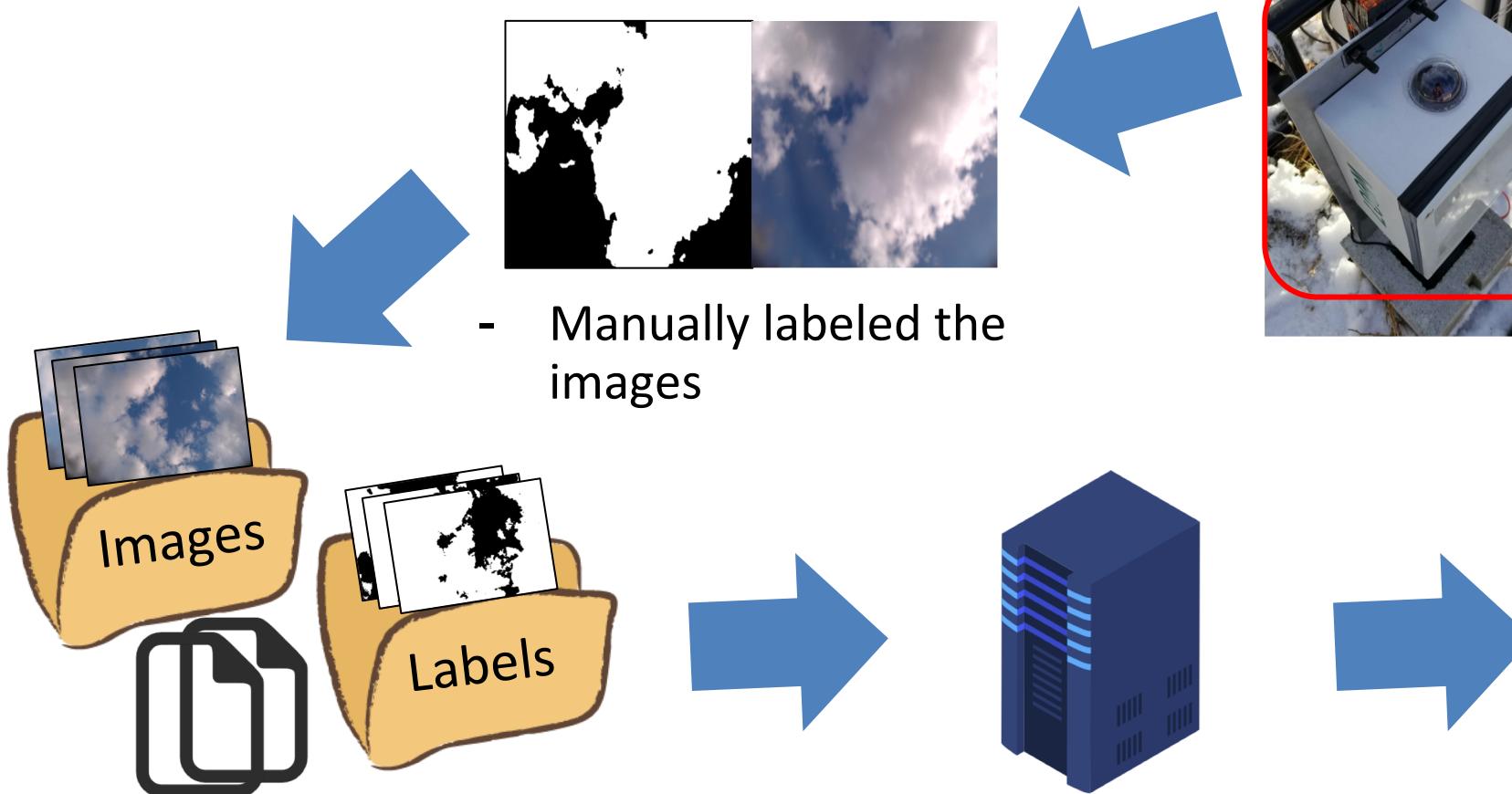
## Software-Defined Sensors

- Flooding detection
- Snow depth and coverage detection
- Smoke and Fire detection
- Cloud coverage determination
- Rime ice detection
- Pedestrian and traffic tracking
- Pedestrian and traffic flow and safety detection
- Urban heat island detection
- Photovoltaics (PV) performance prediction
- And other science problems in real-time



# Example Dataset

## Cloud segmentation dataset

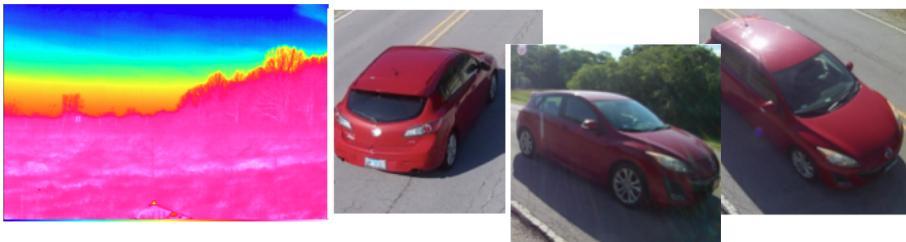


- Images collected from a node at ANL
- Red box: sky facing camera
- Orange box: Solar irradiance measuring sensor

# Labeling

## Auto labeling

- When the target is easily distinguishable by color or other factors
- Or can be labeled based on the same target from different perspectives
  - e.g. vehicle labeling from multi-perspective images



## AI assisted labeling

- fine grained labeling
  - e.g. car make and model
  - auto detect vehicles, manual label make and model

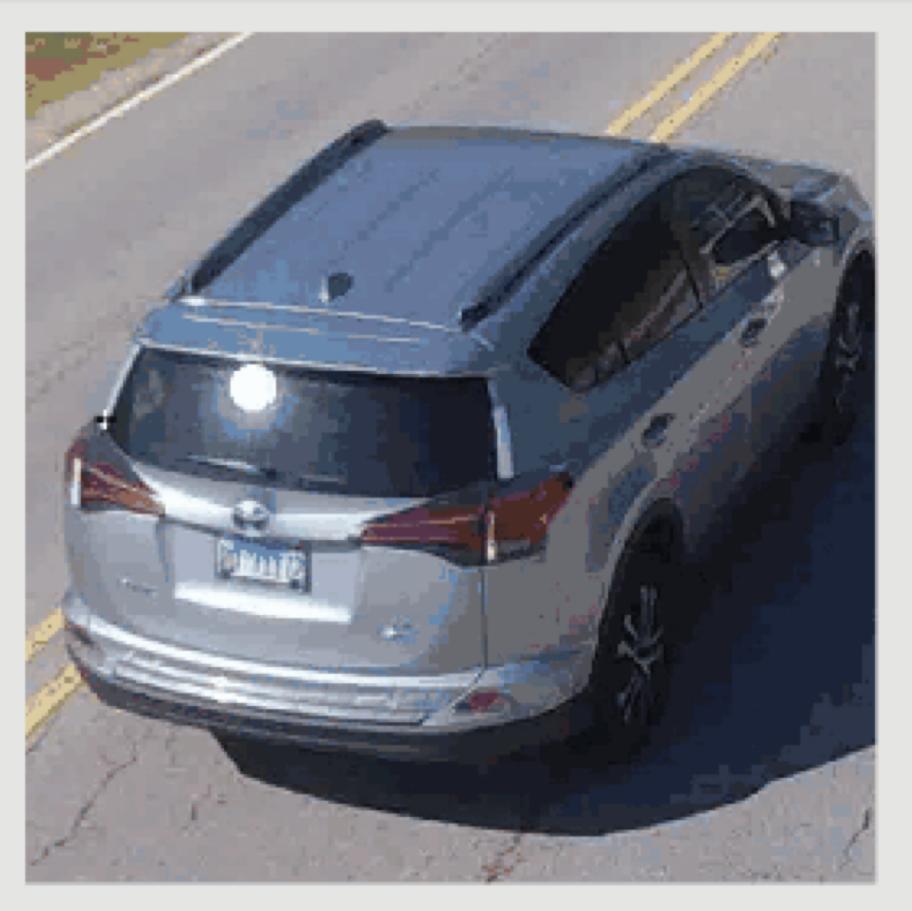


## Manual labeling

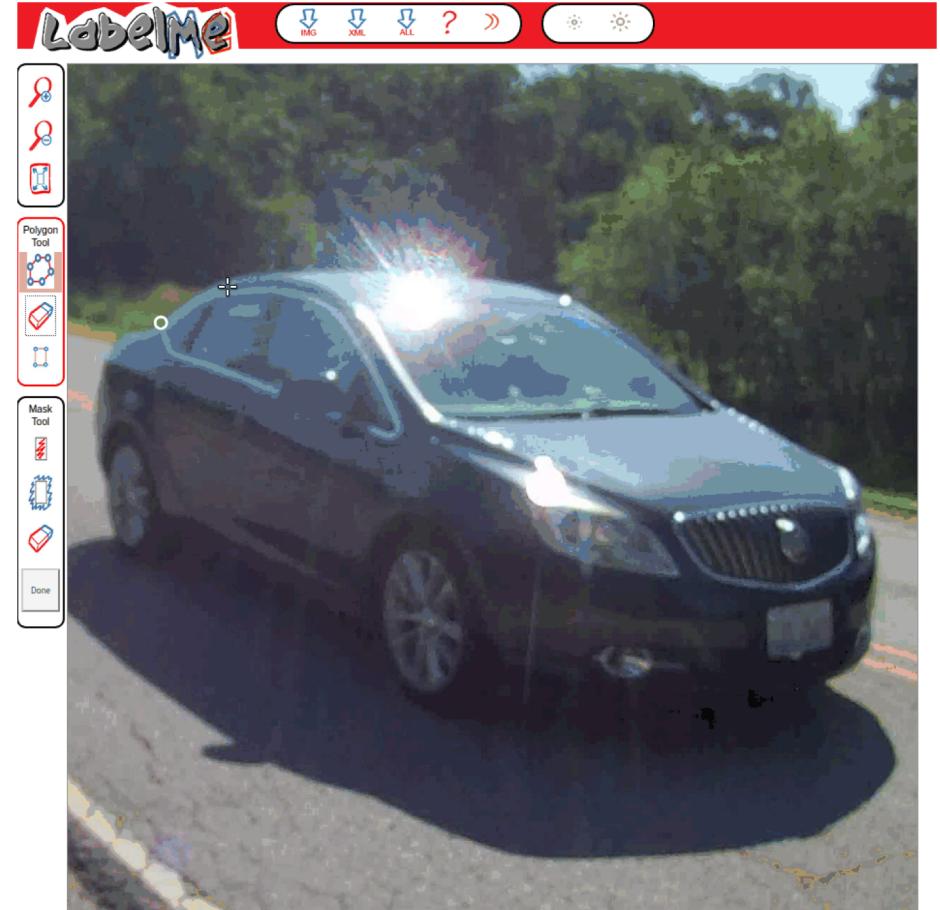
- Most of other cases
  - e.g. water depth and coverage, cloud depth and coverage, etc.



# Manual Labeling Tools

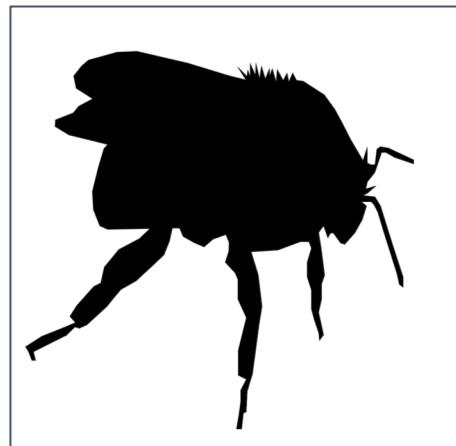
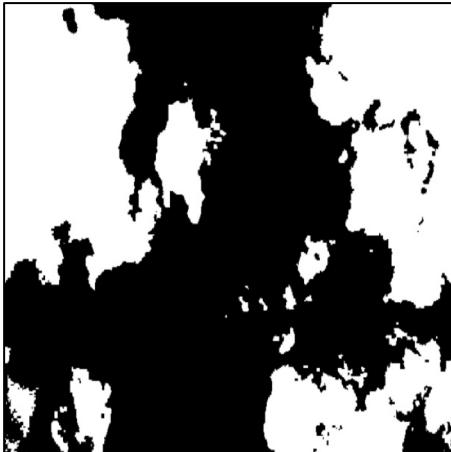
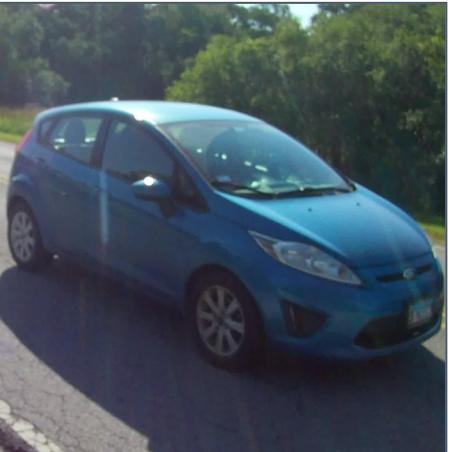
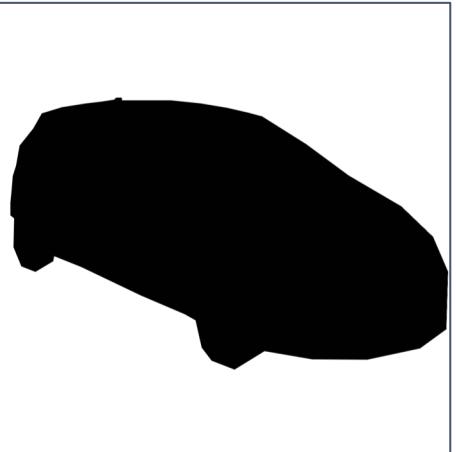


[LabelBox](#)



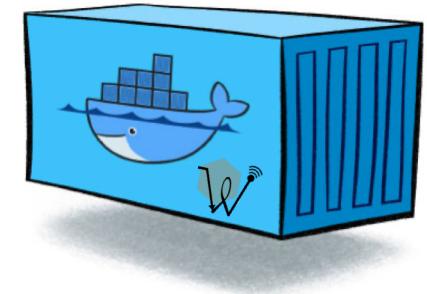
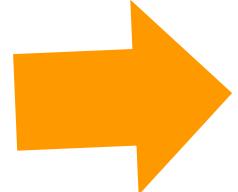
[LabelMe](#)

# Labeling Examples



# Cloud Training Software Stack (CTSS)

To provide users software environment for training



Servers or other resource containing machine

- Dataset, DL network, pre-trained model, and other materials

A training computing environment

- communicate with servers or other resource containing machine
- provide training and monitoring environment
- Access to virtual Waggle for pre-field test of the model

Virtual Waggle

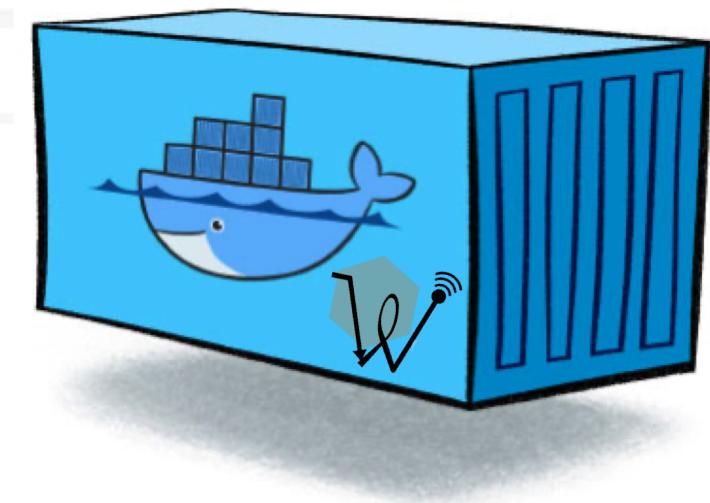
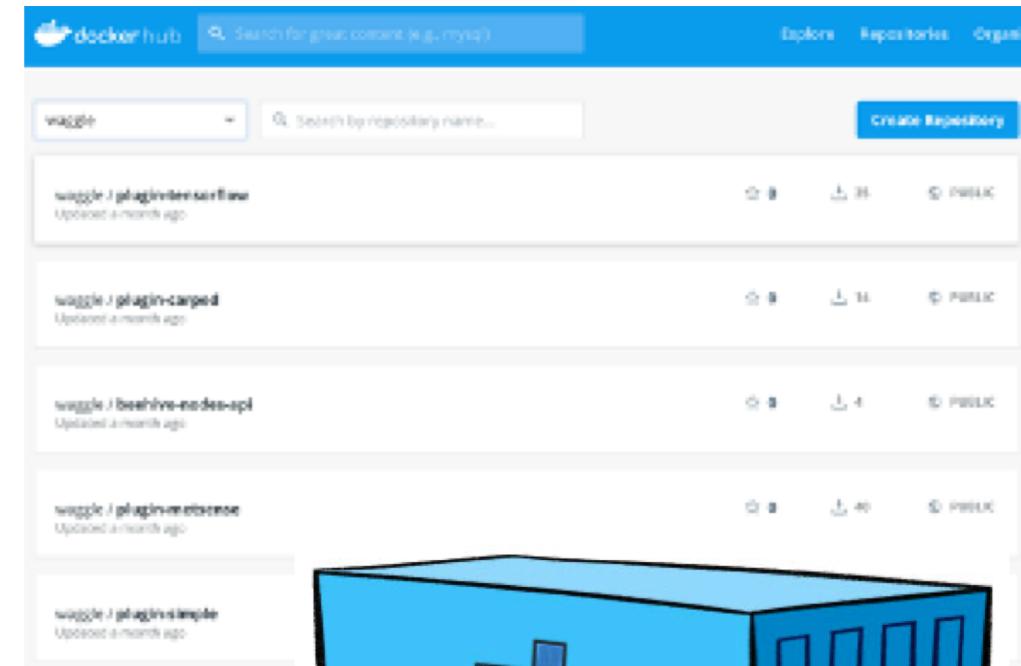
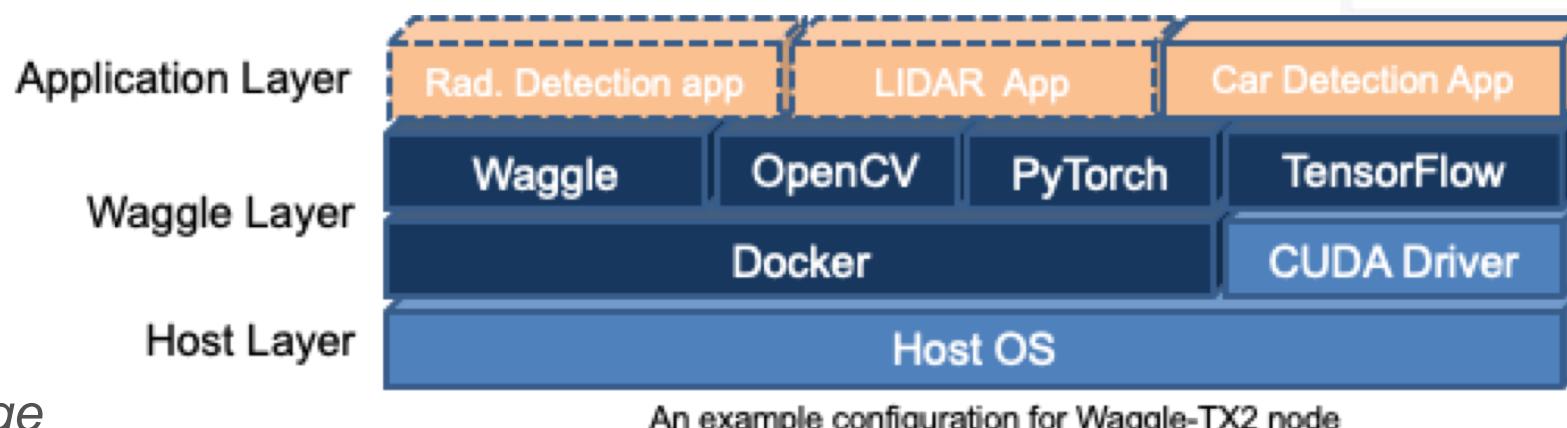
- Pre-field test

# SAGE Docker Base Images

Guarantee compatibility and support applications

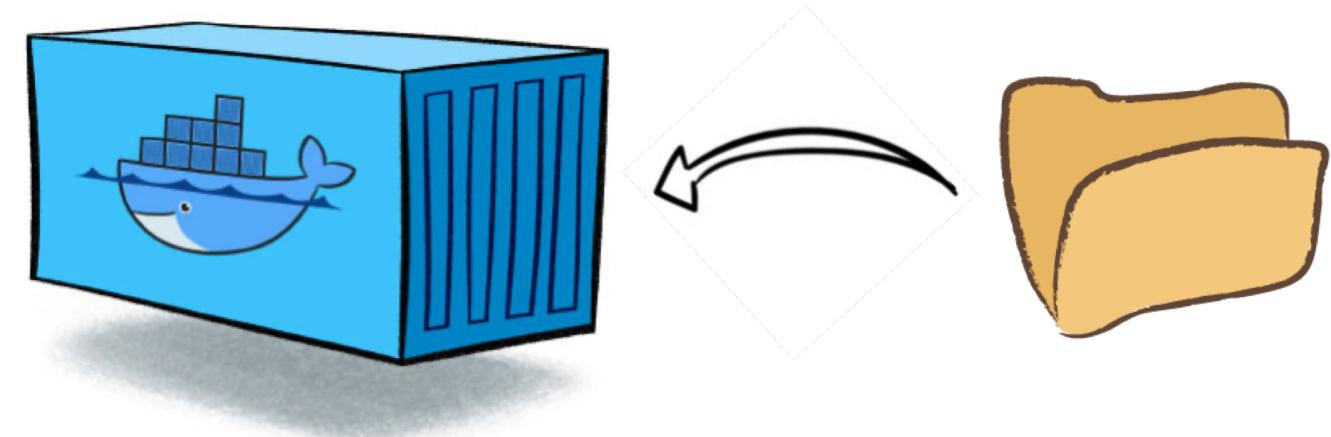
What SAGE Docker containers bring to you:

- Waggle Base Image
- Libraries for the machine learning tools (Python, PyTorch, Tensorflow, etc.)
- CUDA 10.X supported



# How to Run the Docker Container

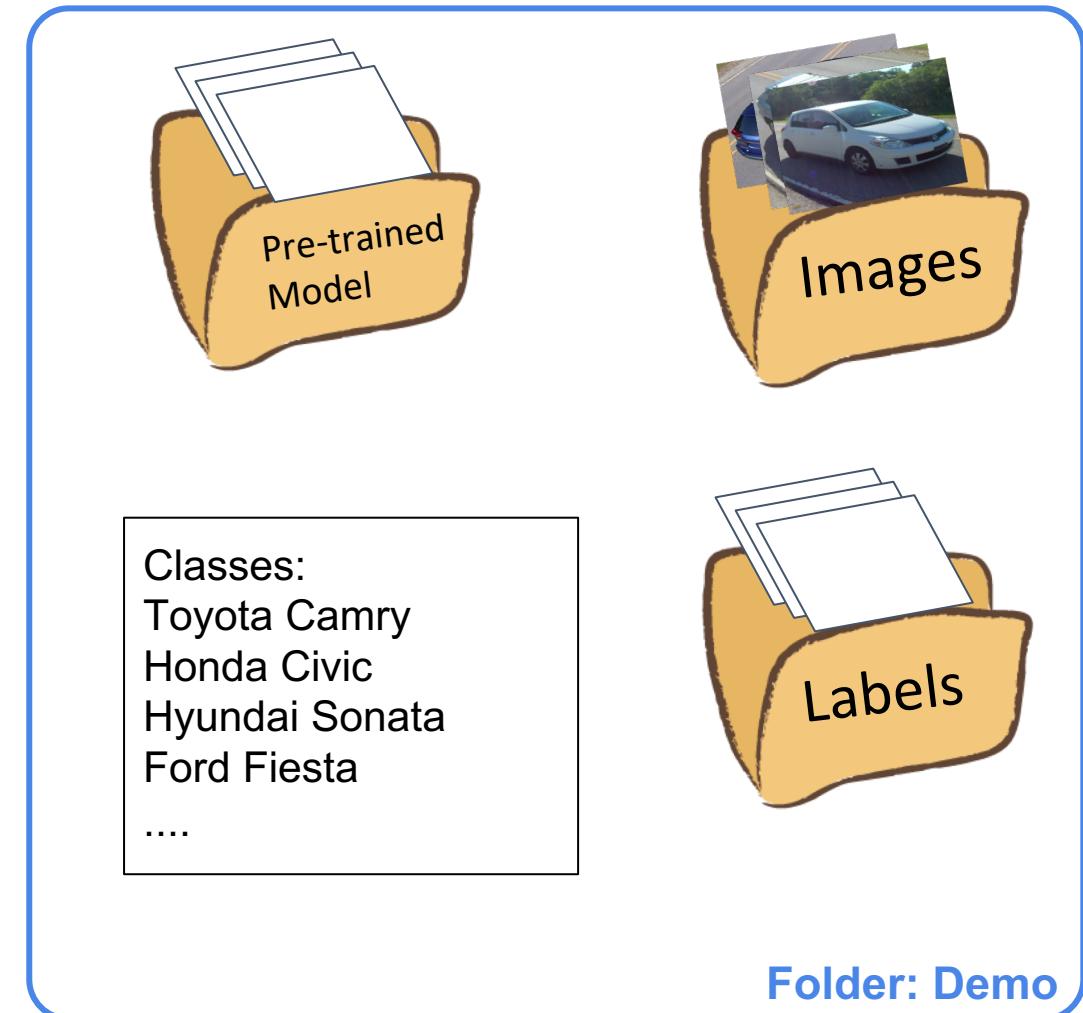
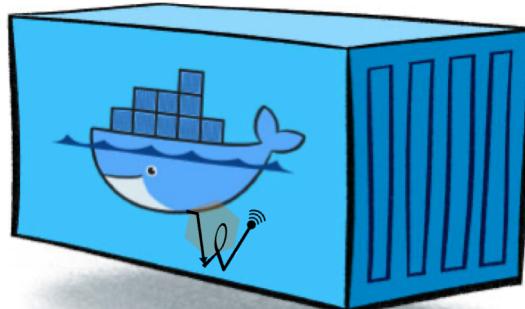
- The SAGE Docker images for training a deep learning model and inference using the model are designed to execute the training or inference module right away when they are called
- Call the container with the folder where the training images, labels, training configuration, and class name file locate
  - To let the container access files in the folder
- What users bring to run the containers
  - Training images and labels
  - Training configuration
  - Class names



# Example: car make and model identification

- PyTorch 1.3
  - including Torchvision
- CUDA 10.1
- python libs:
  - numpy
  - pillow
  - tqdm
  - terminaltables

waggle/plugin-training-yolov3

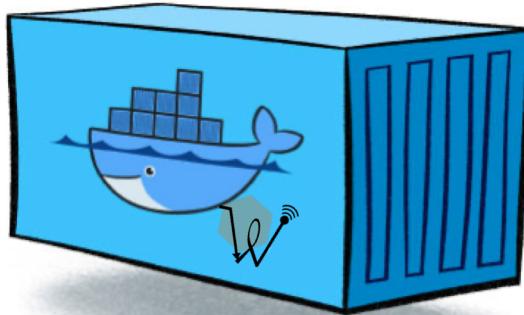


# Example: cloud segmentation

Training + Test (inference)

```
waggle@edge-dev:/demo$ ls
class_names.list config.list images labels
[waggle@edge-dev:/demo$ docker run -d --rm --name plugin-training-fcn --runtime nvidia
a --shm-size 16G -v /demo:/storage waggle/plugin-training-fcn --config config.list -
-image_type waggle_cloud
361b0815bb99e36e1709814c8067ff4210e402ee005da691ac5a9299712f89bd
[waggle@edge-dev:/demo$ docker logs -f plugin-training-fcn
Downloading: "https://download.pytorch.org/models/resnet101-5d3b4d8f.pth" to /root/.cache/torch/checkpoints/resnet101-5d3b4d8f.pth
100%[=====]| 170M/170M [00:02<00:00, 85.9MB/s]
Train:  0% | 0/163 [00:00<?, ?it/s]
^Cain epoch=0:  3%|9 | 21/615 [00:38<04:42,  2.11it/s]
[waggle@edge-dev:/demo$ docker logs -f plugin-training-fcn
Downloading: "https://download.pytorch.org/models/resnet101-5d3b4d8f.pth" to /root/.cache/torch/checkpoints/resnet101-5d3b4d8f.pth
100%[=====]| 170M/170M [00:02<00:00, 85.9MB/s]
Train:  0% | 0/163 [00:00<?, ?it/s]
Train epoch=0:  6%#5 | 36/615 [00:45<04:21,  2.22it/s]
```

waggle/plugin-training-fcn



Configuration:

```
{
    "max_iteration": 100000,
    "lr": 1e-10,
    "momentum": 0.99,
    "weight_decay": 0.0005,
    "interval_validate": 4000,
    "batch_size": 8,
    "backbone": "resnet",
    "fcn": "101",
    "output_dir": "resnet101",
    "pretrained_net": "",
    "n_workers": 6
}
```



Classes:  
sky  
cloud

Folder: Demo

# Output of the Trainings

- Checkpoints (.pth)
- Last checkpoint (Final model, .pth)
- Training logs (Tensorboard, CSV file)
  - Training accuracy, training loss, validation accuracy, validation loss, etc.
- Training configuration

## To Train or Deploy Your Model

- Create a plugin by utilizing Docker container to perform target detection, segmentation, or other work with your model
- Upload the relative sources on SAGE edge plugin pipeline

# Cyber Infrastructure for Users

To train your AI

- Develop and deploy SAGE nodes for collecting data
- Create datasets for users
- Provide methods to access the dataset easily
- Support students and scientist to use our training environment easily through command line or/and web UI



Thank you

