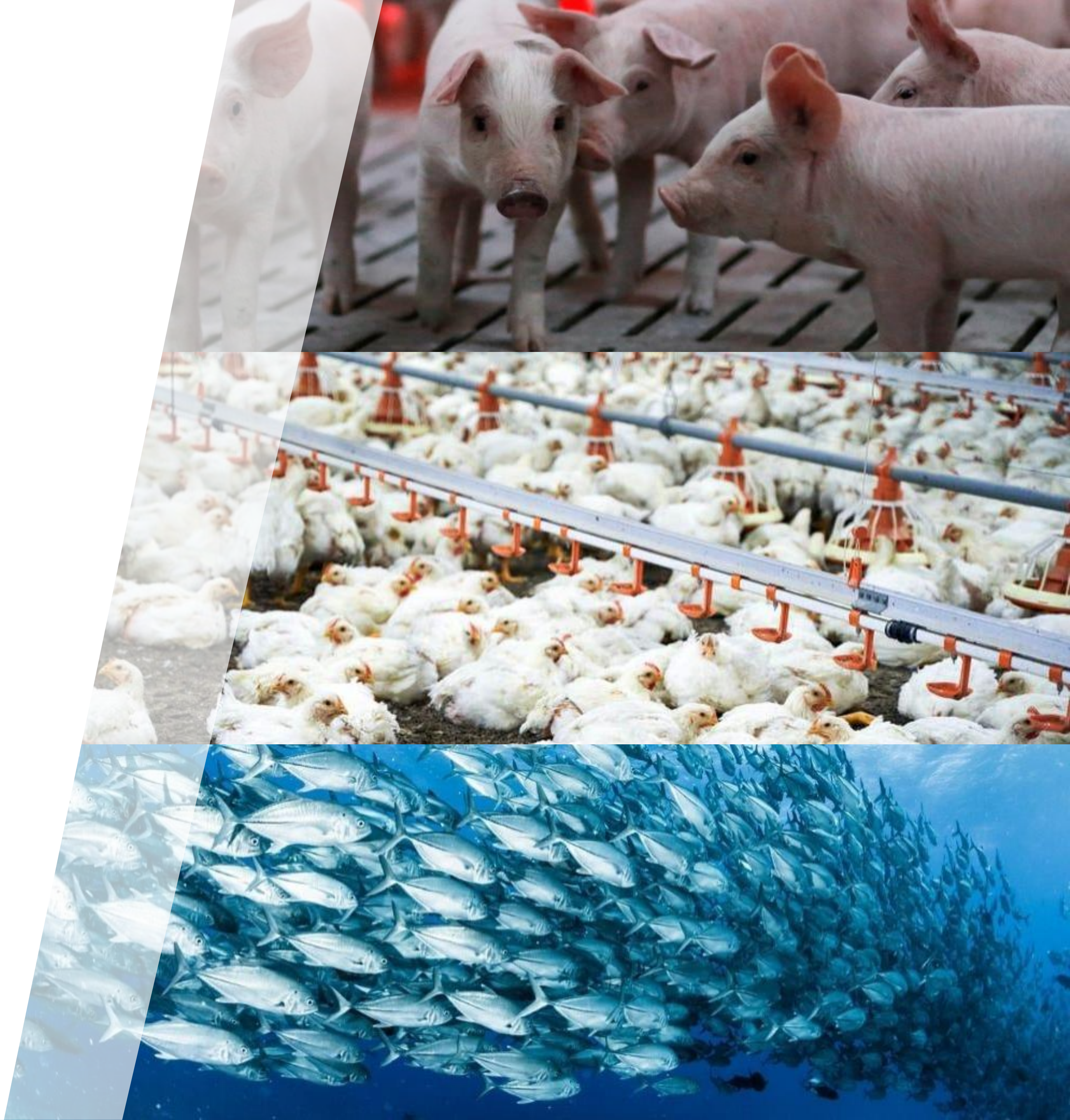
National Taiwan University
Biomechatronics Engineering

National Taiwan University X Council of Agriculture

# 機器視覺技術
# 以智慧農業為例

Yan-Fu Kuo | 27, June 2022

數位學堂

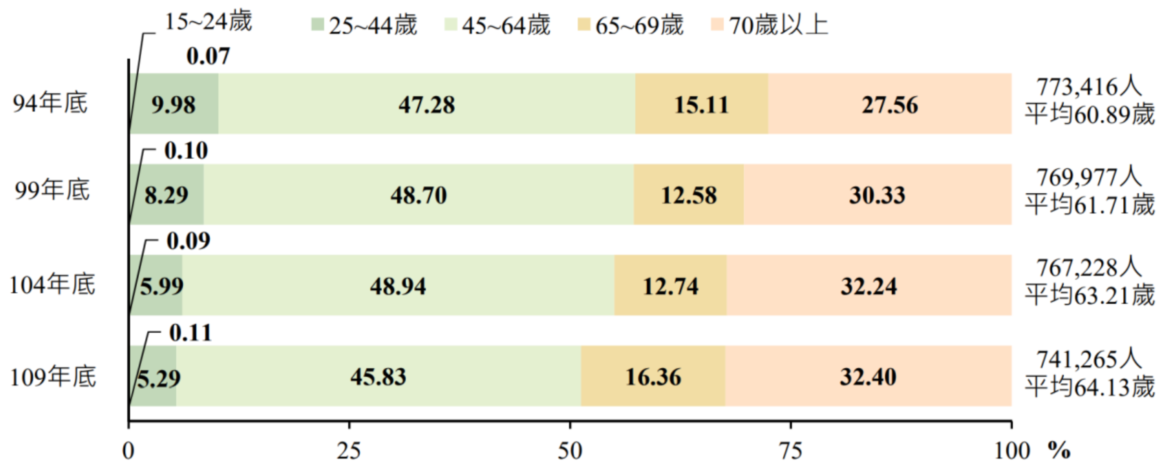


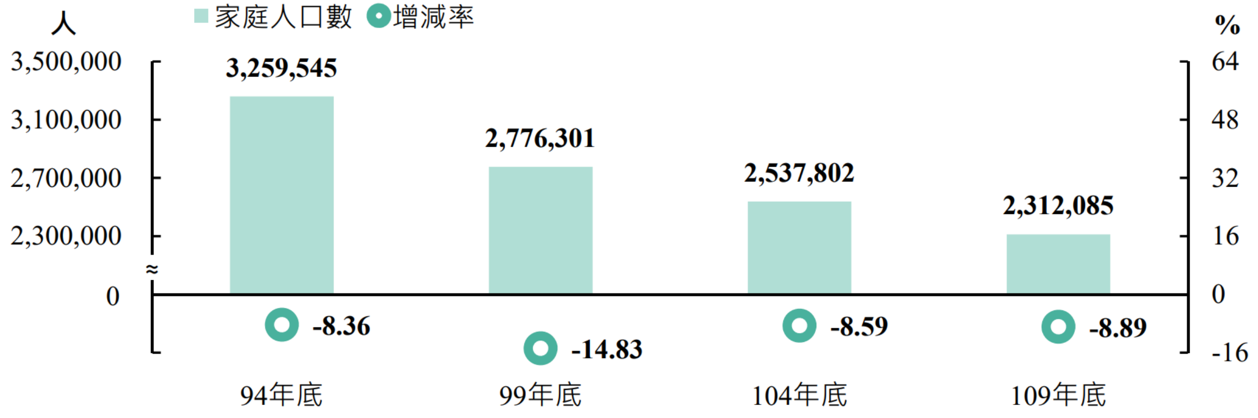
# Food Security

# Traditional Farming

# Why Machine Vision?

- Labor shortage
- Aging of farmers



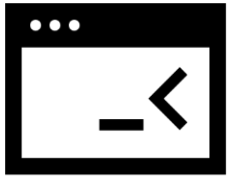




109年農林漁牧業普查初步統計結果提要分析

# Smart Machine Vision

- Optical sensors (e.g., cameras)
- Capturing images of objects
- Calculating and processing the information in the images (e.g., deep learning)
- Monitoring, warning, or taking action using the information






Yan-Fu Kuo
Dept. of Biomechatronics Engineering

# Machine Vision Application


https://www.assemblymag.com/ext/resources/White_Papers/Sep16/Introduction-to-Machine-Vision.pdf
https://medium.com/vsinghbisen/application-of-computer-vision-in-precision-agriculture-farming-79b0600d5a5d

Yan-Fu Kuo
Dept. of Biomechatronics Engineering

# What Smart Machine Vision Can Do?

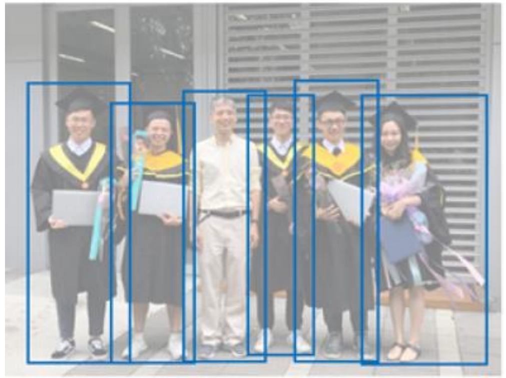
1 **Classification**

Cardboard cut-out

Human

| |
|---|
| AlexNet |
| VGG-16 |
| ResNet-55 |
| EfficientNet |

2 **Localization and Classification**

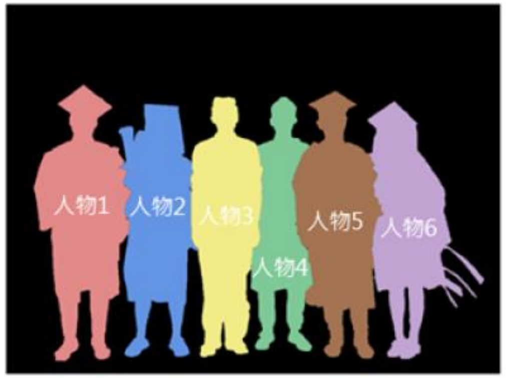


| |
|---|
| Fast R-CNN |
| Faster R-CNN |
| YOLO v4 |
| YOLO v5 |

3 **Semantic segmentation**


| |
|---|
| U-Net |
| FCN |
| DeepLabv3+ |

4 **Instance segmentation**



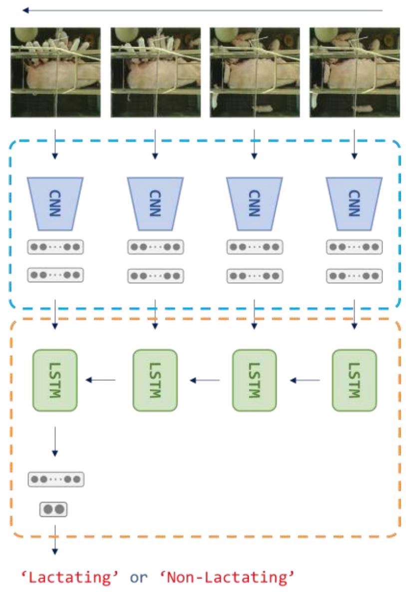

| |
|---|
| Mask R-CNN |
| MaskLab |
| YOLACT |

Yan-Fu Kuo
Dept. of Biomechatronics Engineering

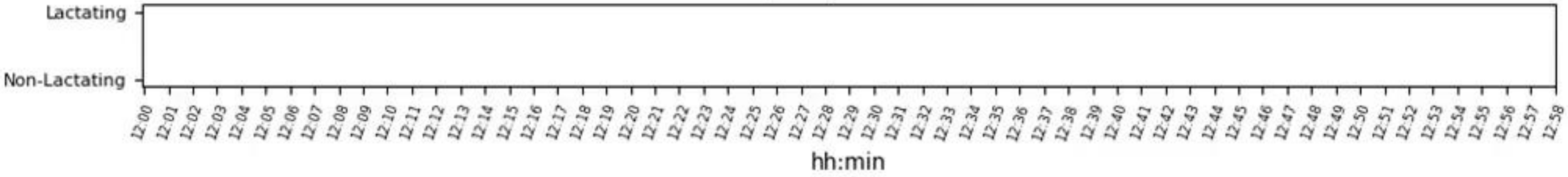
# What Smart Machine Vision Can Do?

5 **Behavior Recognition**

Input video




Ouput prediction



# Implementation Flow of Machine Vision

## Image acquisition

Acquire images manually by human or automatically by installed cameras and image transmission equipments.

## Database

Store images with labels and feature locations, and store other useful information such as temperature.

## User access

Establish microservice and serve as a bridge between the model and public by docker, kubernetes...

## Machine vision

**Dataset**

Split images into training, validation, and testing dataset.

**Image augmentation**

Increase image variety to improve model robustness.

**Model architecture**

Utilize suitable machine learning models to achieve the target.

**Model training**

Use training dataset and choose  suitable optimizer to update parameters and use validation dataset to validate.

**Model evaluation**

Test model performance by testing dataset.

**Framework**

Select suitable framework for developing models

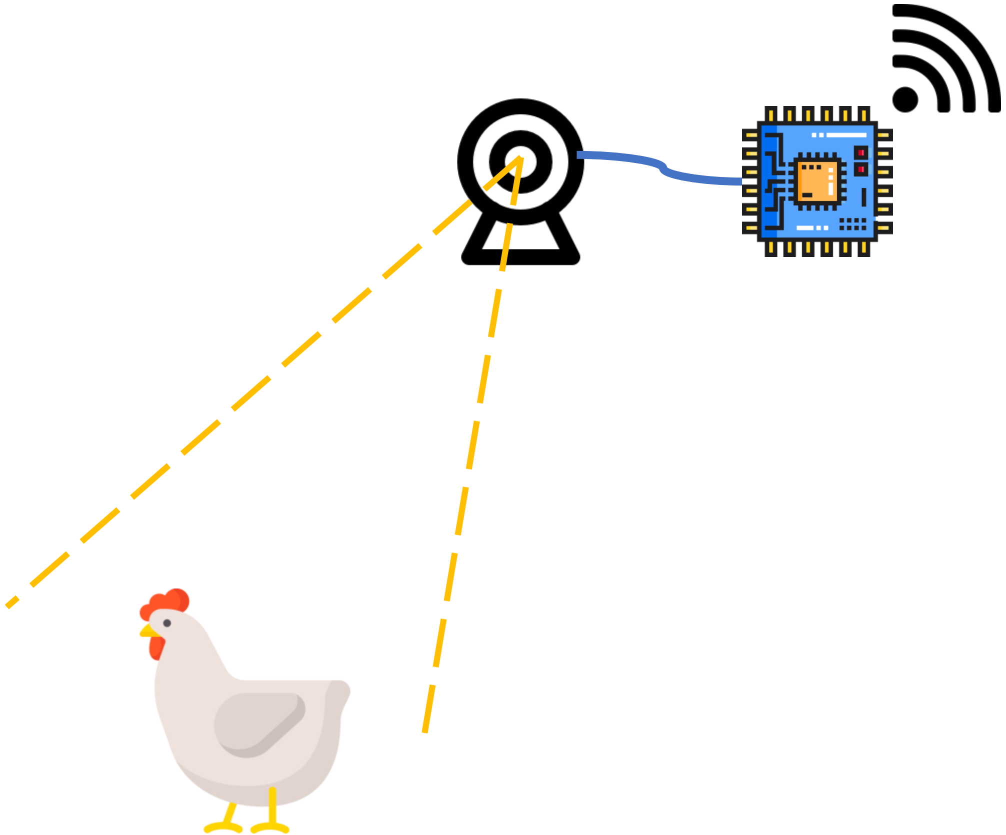
# Image Acquisition

## Image acquisition

Acquire images manually by human or automatically by installed cameras and image transmission equipments.

# Implementation Situations

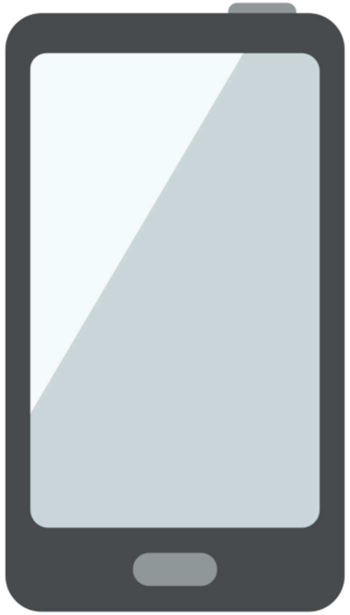
One-time



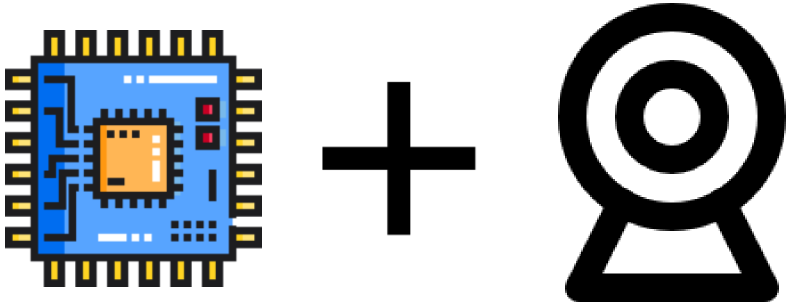
Continuous

Yan-Fu Kuo
Dept. of Biomechatronics Engineering

# Image Acquisition



Cellphone



Embedded system/ ip camera

# Implementation Flow of Machine Vision

## Image acquisition

Acquire images manually by human or automatically by installed cameras and image transmission equipments.

## Database

Store images with labels and feature locations, and store other useful information such as temperature.

## User access

Establish microservice and serve as a bridge between the model and public by docker, kubernetes...

## Machine vision

**Dataset**
Split images into training, validation, and testing dataset.

**Image augmentation**
Increase image variety to improve model robustness.

**Model architecture**
Utilize suitable machine learning models to achieve the target.

**Model training**
Use training dataset and choose suitable optimizer to update parameters and use validation dataset to validate.

**Model evaluation**
Test model performance by testing dataset.

**Framework**
Select suitable framework for developing models



Yan-Fu Kuo
Dept. of Biomechatronics Engineering

# Machine Vision

## Machine vision

**Dataset**

Split images into training, validation, and testing dataset.

**Image augmentation**

Increase image variety to improve model robustness.

**Model architecture**

Utilize suitable machine learning models to achieve the target.

**Model training**

Use training dataset and choose suitable optimizer to update parameters and use validation dataset to validate.

**Model evaluation**

Test model performance by testing dataset.

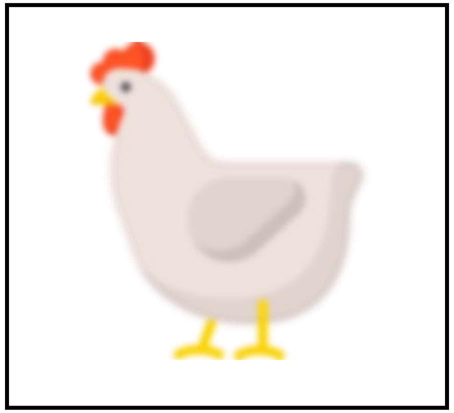
**Framework**

Select suitable framework for developing models

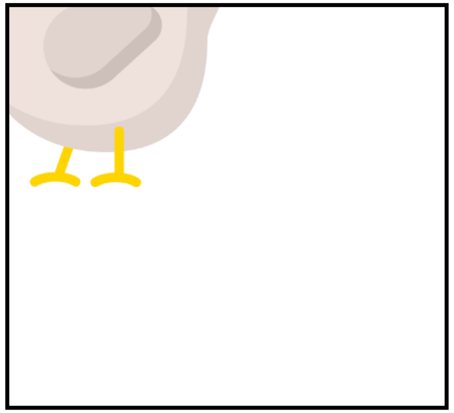
# Procedure – Training A Deep Learning Model

- Step 1 | Preparing at least 500 images for each category
- Step 2 | Generalizing the images using augmentation
- Step 3 | Choosing a framework
- Step 4 | Choosing a suitable model architecture
- Step 5 | Training the model
- Step 6 | Evaluating the model performance

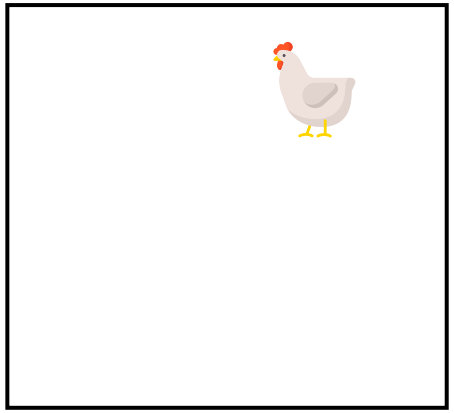
# Image Collection



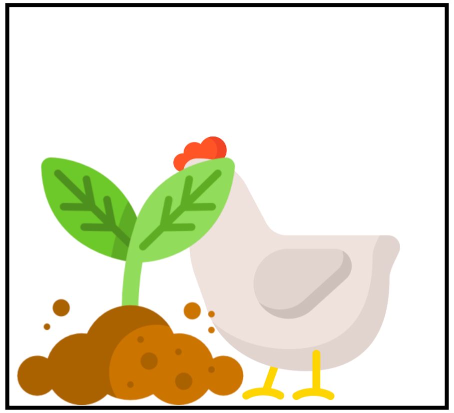
Blur



Missing object

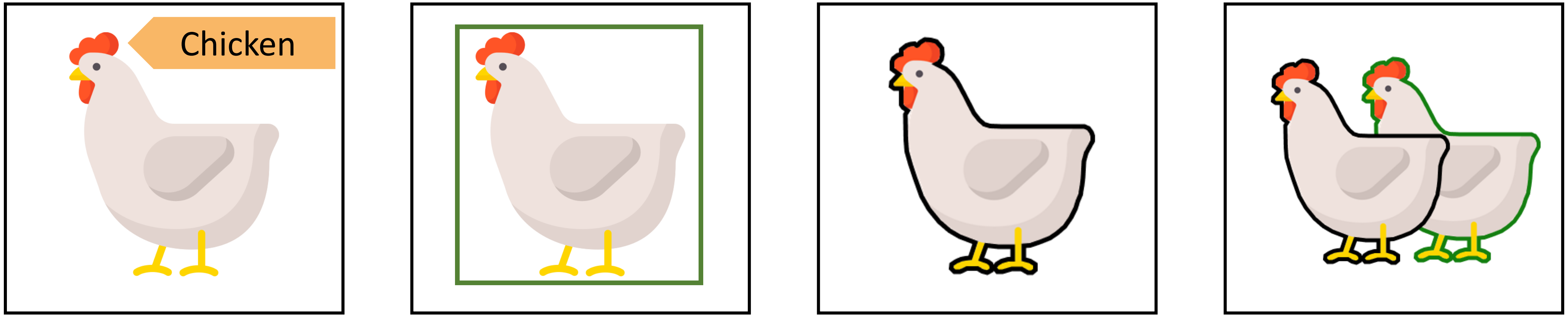


Too small



Occluded

# Image Annotation




Classification

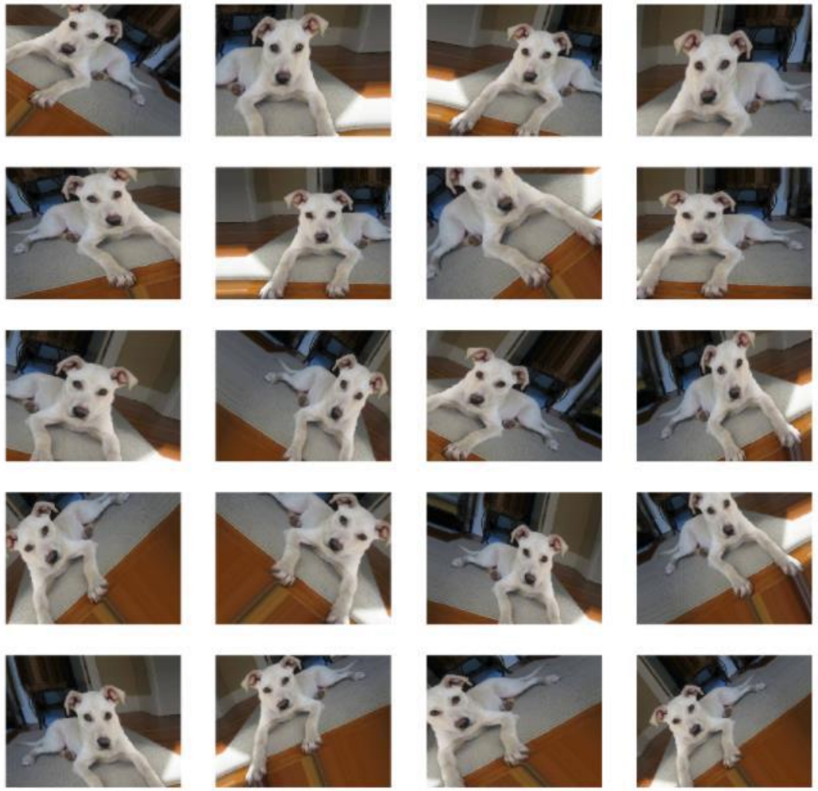
Localization and classification

Semantic segmentation

Instance segmentation

# Image Augmentation

- Adjusting existing training images to generalize to other situations
- Allowing the model to learn from a wider array of situations

# Smart Machine Vision Tasks – Static

**Classification**

Cardboard cut-out

Human

AlexNet

VGG-16

ResNet-55

EfficientNet

**Localization and Classification**



Fast R-CNN

Faster R-CNN

YOLO v4

YOLO v5

**Semantic segmentation**




U-Net

FCN

DeepLabv3+

**Instance segmentation**

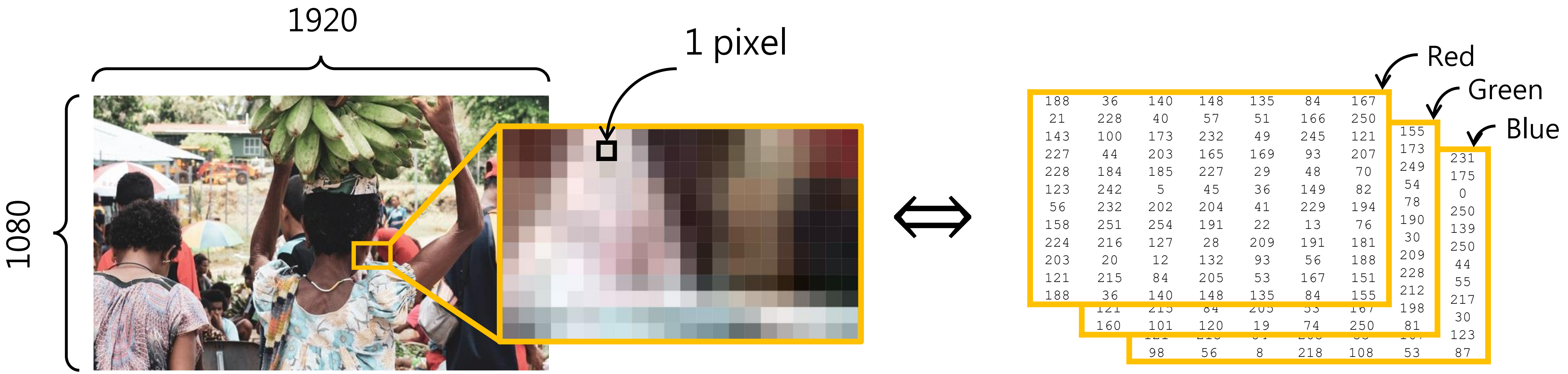



Mask R-CNN

MaskLab

YOLACT

# What Is An Image?




What humans see

What machines see

Yan-Fu Kuo
Dept. of Biomechatronics Engineering

# Classification Using A Convolutional Neural Network

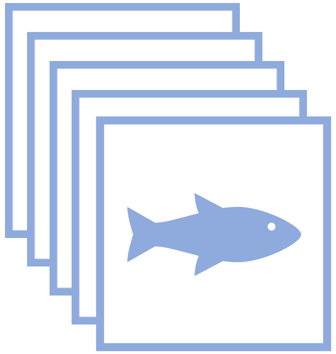
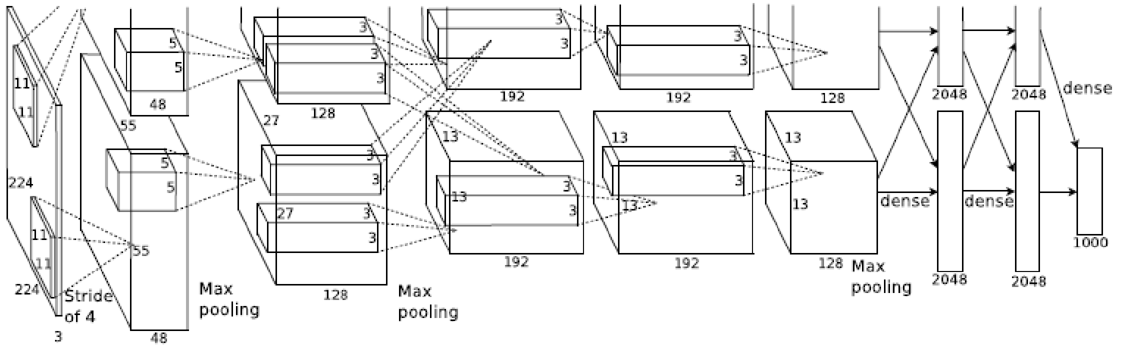


Image Collection

Convolutional Neural Network (CNN)




Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. Advances in neural information processing systems, 25.

Model Training

TUNA

SHARK

WHALE

Prediction

# Smart Machine Vision Tasks – Static

**Classification**

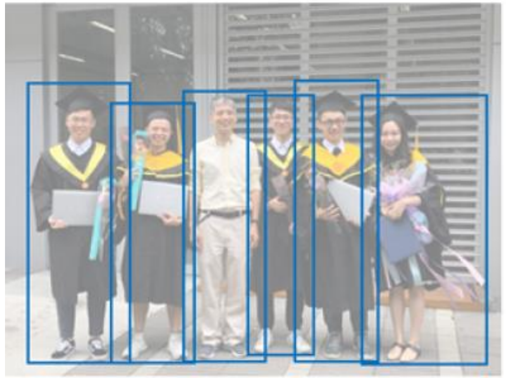


Cardboard cut-out

Human

| |
|---|
| AlexNet |
| VGG-16 |
| ResNet-55 |
| EfficientNet |

**Localization and Classification**



| |
|---|
| Fast R-CNN |
| Faster R-CNN |
| YOLO v4 |
| YOLO v5 |

**Semantic segmentation**



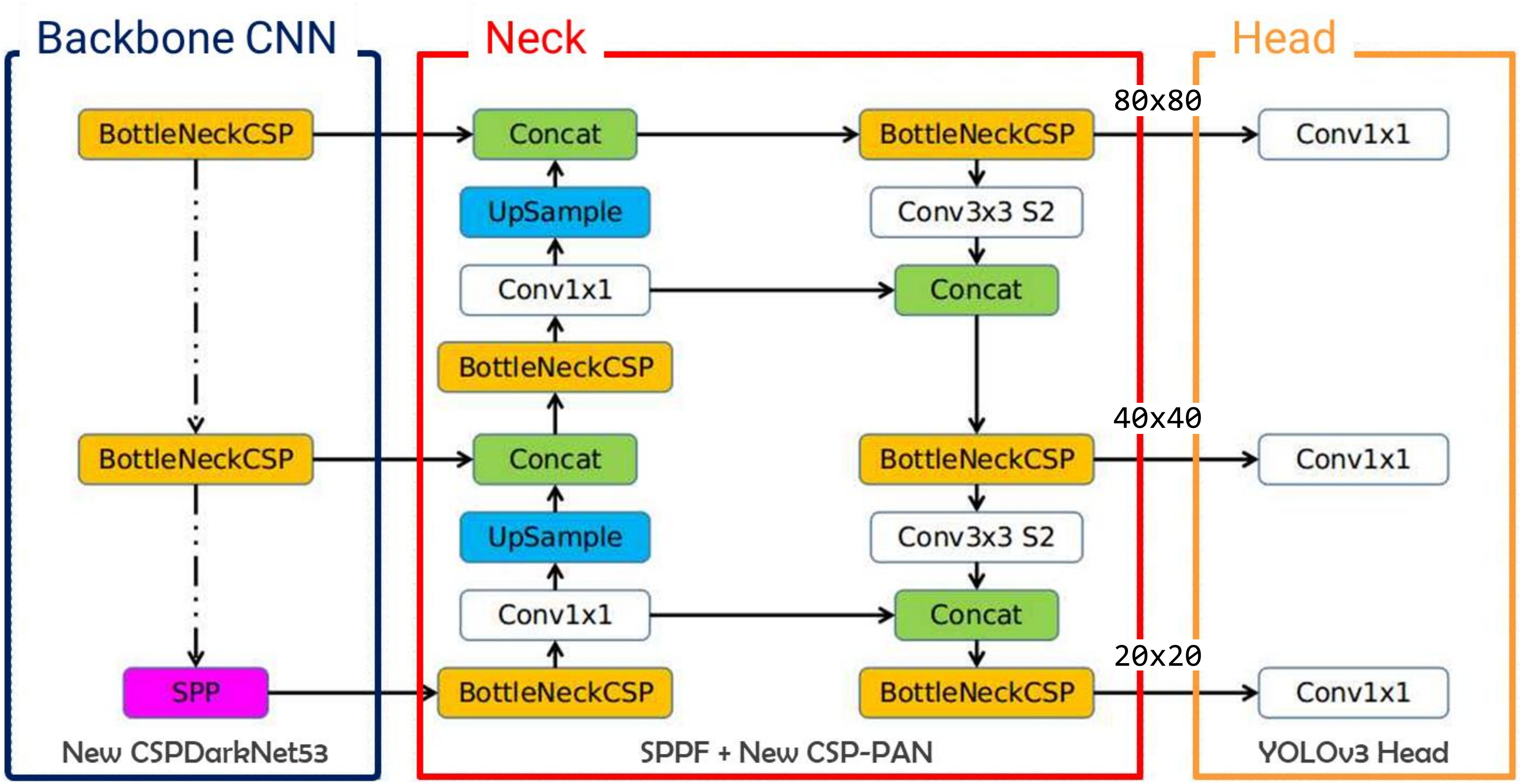

| |
|---|
| U-Net |
| FCN |
| DeepLabv3+ |

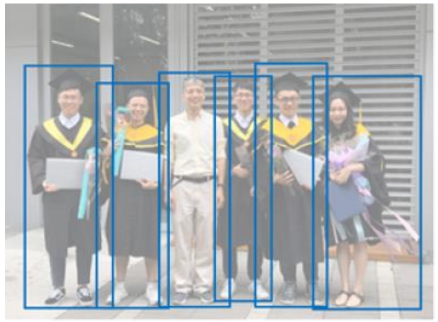
**Instance segmentation**




| |
|---|
| Mask R-CNN |
| MasLab |
| YOLACT |

# Architecture – Localization and Classification




```
"objects": [{
"class_id":0,
"name": “People",
"coordinates":{
"center_x":0.564741,
"center_y":0.438410,
"width":0.691837,
"height":0.637432},
"confidence":0.999482
}]
```


- Architecture of the YOLOv5

# Smart Machine Vision Tasks – Static

**Classification**

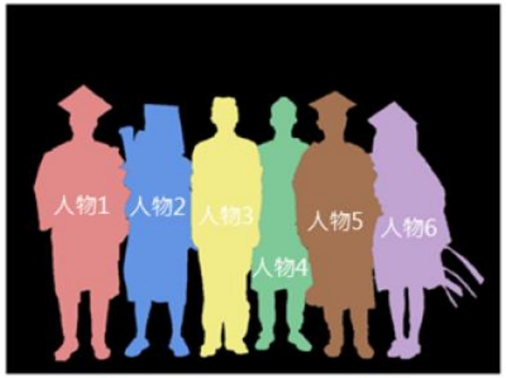



**Localization and Classification**



**Semantic segmentation**


**Instance segmentation**

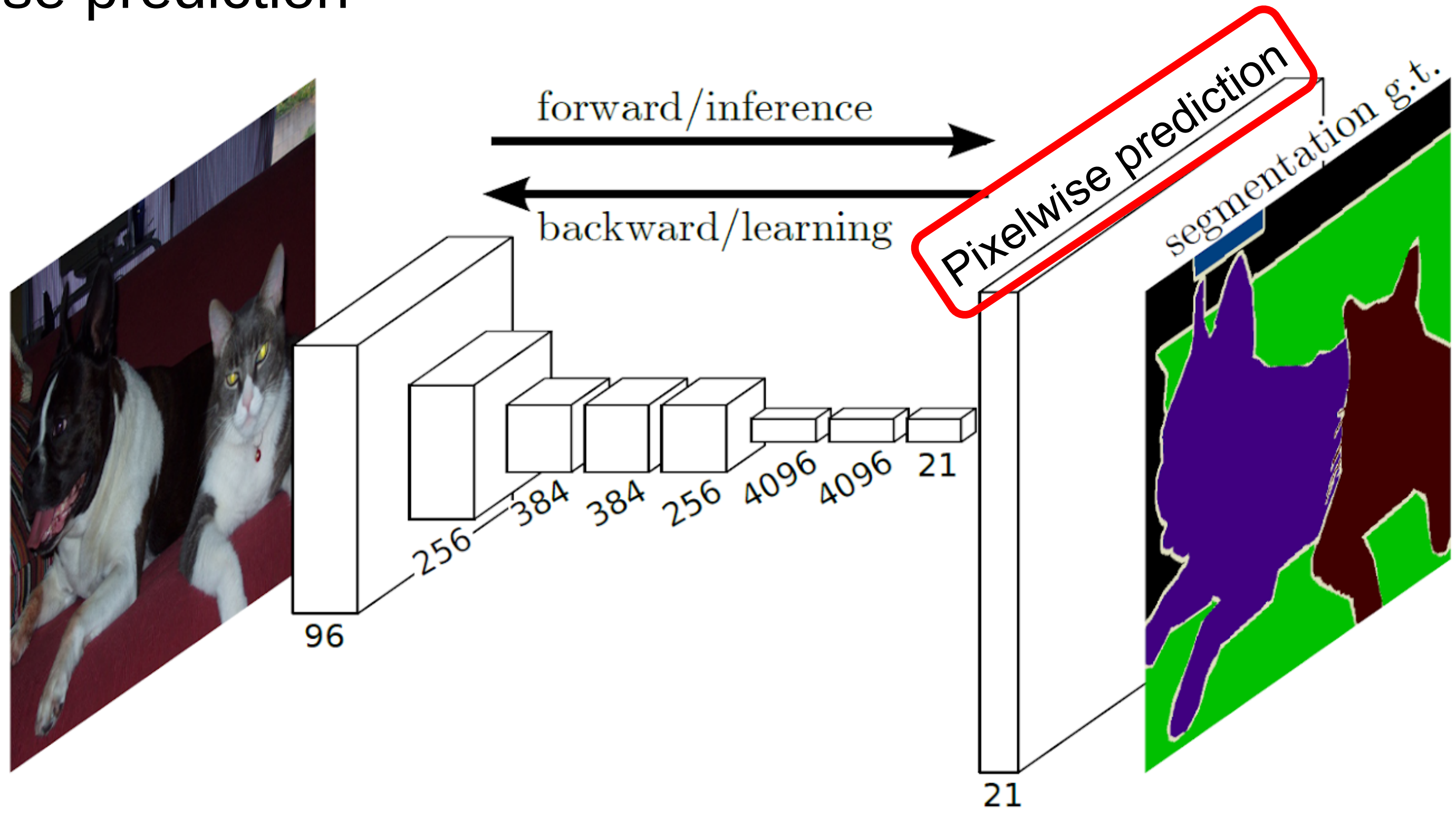



| Classification | Localization and Classification | Semantic segmentation | Instance segmentation |
|---|---|---|---|
| AlexNet | Fast R-CNN | U-Net | Mask R-CNN |
| VGG-16 | Faster R-CNN | FCN | MaskLab |
| ResNet-55 | YOLO v4 | DeepLabv3+ | YOLACT |
| EfficientNet | YOLO v5 | | |

Yan-Fu Kuo
Dept. of Biomechatronics Engineering

# Architecture – Segmentation

- Pixel-wise prediction

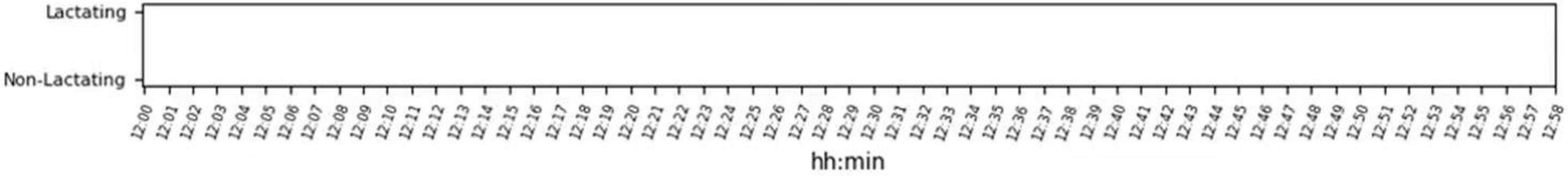



▲ Architecture of the Fully Convolutional Network (FCN)

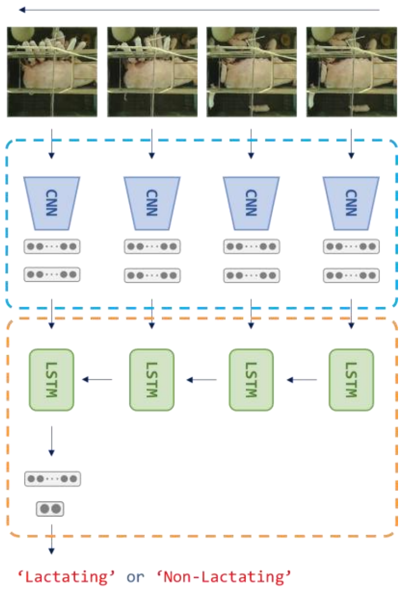
# Smart Machine Vision Tasks – Dynamics

Input video

Ouput prediction

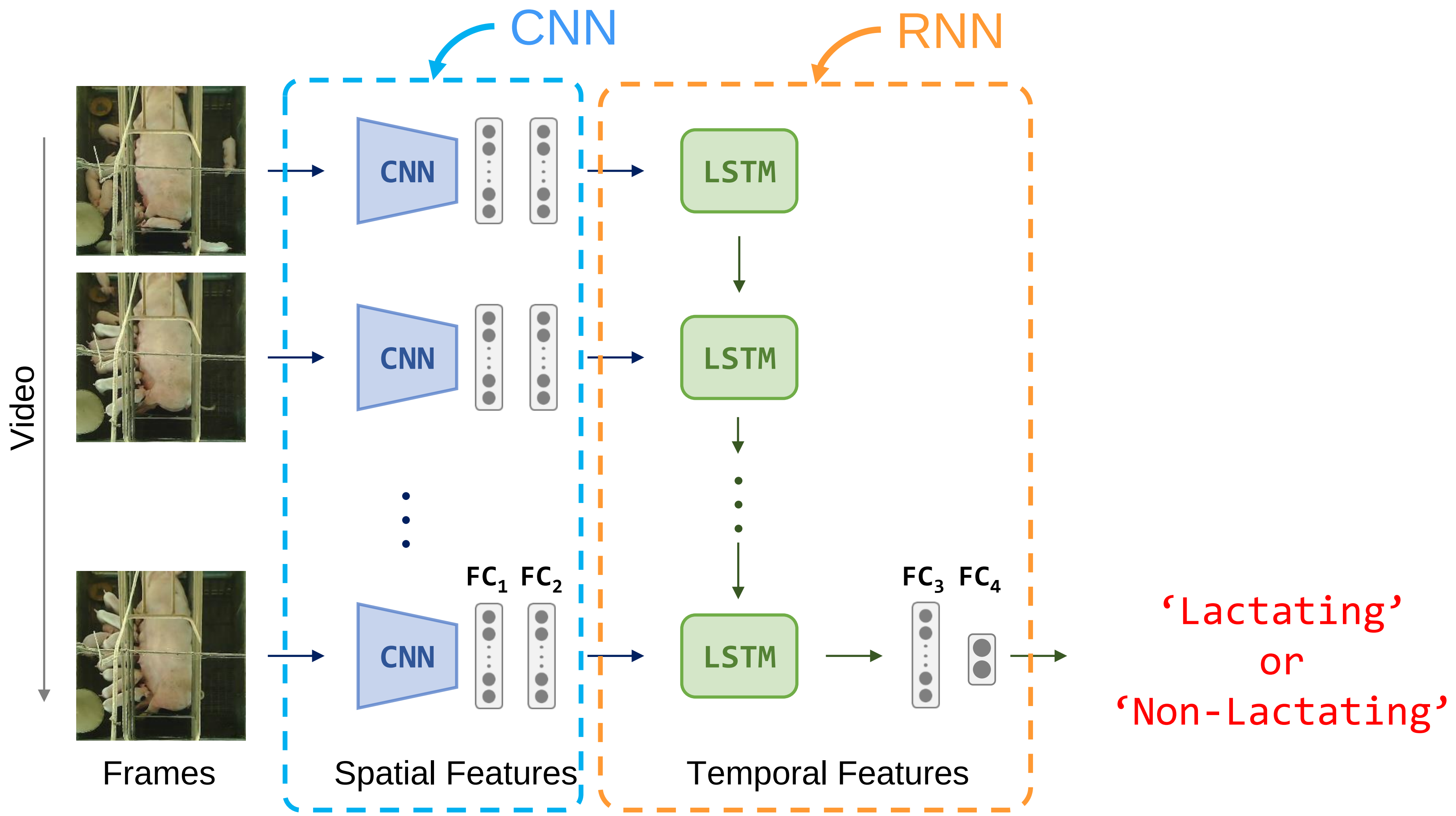



CRNN



Yan-Fu Kuo
Dept. of Biomechatronics Engineering

# Architecture – Action Detection

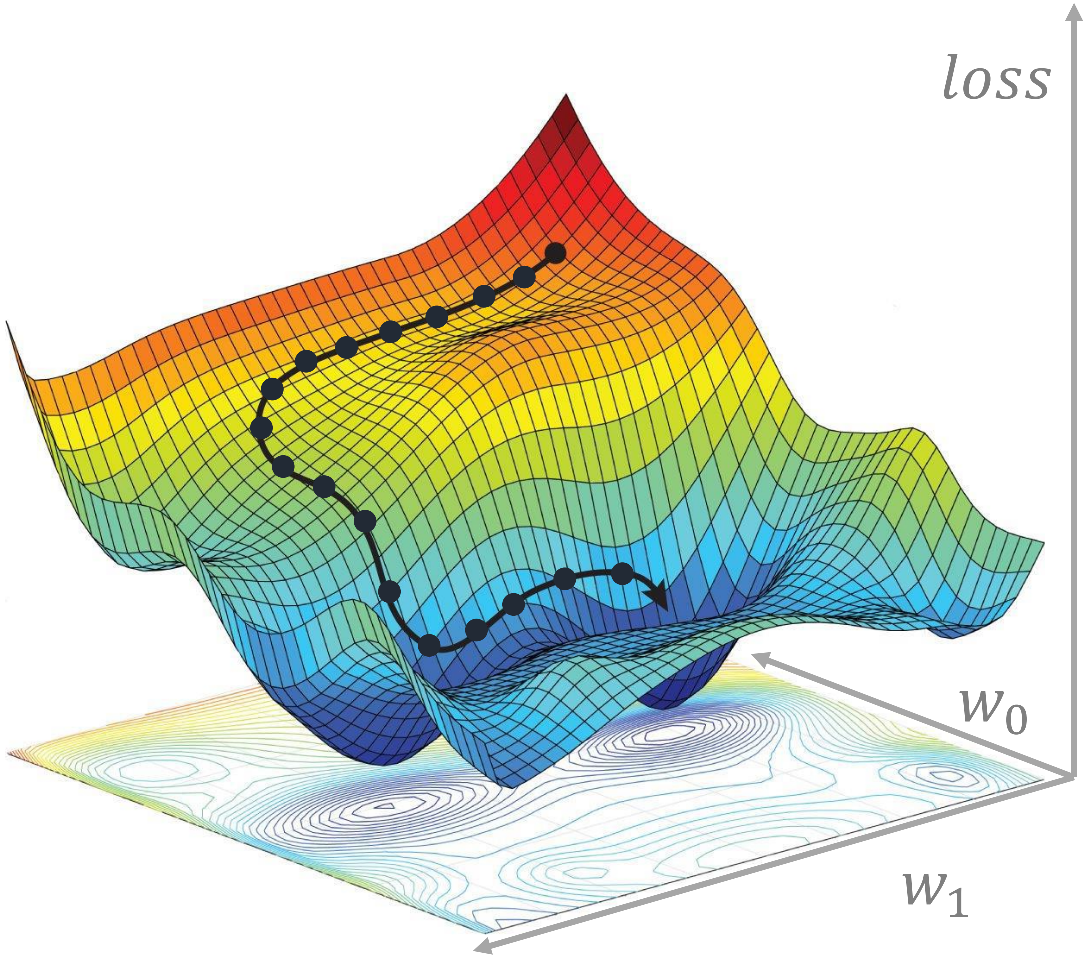



▲ Architecture of a Convolutional Recurrent Neural Network (CRNN)

# Optimizer

- An algorithm that reduce the “loss”

| |
|---|
| SGD |
| ★ SGD + Momentum |
| AdaGrad |
| RMSprop |
| ★ Adam |
| ⋮ |



# Hyperparameters

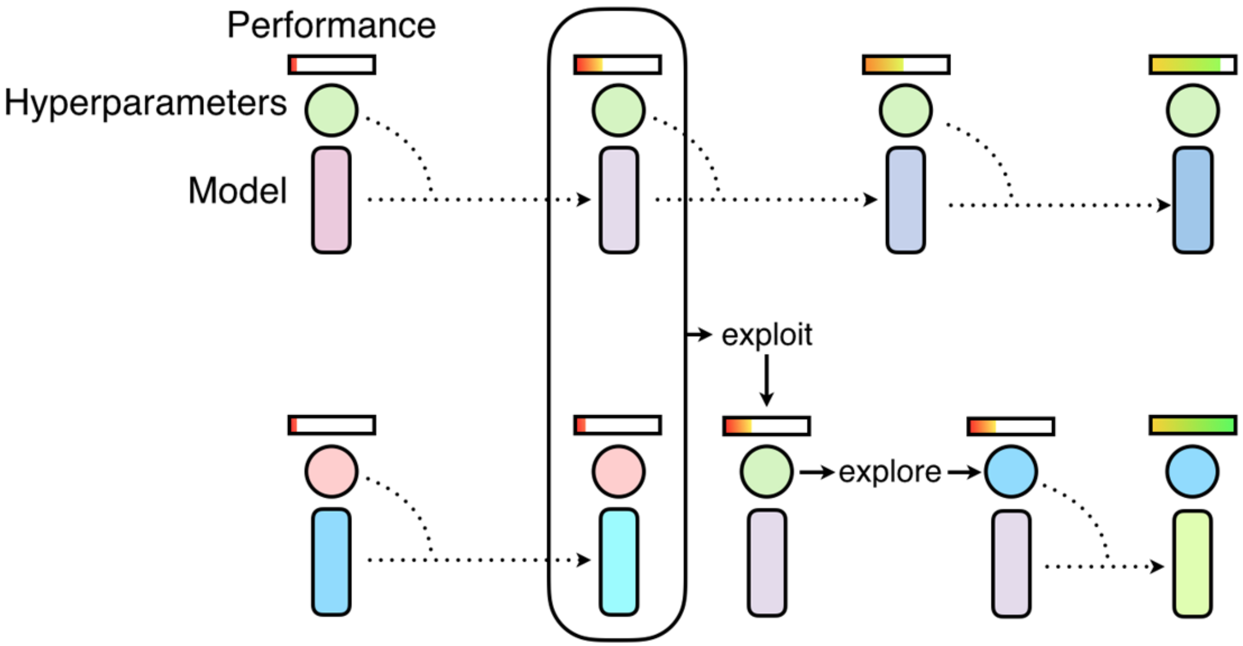
- Used to control the learning process
- Determined manually

| Training strategy | Optimizer | Loss function |
|---|---|---|
| Epochs | Learning rate | Class loss |
| Batch size | Momentum | Object loss |
| Confidence threshold | Bias | Box loss |
| IOU threshold | Decay rate | Layer loss |
| ⋮ | ⋮ | ⋮ |

# Hyperparameter Tuning

- Automatically choosing the best hyperparameters
- Required very huge GPU resource

Population Based Training



# Training Facility

Google Cloud

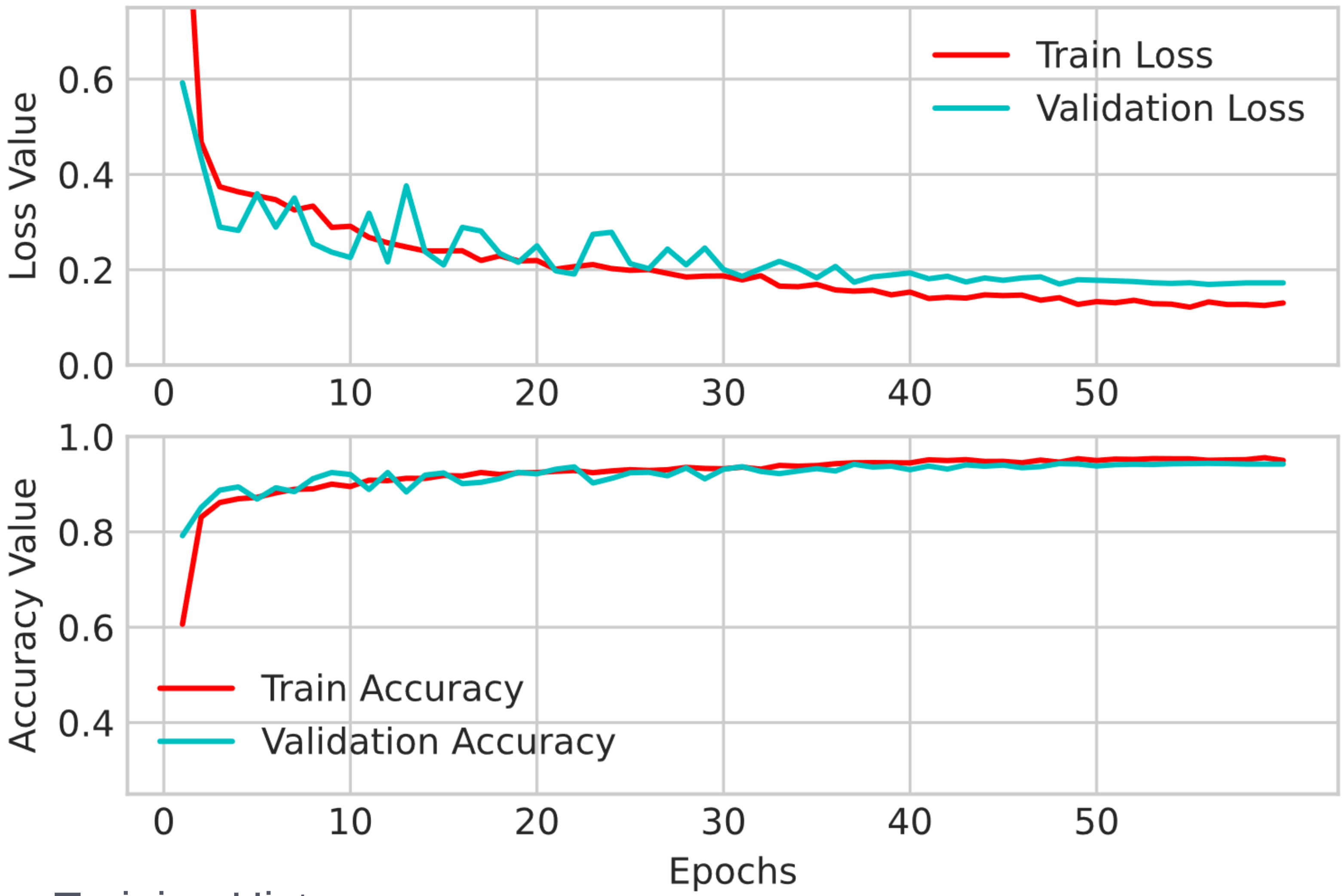
Azure

amazon web services

工具人實驗室伺服器

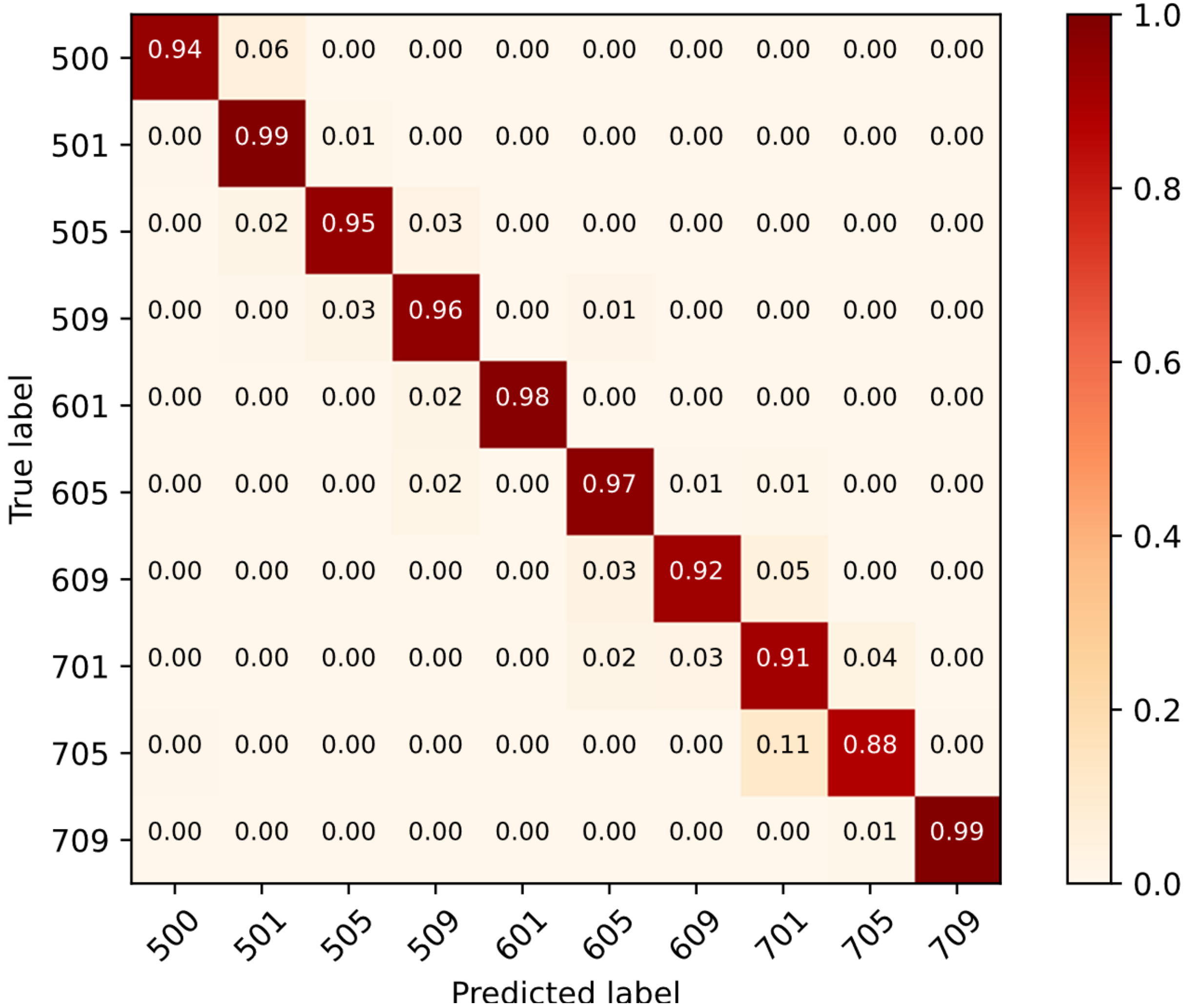
https://www.mlmvlab.bime.ntu.edu.tw/

# Model Evaluation – Model Training




▲ Training History

# Model Evaluation – Confusion Matrix




- Confusion Matrix

# Deep Learning Frameworks

Microsoft Cognitive Toolkit

mxnet

Keras

PyTorch

ONNX

TensorFlow

Caffe2

DL4J

# Implementation Flow of Machine Vision

## Image acquisition

Acquire images manually by human or automatically by installed cameras and image transmission equipments.

## Database

Store images with labels and feature locations, and store other useful information such as temperature.

## User access

Establish microservice and serve as a bridge between the model and public by docker, kubernetes...

## Machine vision

**Dataset**
Split images into training, validation, and testing dataset.

**Image augmentation**
Increase image variety to improve model robustness.

**Model architecture**
Utilize suitable machine learning models to achieve the target.

**Model training**
Use training dataset and choose suitable optimizer to update parameters and use validation dataset to validate.

**Model evaluation**
Test model performance by testing dataset.

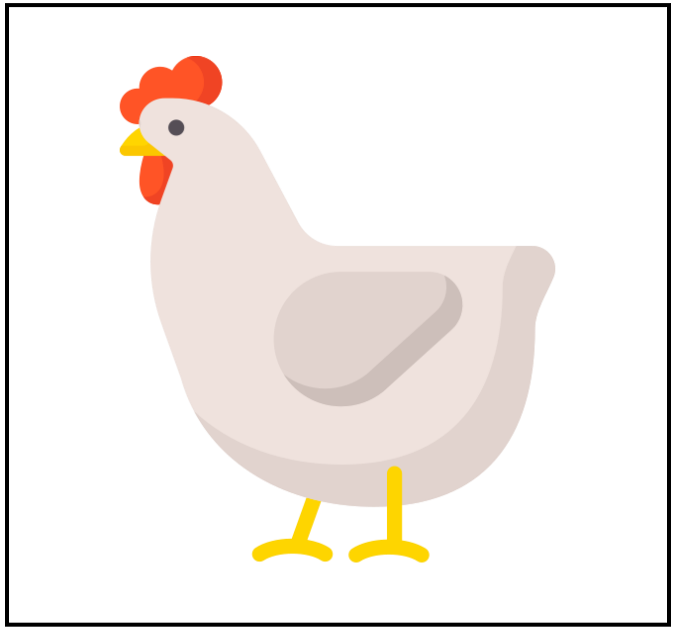
**Framework**
Select suitable framework for developing models

# Implementation Flow of Machine Vision

**Database**

Store images with labels and feature locations, and store other useful information such as temperature.

# Database



Image

Label

Useful information

# Database

**Internal user**

1. Data analysis
2. Image manipulation
3. Developing model

**External user**

1. Open source
2. Data analysis
3. Providing new source

# Implementation Flow of Machine Vision

## Image acquisition

Acquire images manually by human or automatically by installed cameras and image transmission equipments.

## Database

Store images with labels and feature locations, and store other useful information such as temperature.

## User access

Establish microservice and serve as a bridge between the model and public by docker, kubernetes...

## Machine vision

**Dataset**
Split images into training, validation, and testing dataset.

**Image augmentation**
Increase image variety to improve model robustness.

**Model architecture**
Utilize suitable machine learning models to achieve the target.

**Model training**
Use training dataset and choose suitable optimizer to update parameters and use validation dataset to validate.

**Model evaluation**
Test model performance by testing dataset.

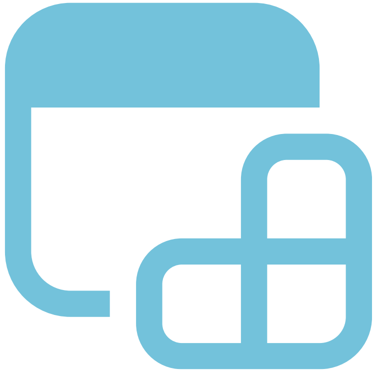
**Framework**
Select suitable framework for developing models

# Implementation Flow of Machine Vision

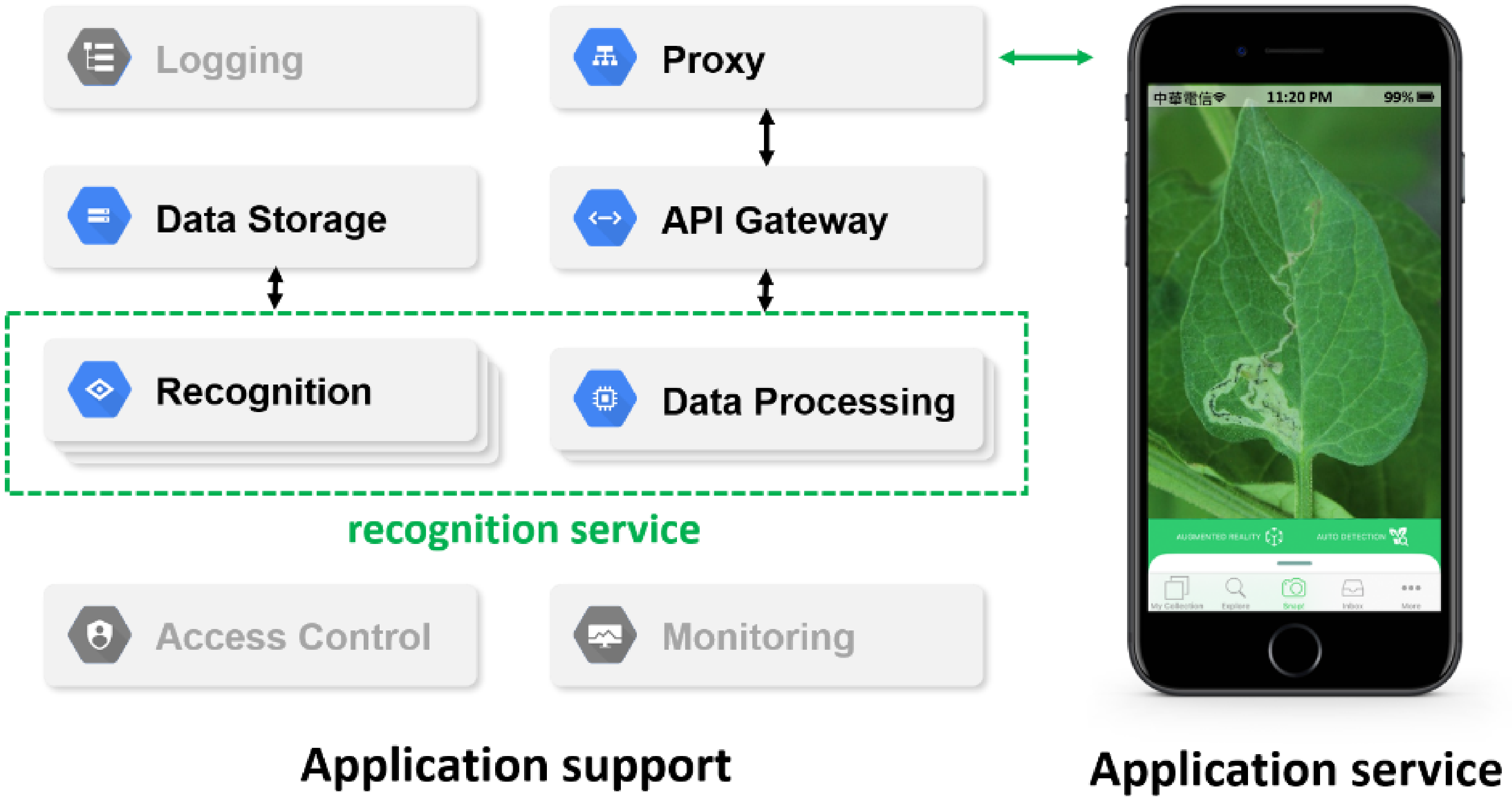
**User access**

Establish microservice and serve as a bridge between the model and public by docker, kubernetes...

# User Access – Interface



# User Access – Service Architecture

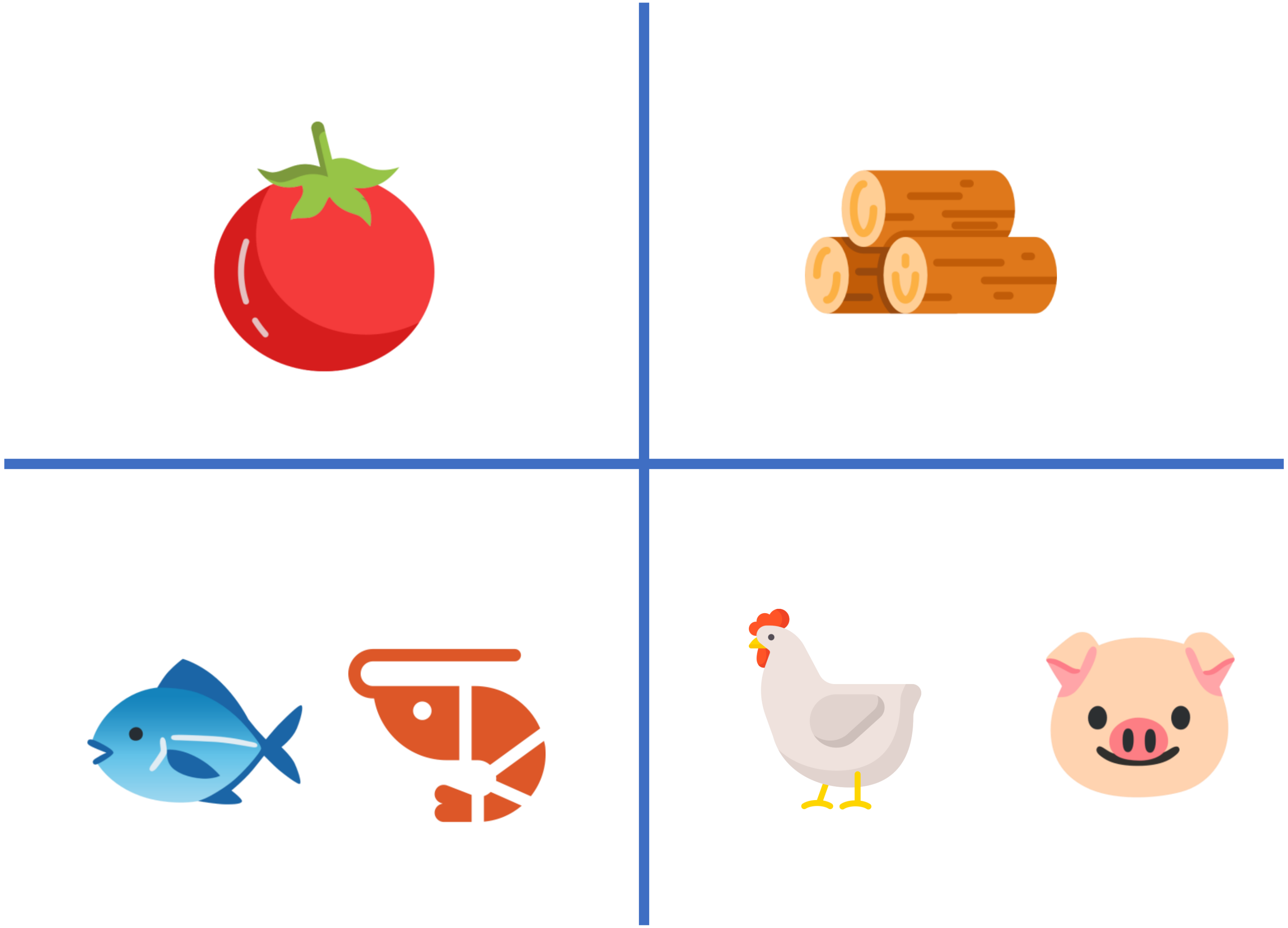


# User Access – Useful Service Tools

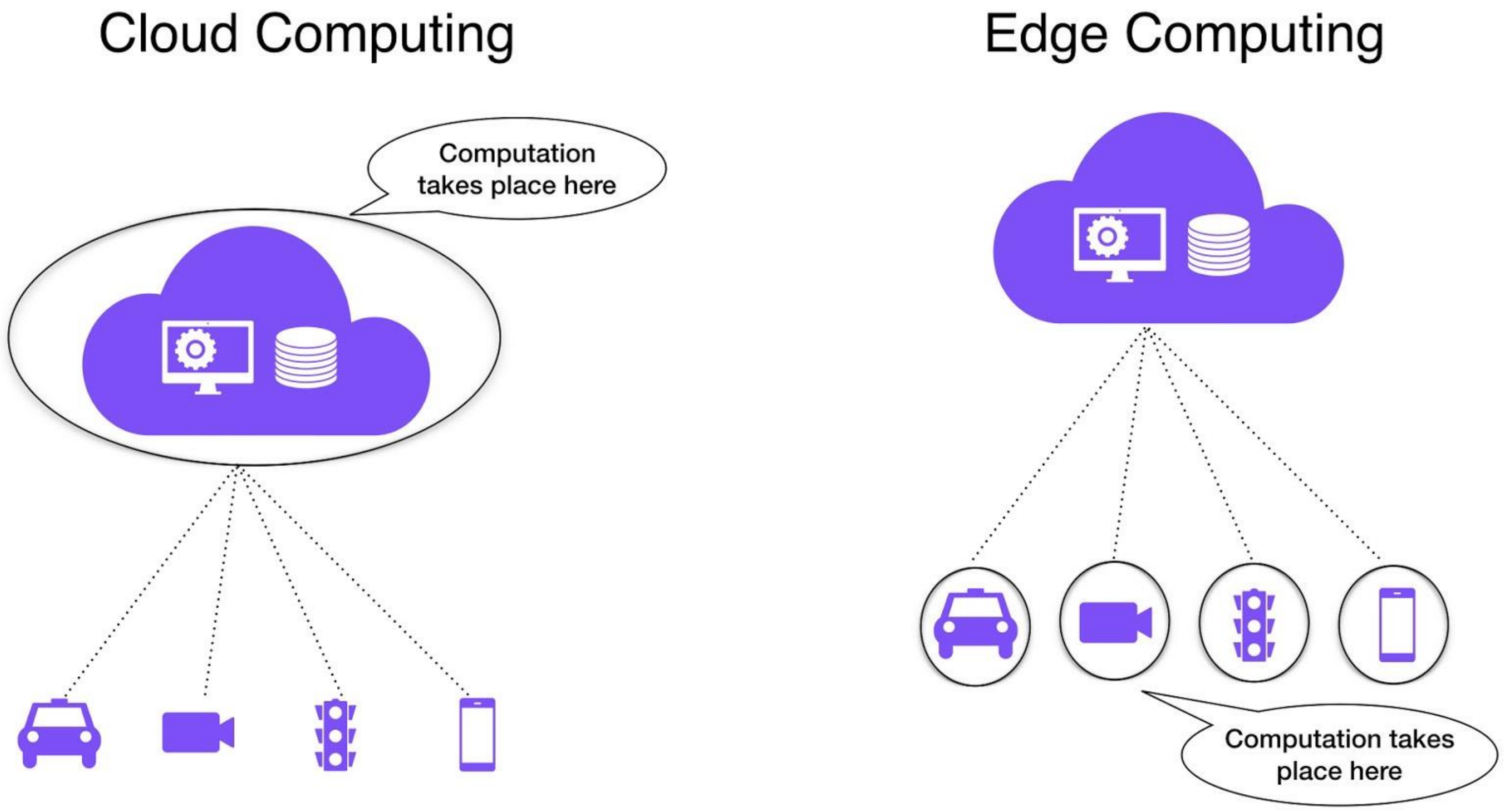
docker

kubernetes

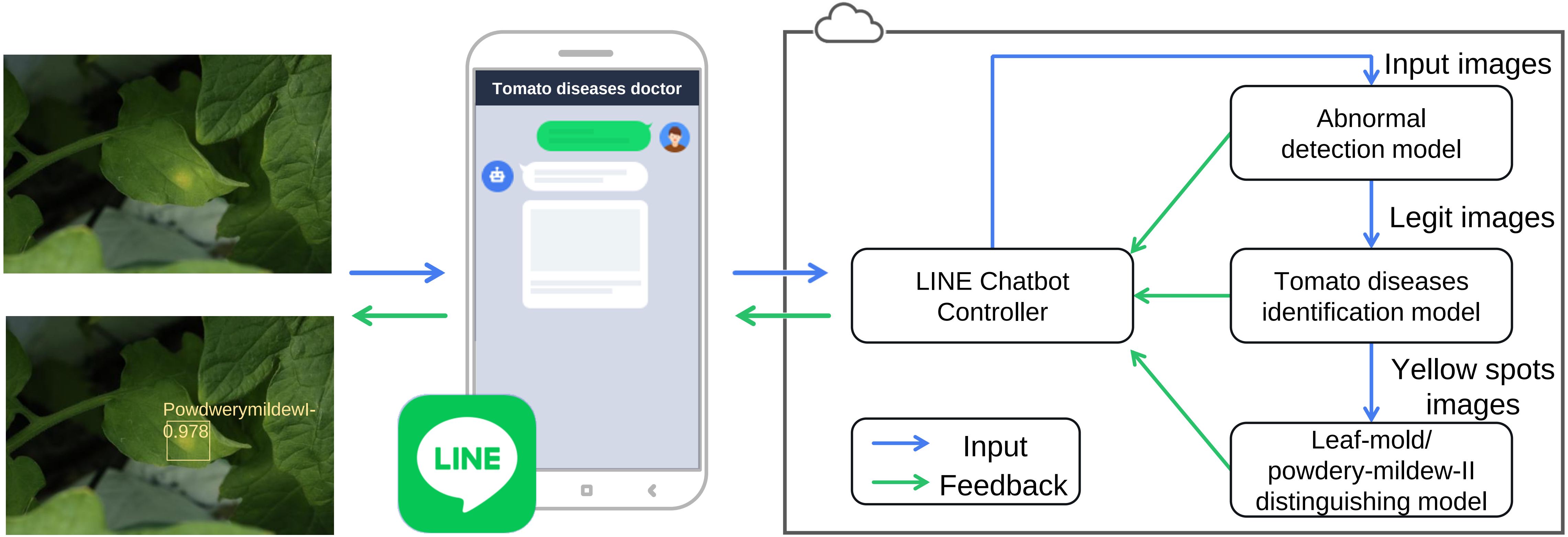
# Applications of Machine Vision



# Computing Methods



# Tomato Disease Identification

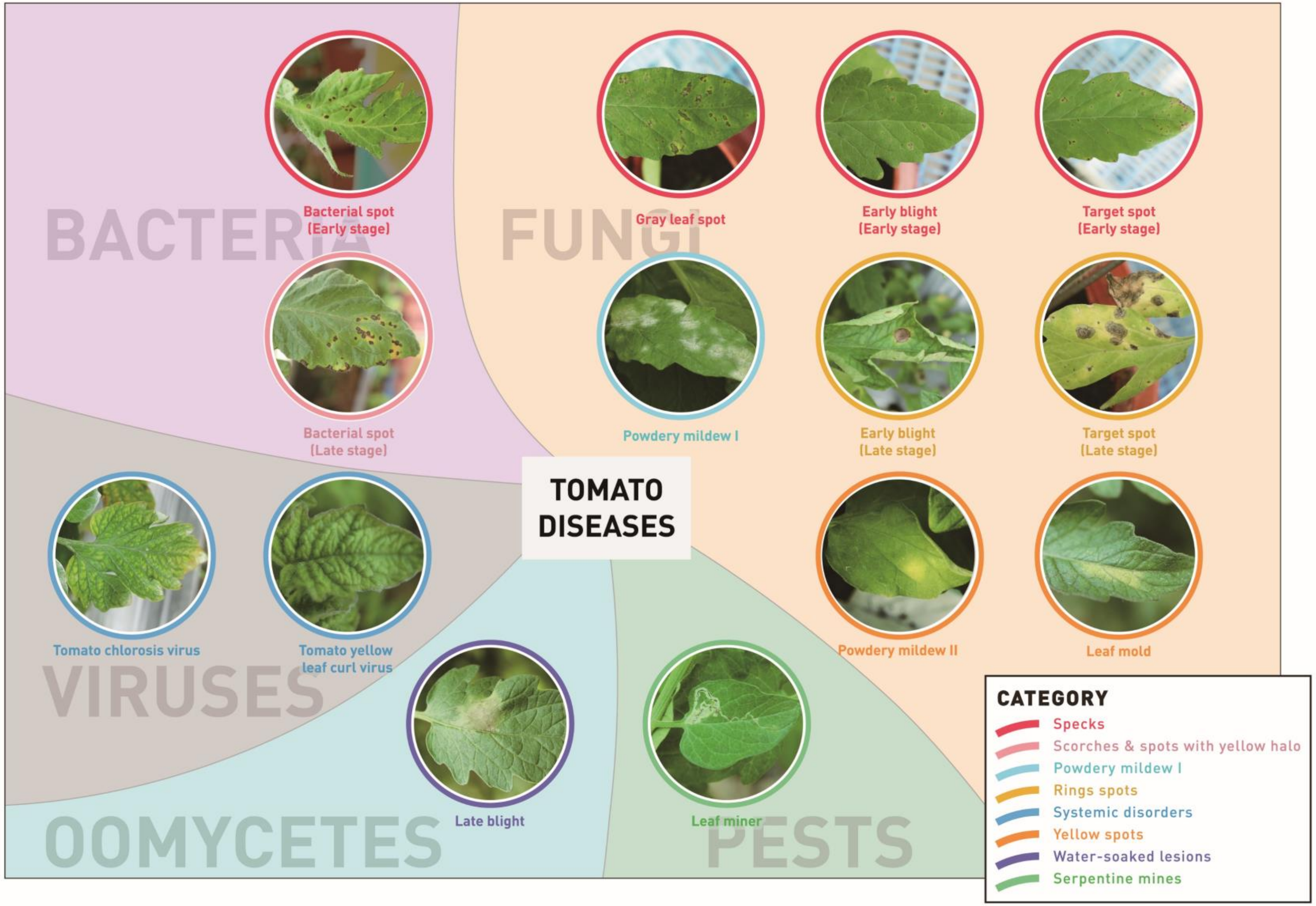



Tomato leaves images

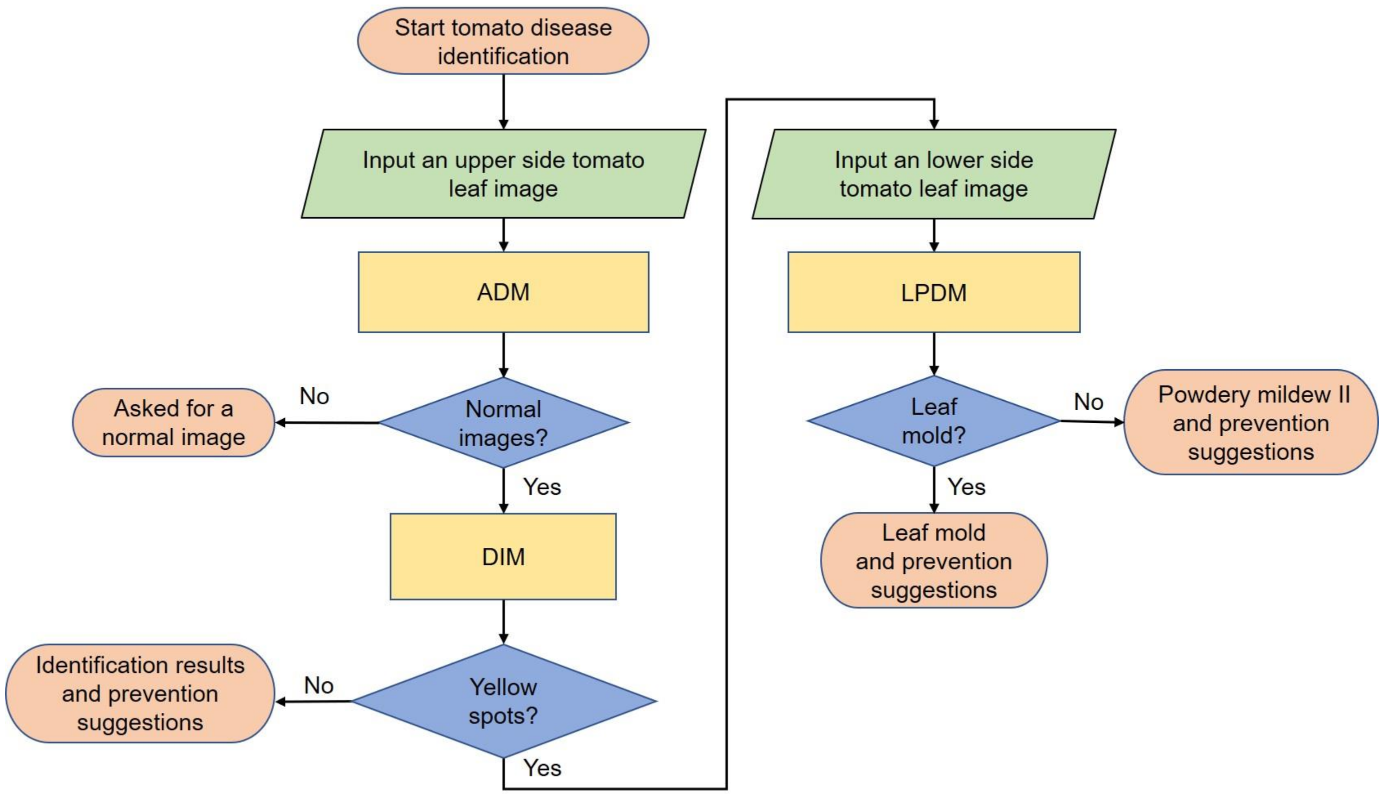
LINE Chatbot

Tomato diseases identification system

# Tomato Diseases



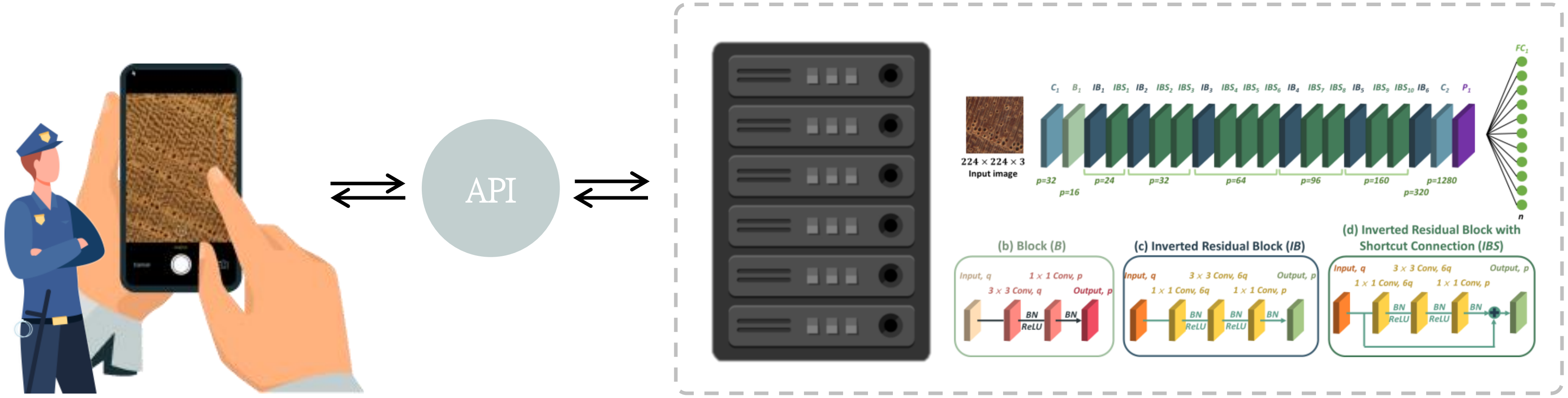
# Tomato Disease Identification





Yan-Fu Kuo
Dept. of Biomechatronics Engineering

# Wood Recognition



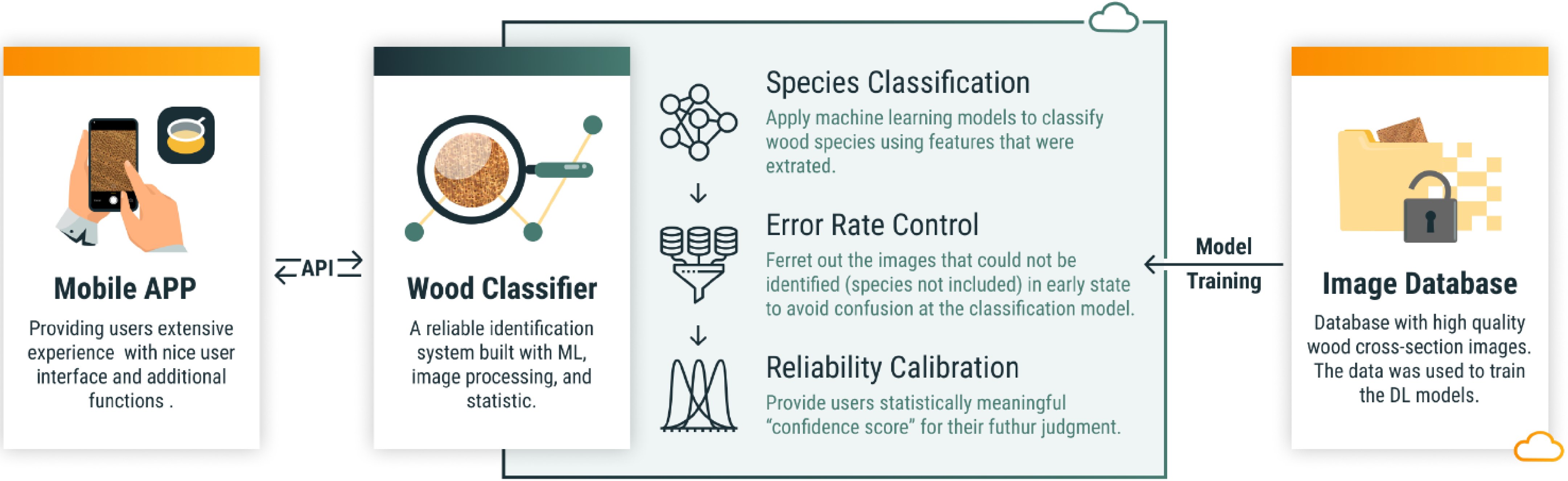

users

species identification and data storage server

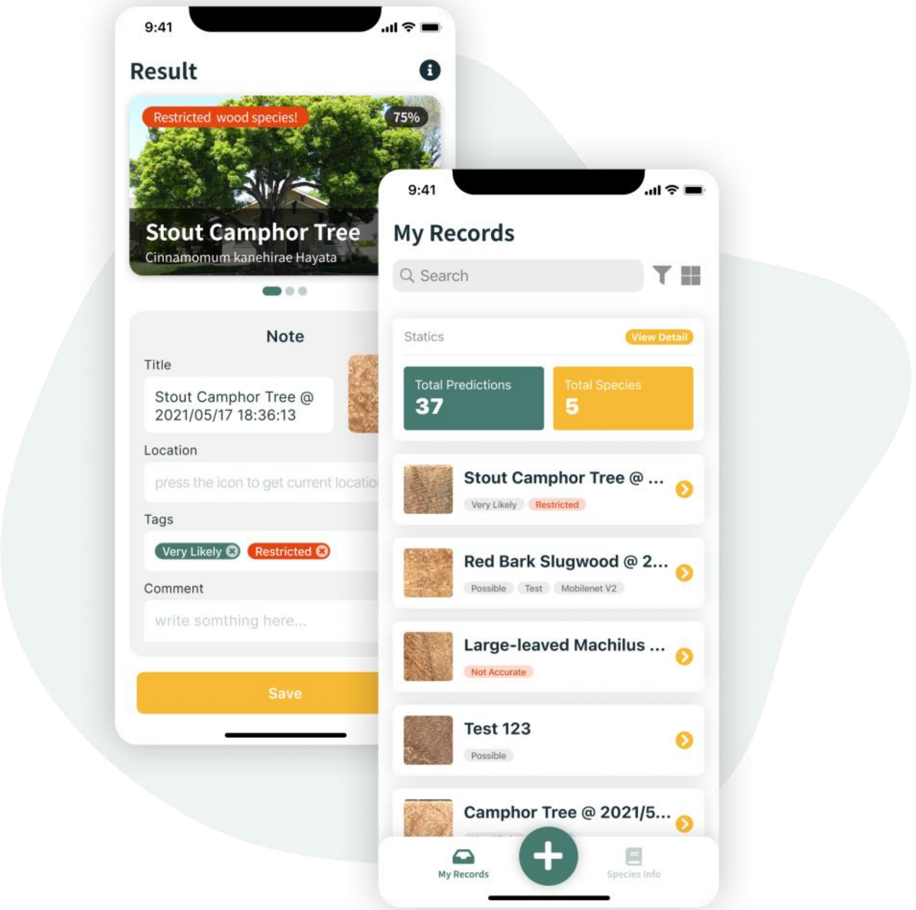
Yan-Fu Kuo
Dept. of Biomechatronics Engineering

# Overview of the Proposed System

- Composed of three components: (1) mobile APP, (2) wood classifier, and (3) image database

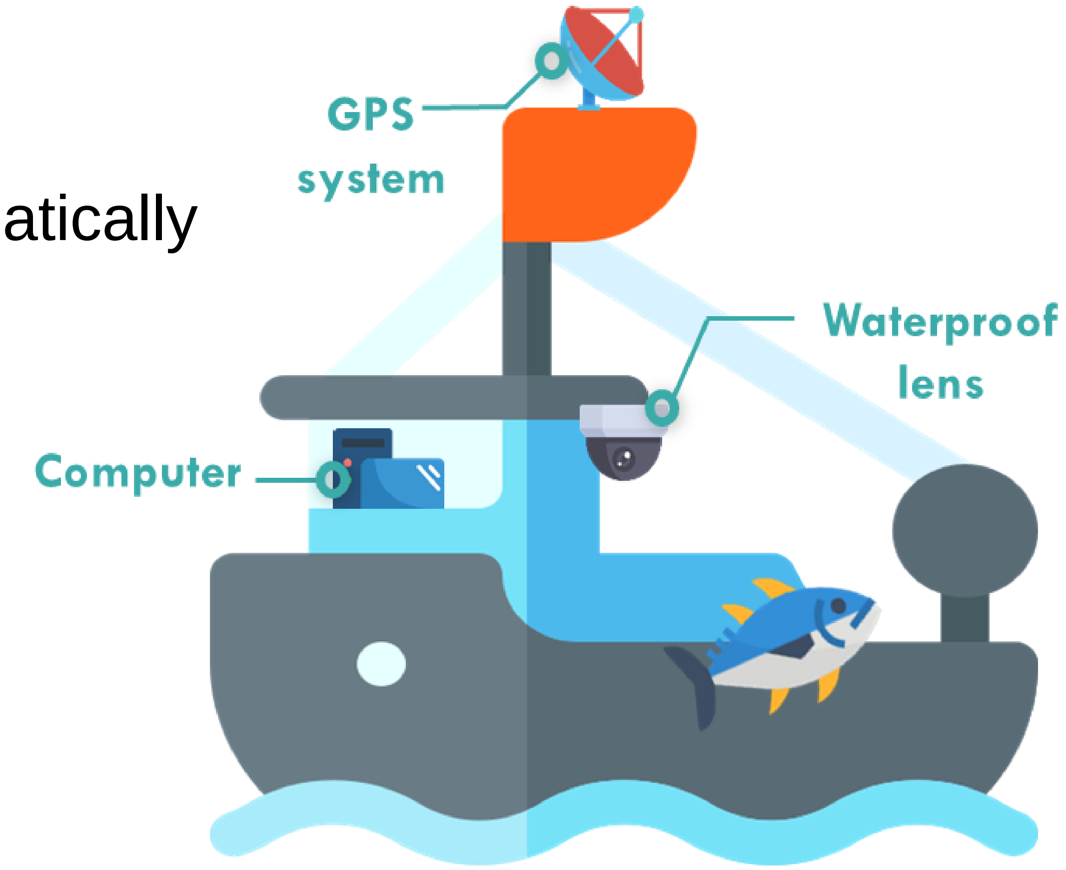


# The Mobile Application

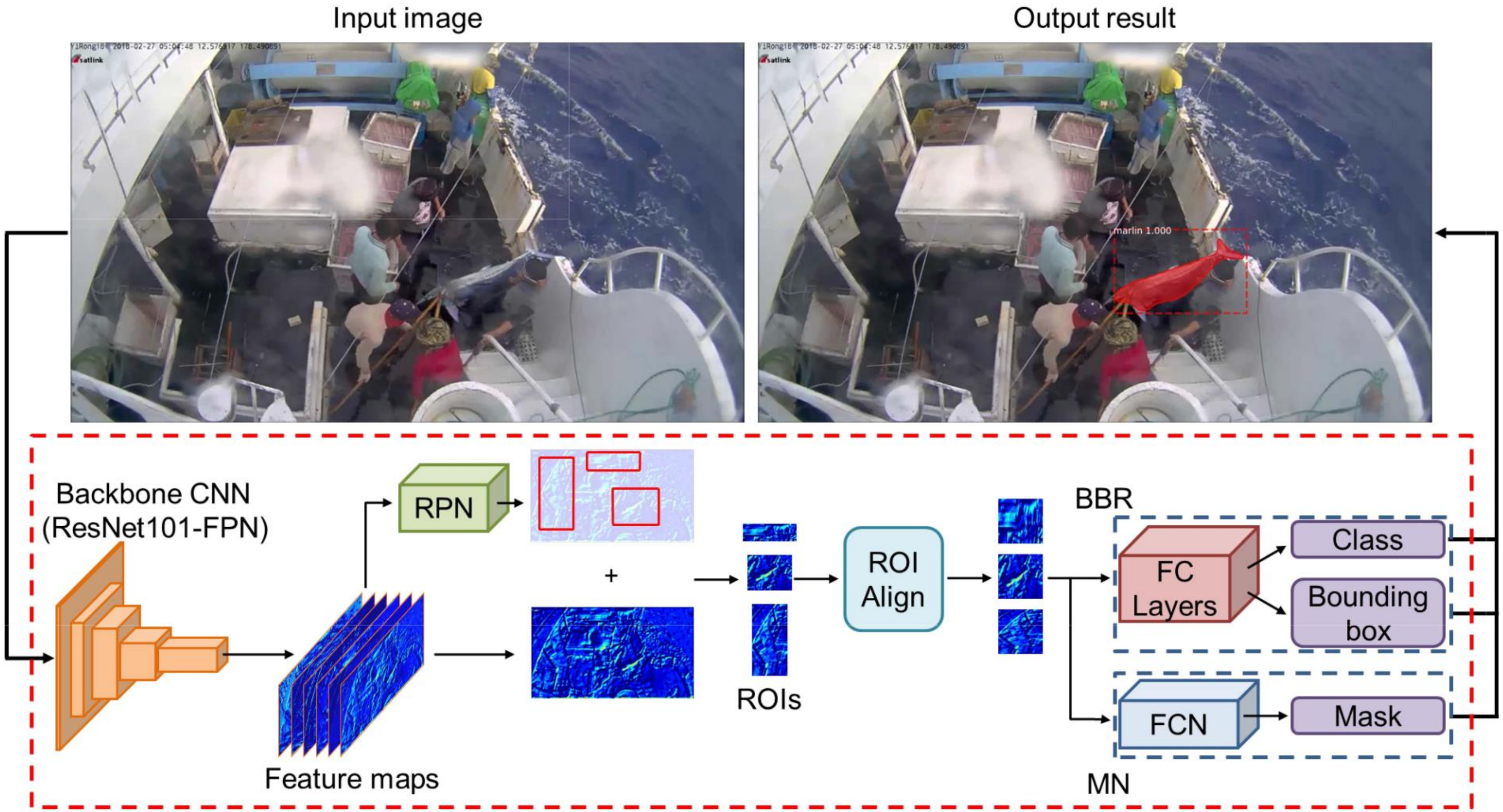



# Fish Type Identification and Counting

- Electronic monitoring system
- Identifying the fish types automatically
- Measuring the length of the fish automatically

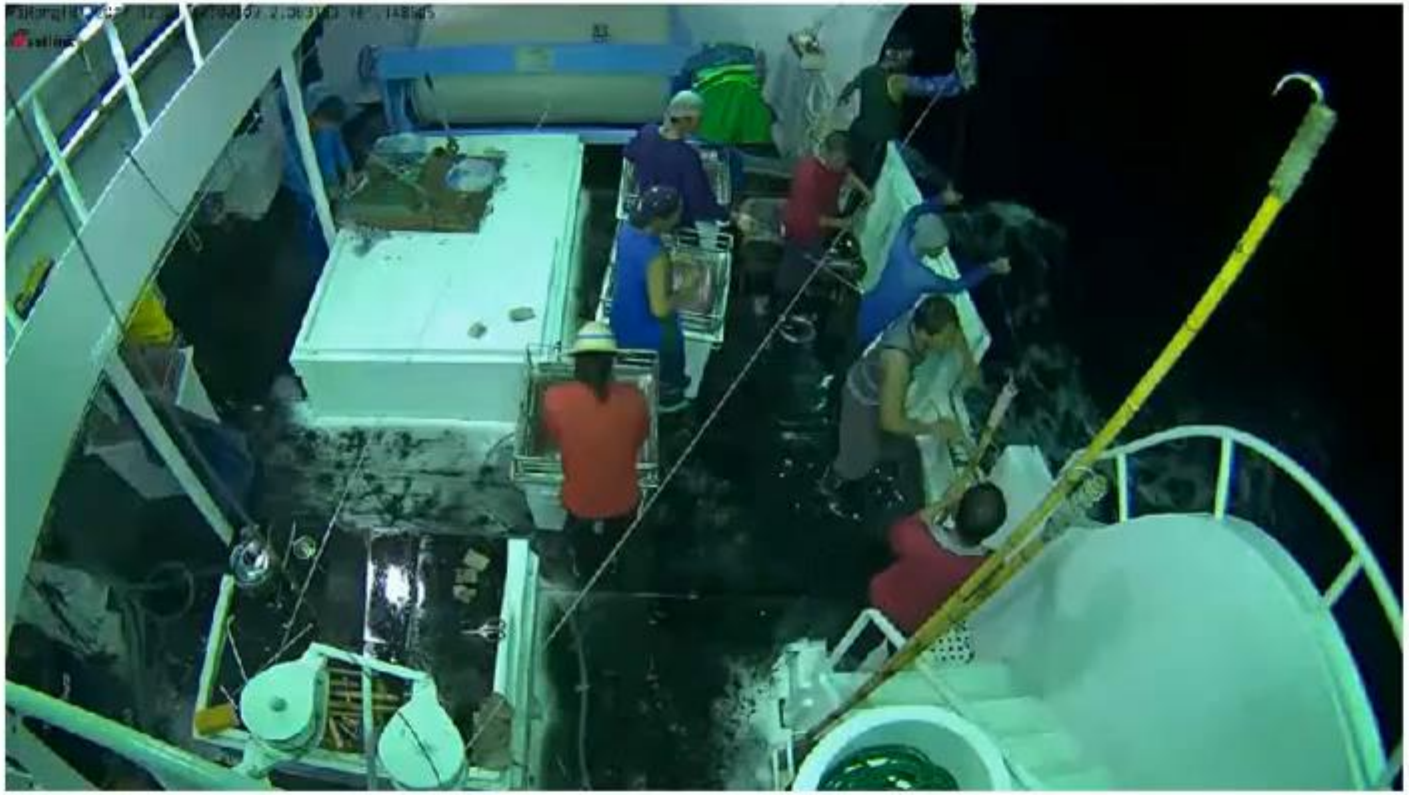


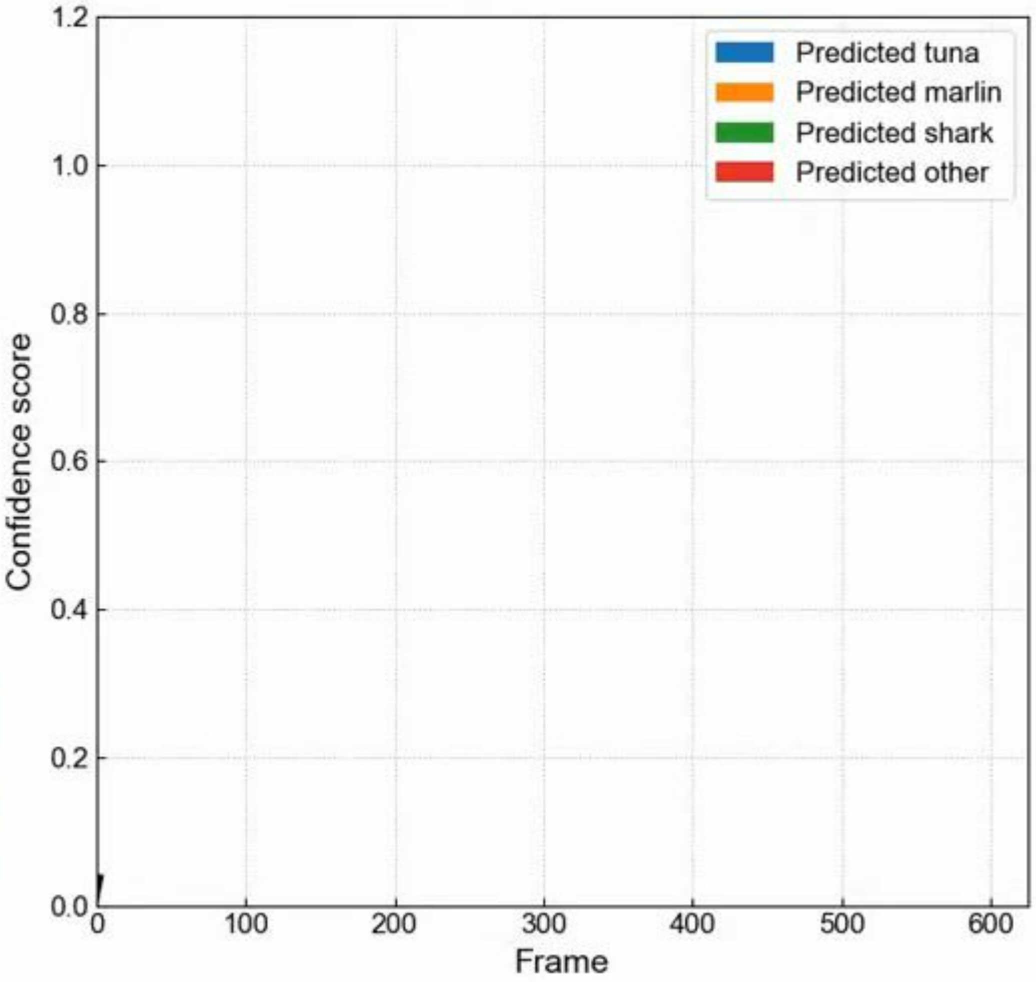
# Fish Type Identification and Counting




Yan-Fu Kuo
Dept. of Biomechatronics Engineering

# Fish Type Identification and Counting

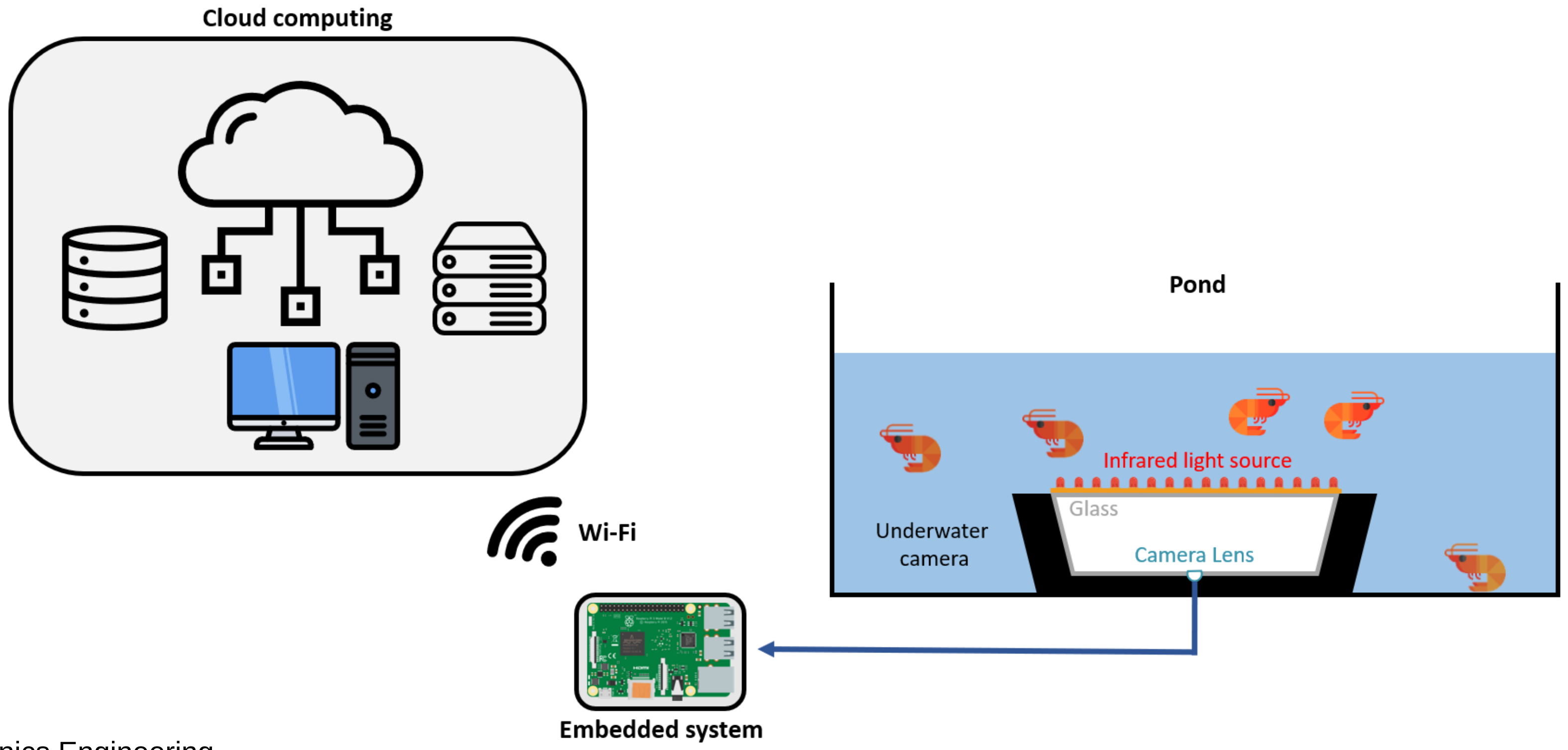




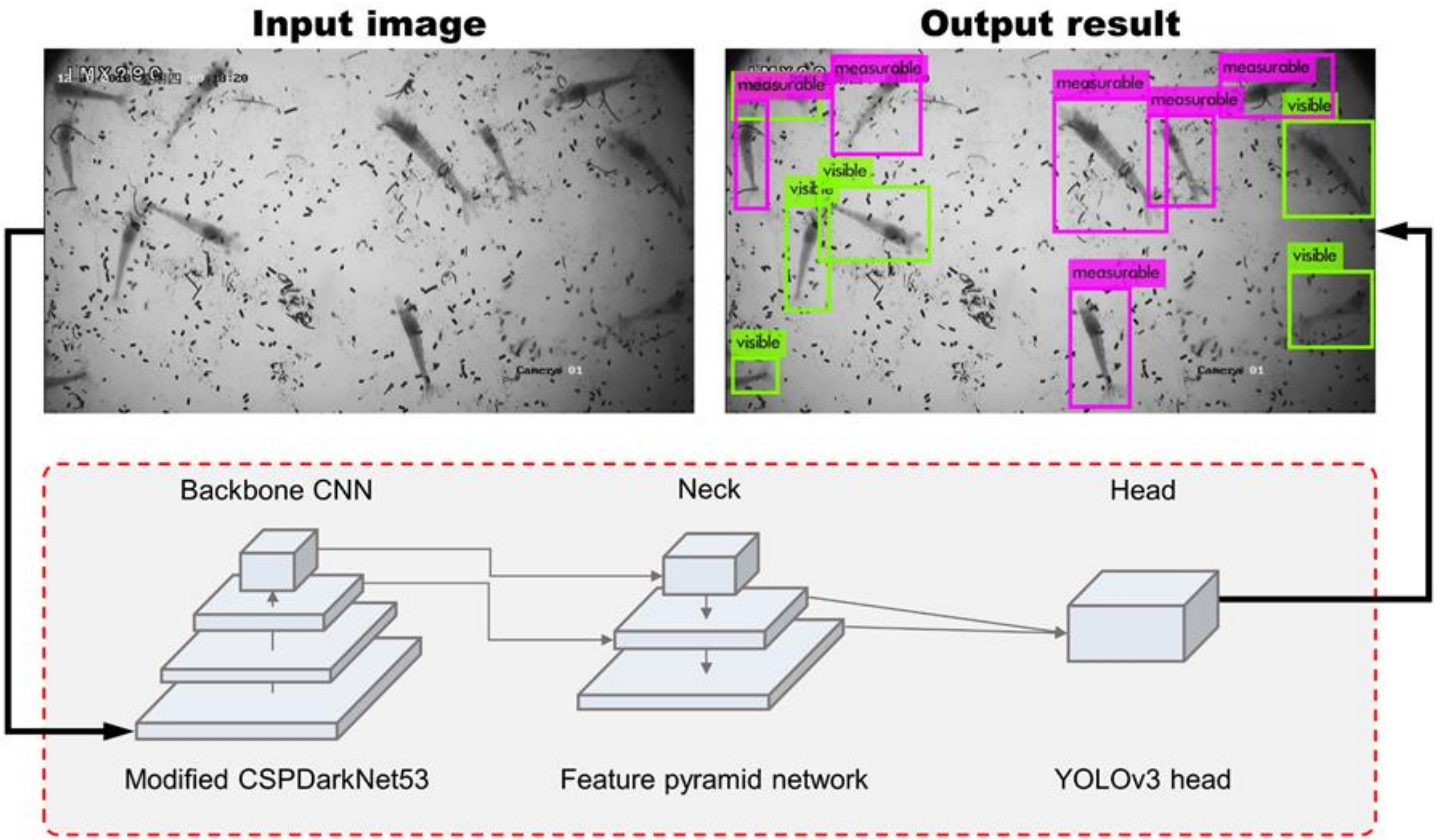

Yan-Fu Kuo
Dept. of Biomechatronics Engineering

# Shrimp Length Measuring

- The shrimps raised in a concrete-walled outdoor ponds
- Videos of the shrimps acquired using an underwater camera

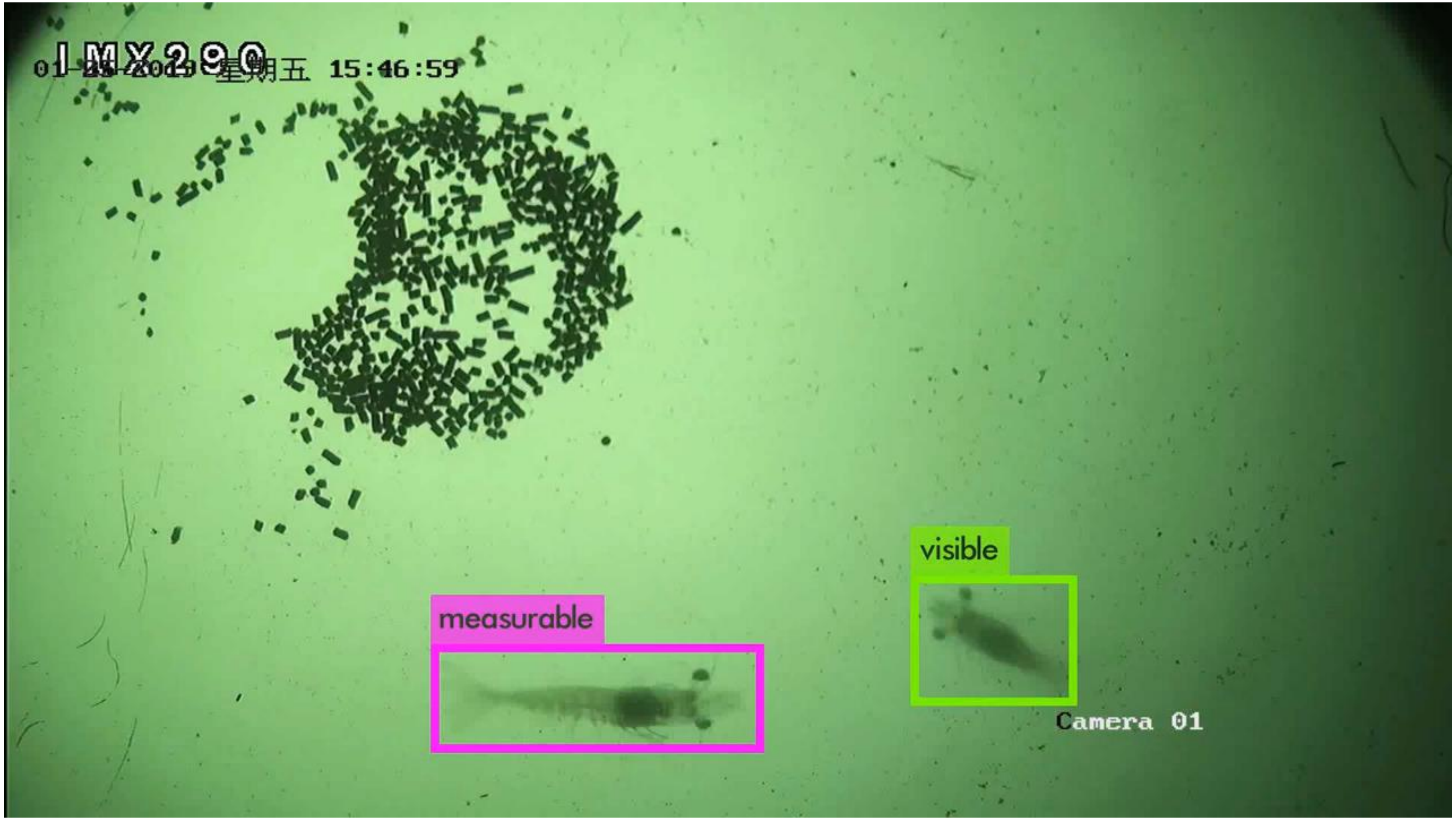


# Shrimp Length Measuring

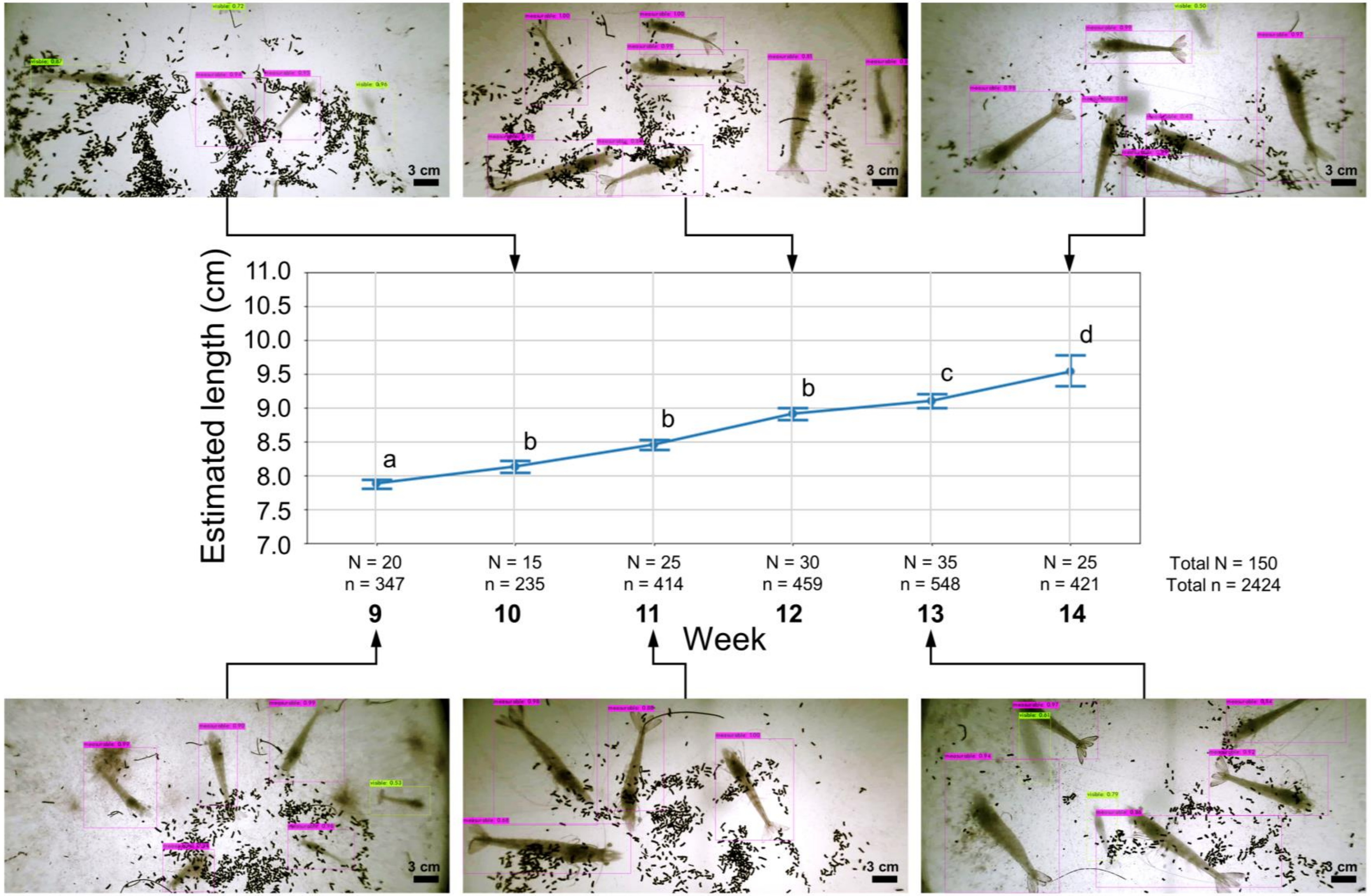



Yan-Fu Kuo
Dept. of Biomechatronics Engineering

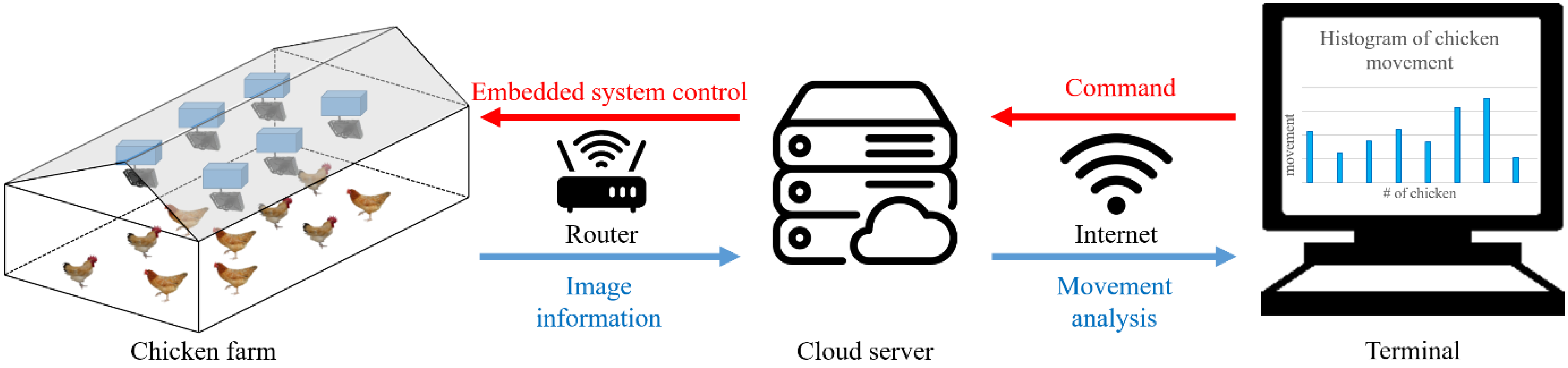
# Shrimp Length Measuring



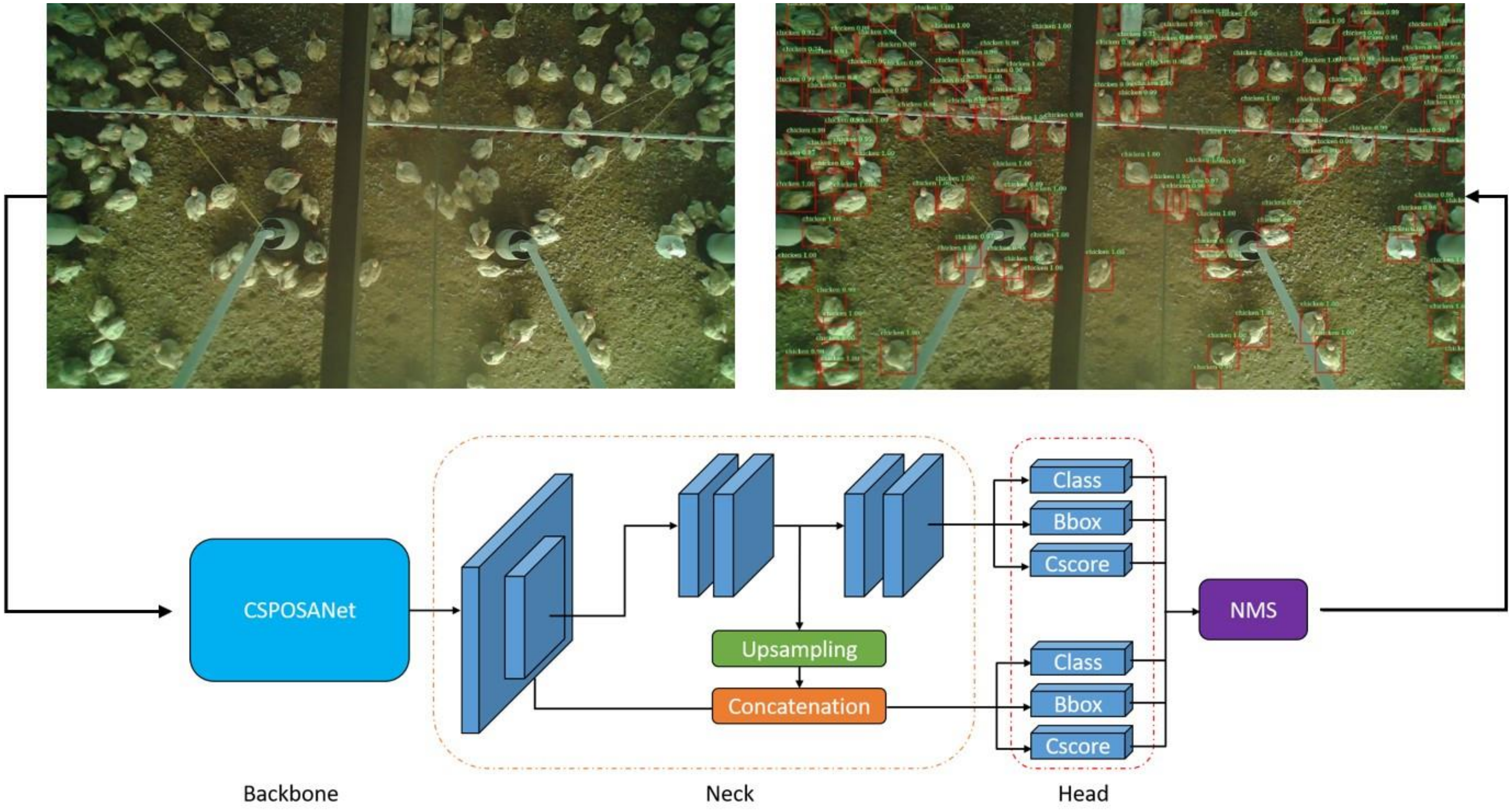
# Long-term Monitoring of Shrimp Length



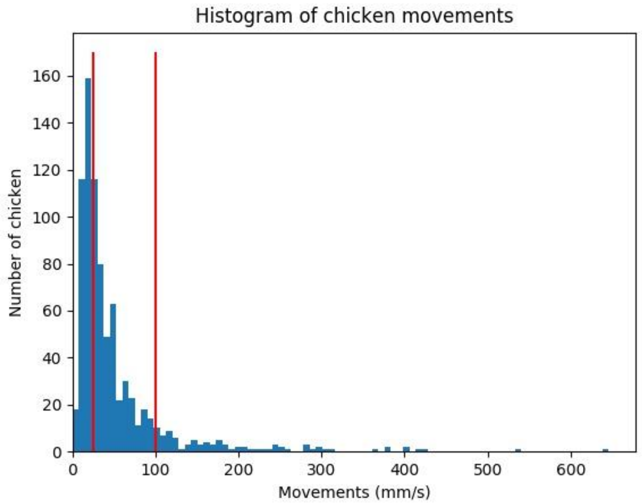
# Chicken Dispersion and Movement Monitoring



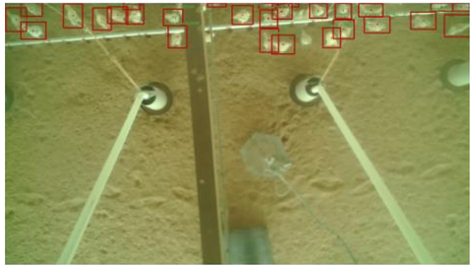
# Chicken Detection



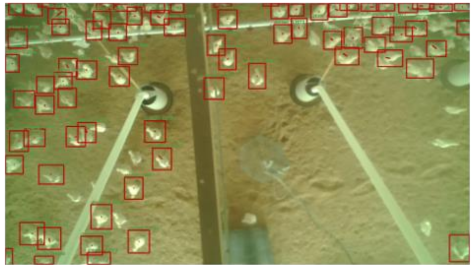
# Chicken Tracking, Movement, and Dispersion



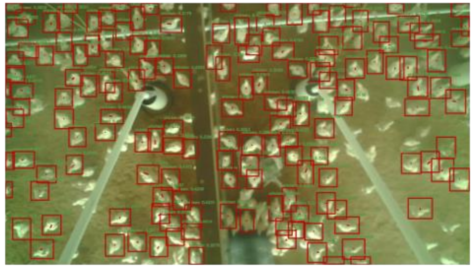

NNI = 0.459



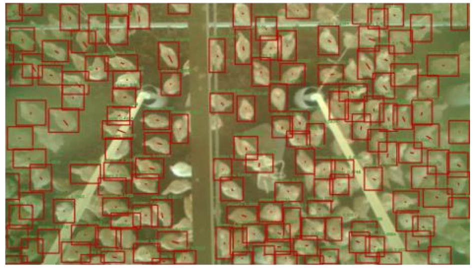
NNI = 1.013



NNI = 1.269

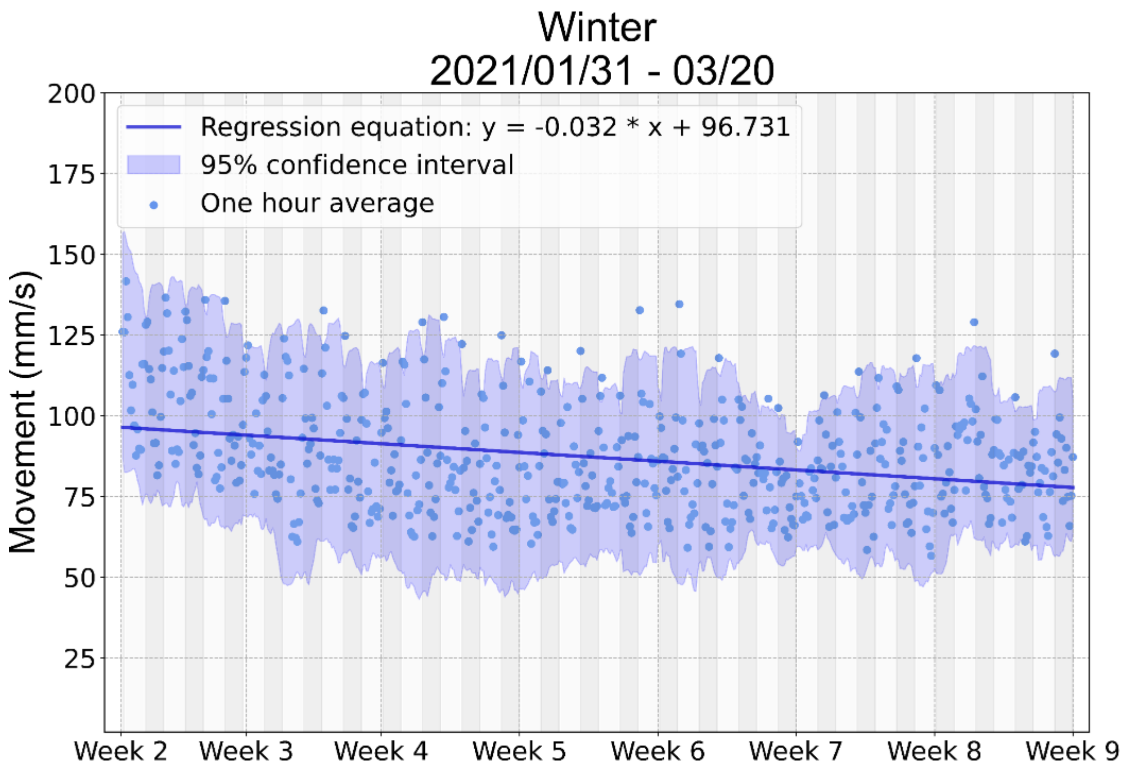


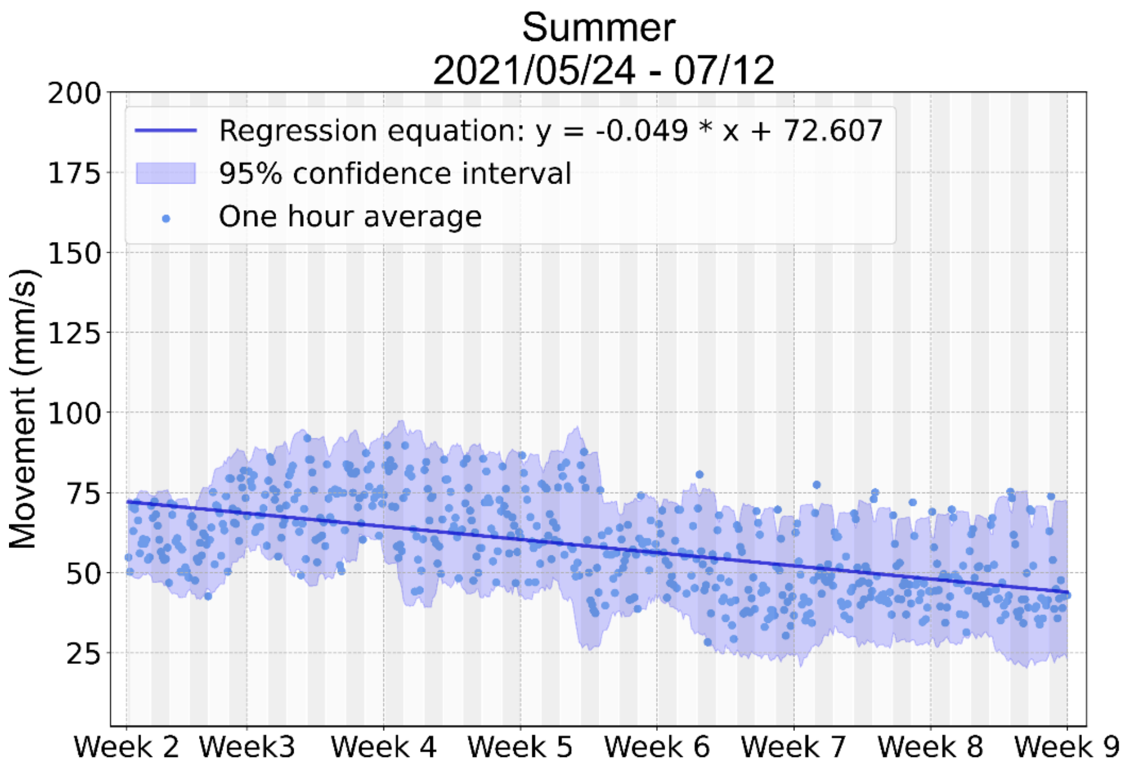
NNI = 1.413



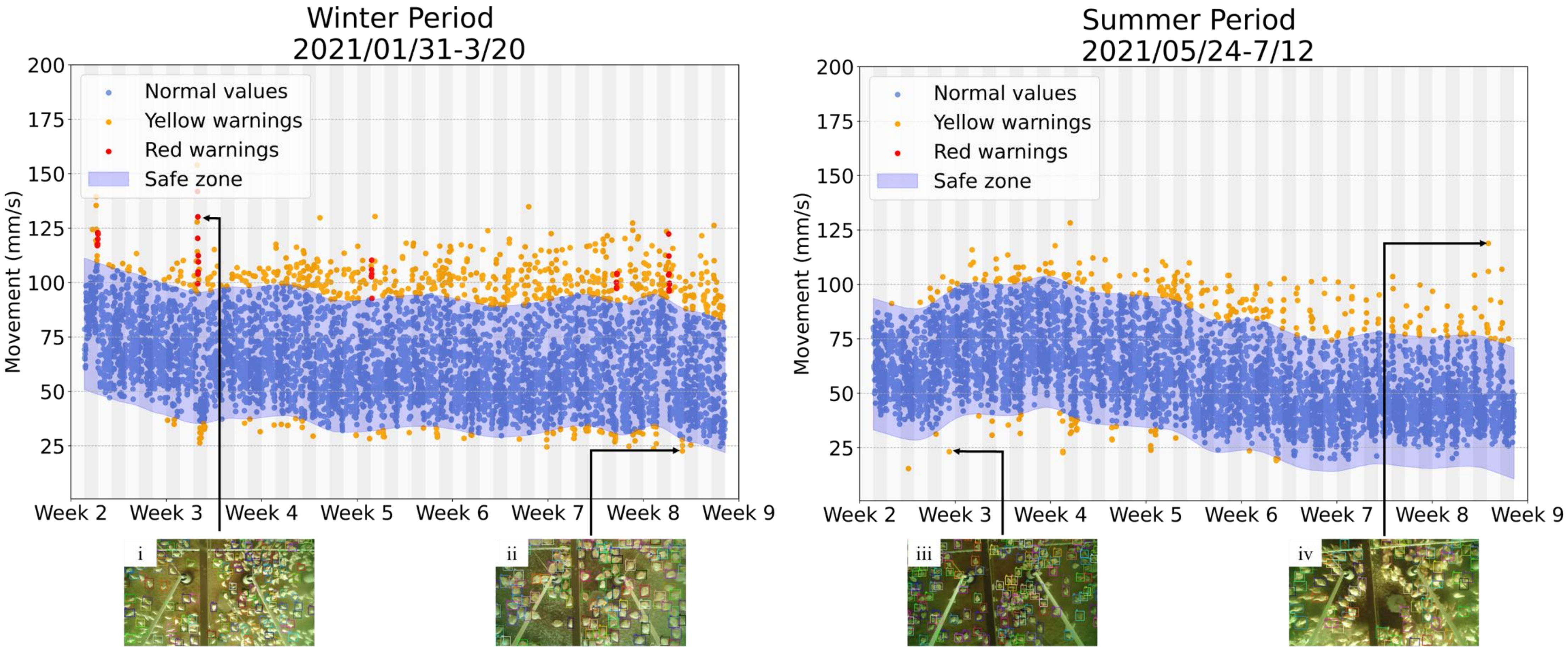
Clustered ⟷ Dispersed

# Long-term Movement Observation



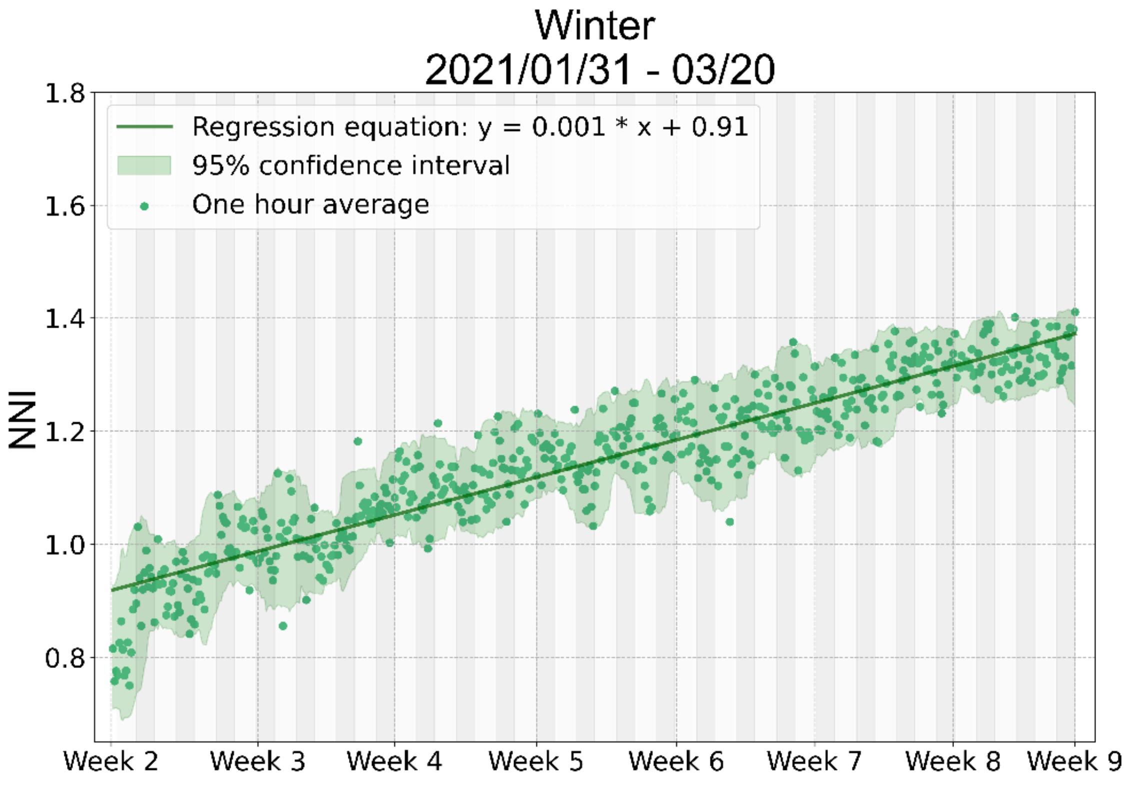



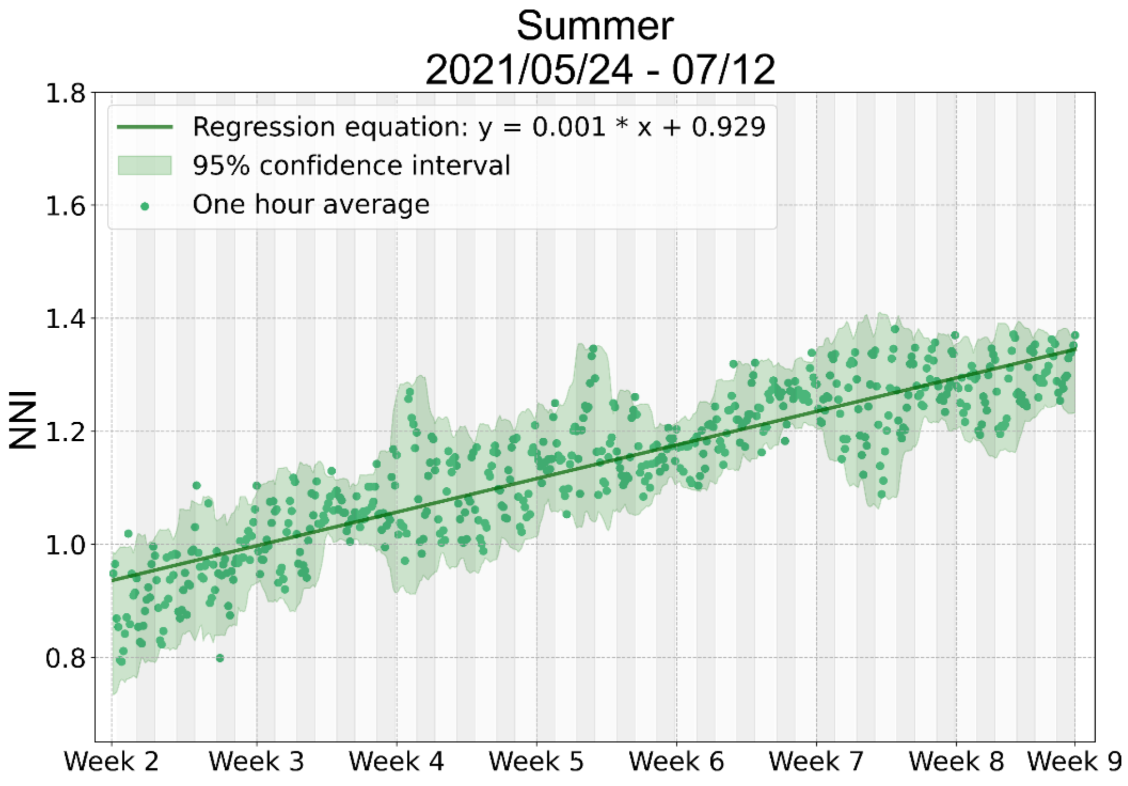
# Movement Warning System



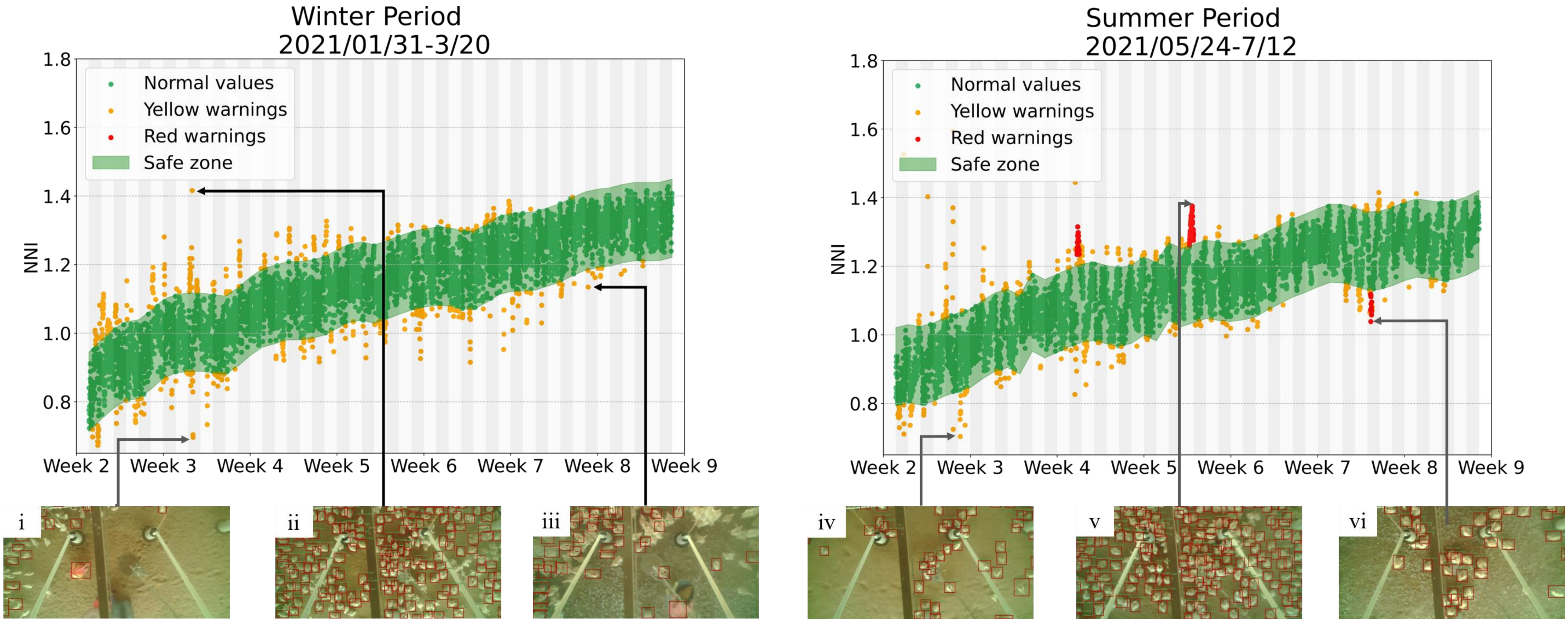

Yan-Fu Kuo
Dept. of Biomechatronics Engineering

# Long-term Dispersion Observation

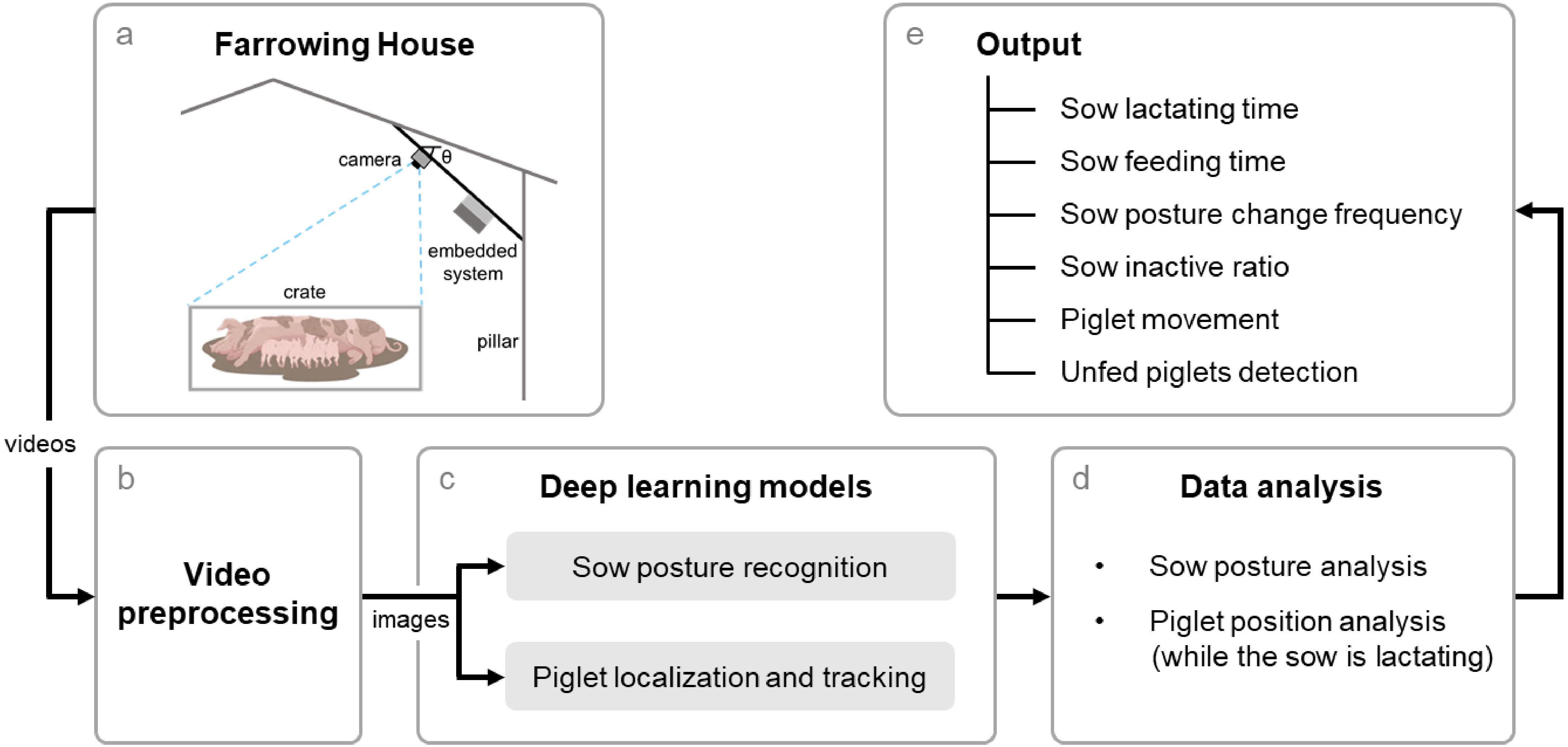


# Dispersion Warning System



# Sow and Piglet Activity Monitoring



# Sow Posture Recognition

Feeding

Standing

Sitting

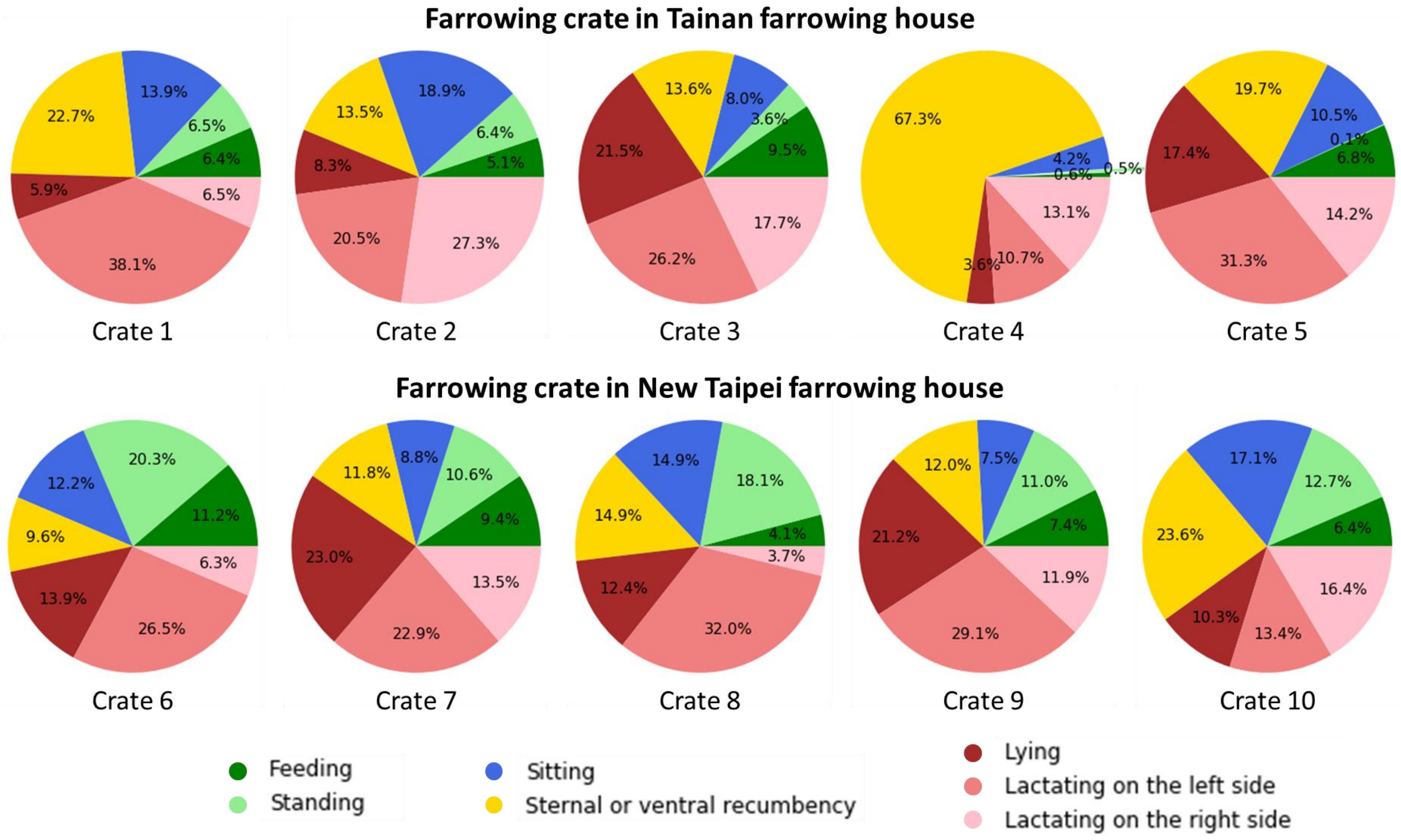
Recumbency

Lying

Lactating (right)

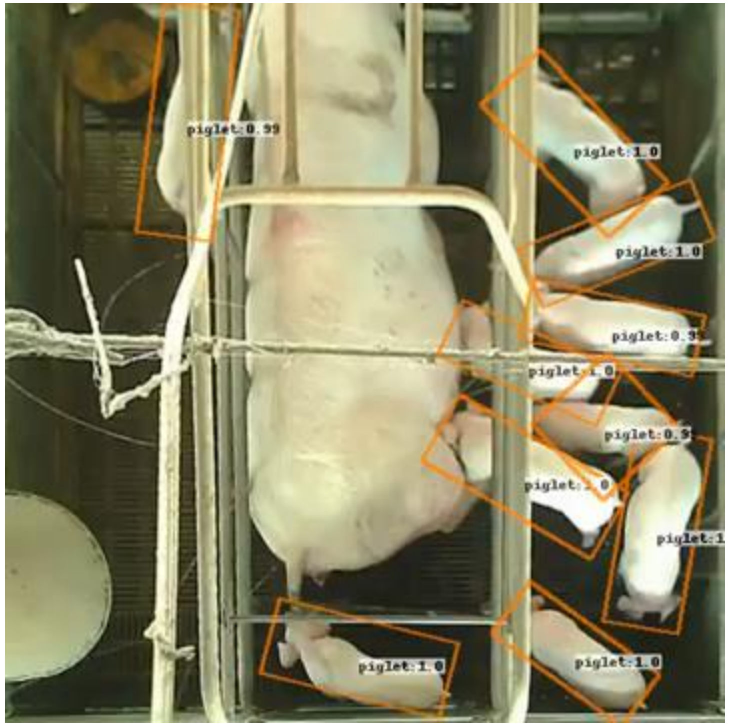
Lactating (left)

# Sow Posture Recognition

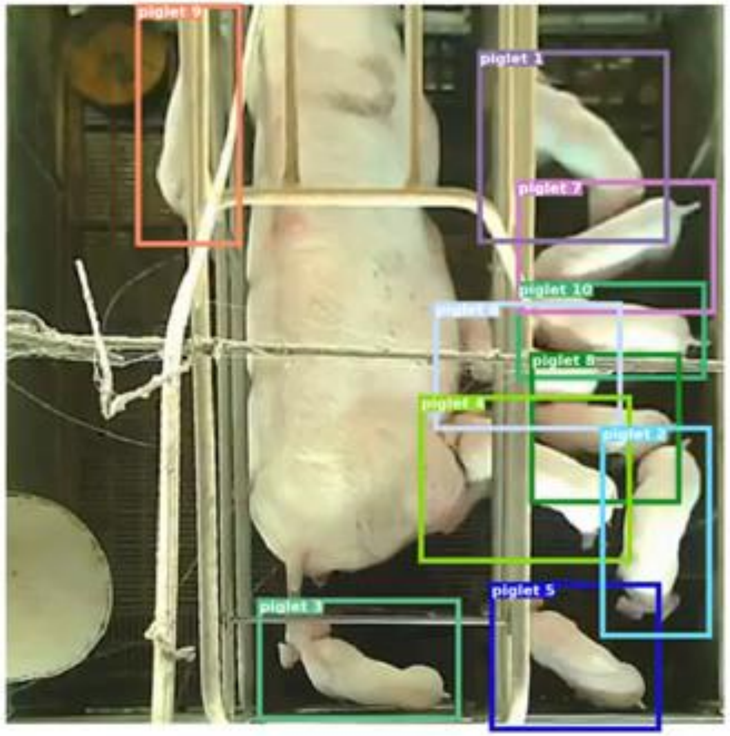


# Piglet Tracking

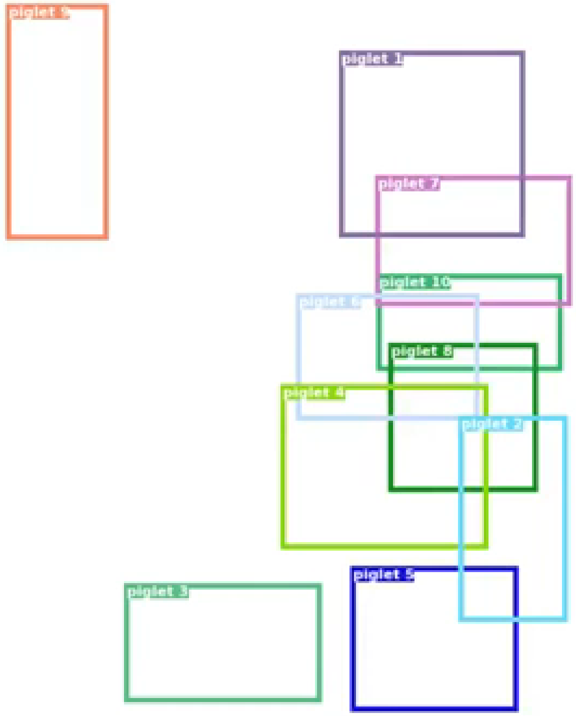
Localization




Tracking

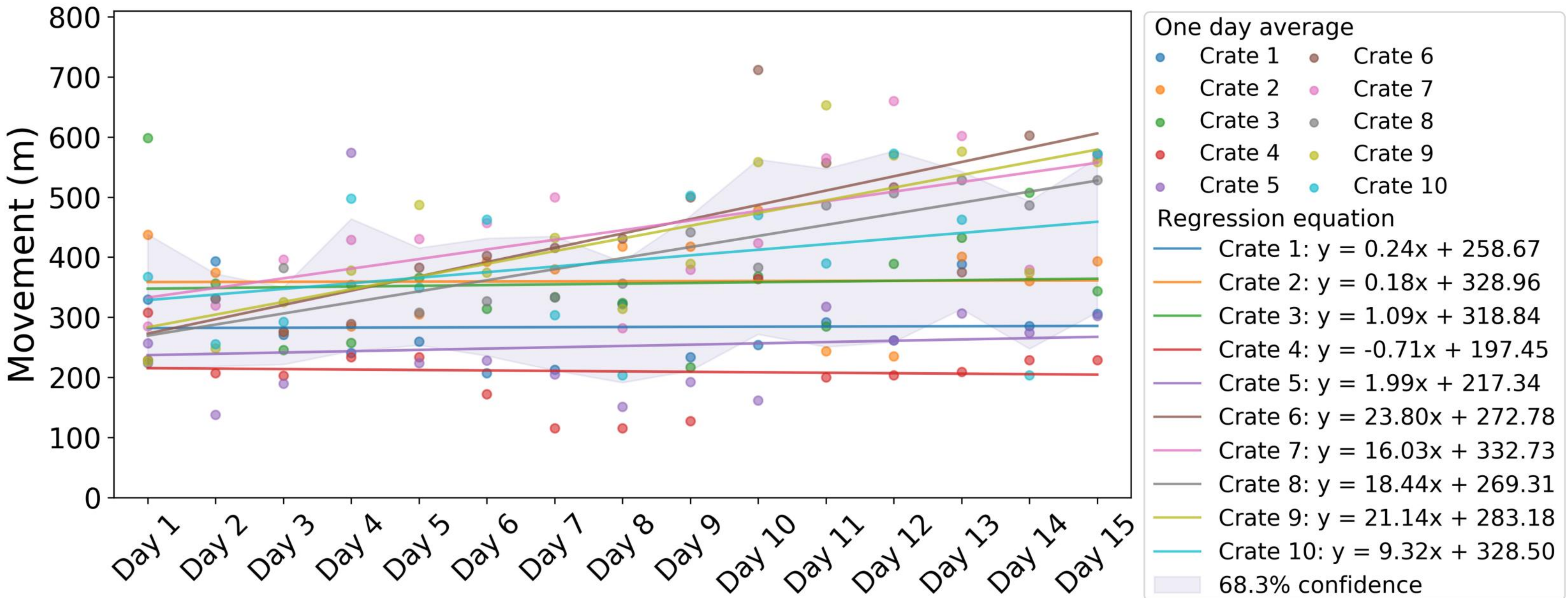



Trajectory

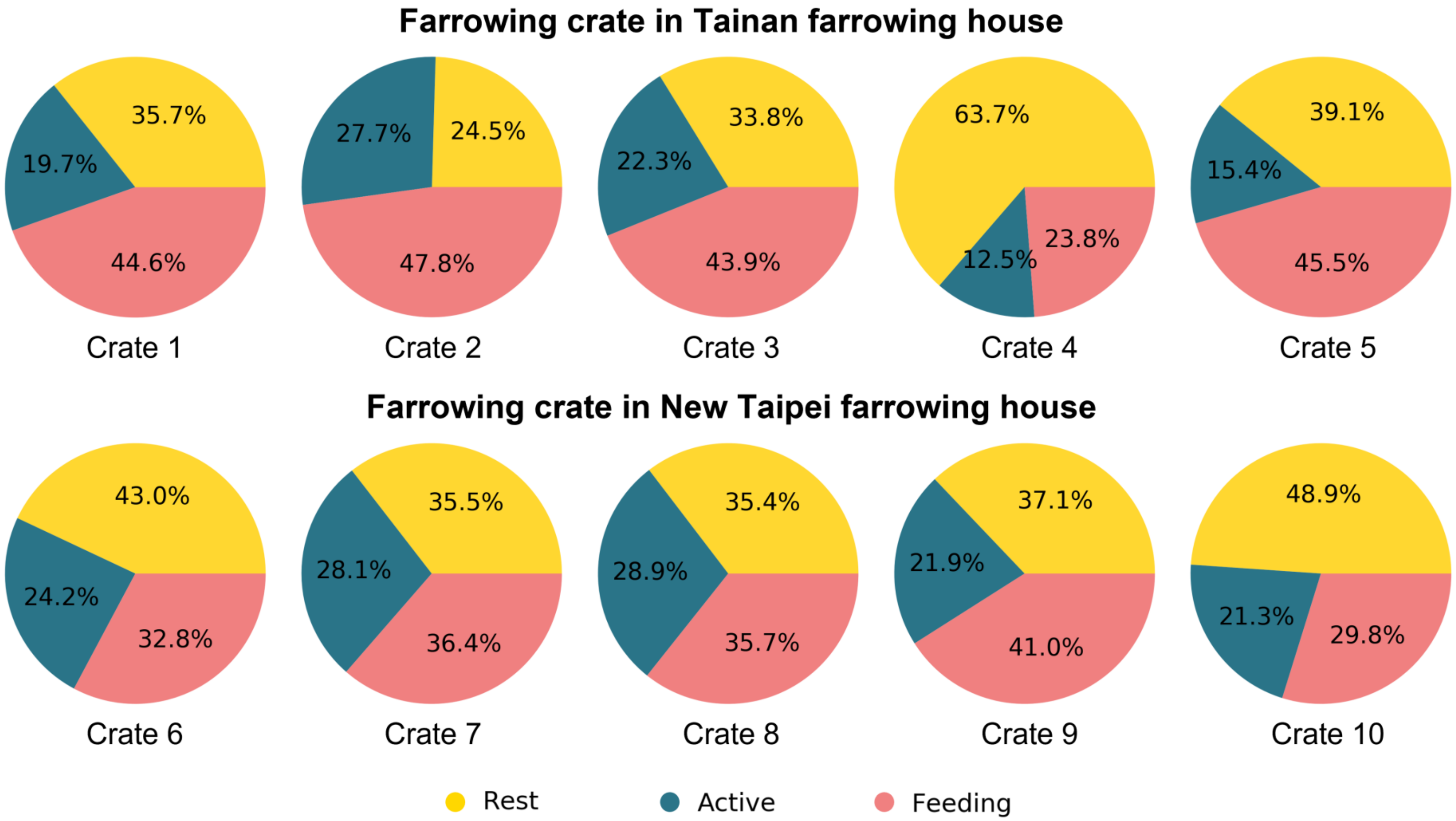



Yan-Fu Kuo
Dept. of Biomechatronics Engineering

# Piglet Activeness

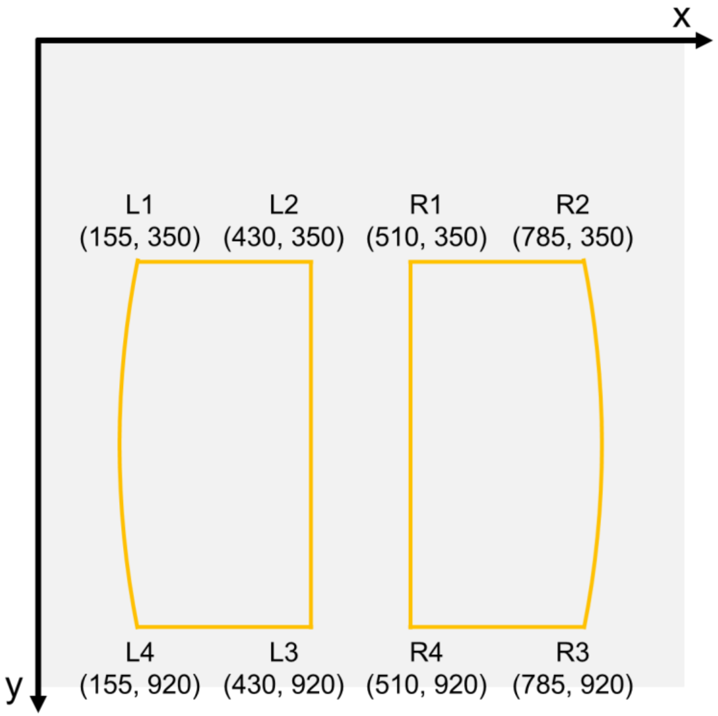


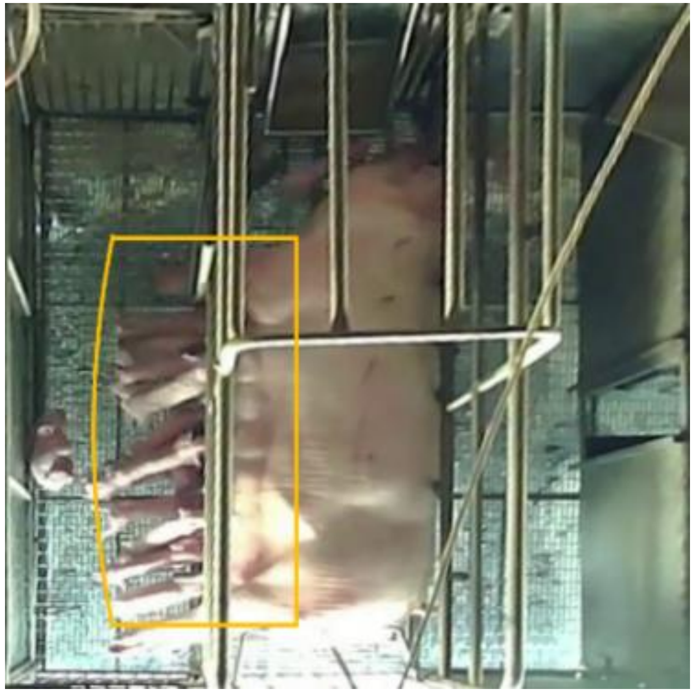
# Piglet Activities



# Unfed Piglet Detection

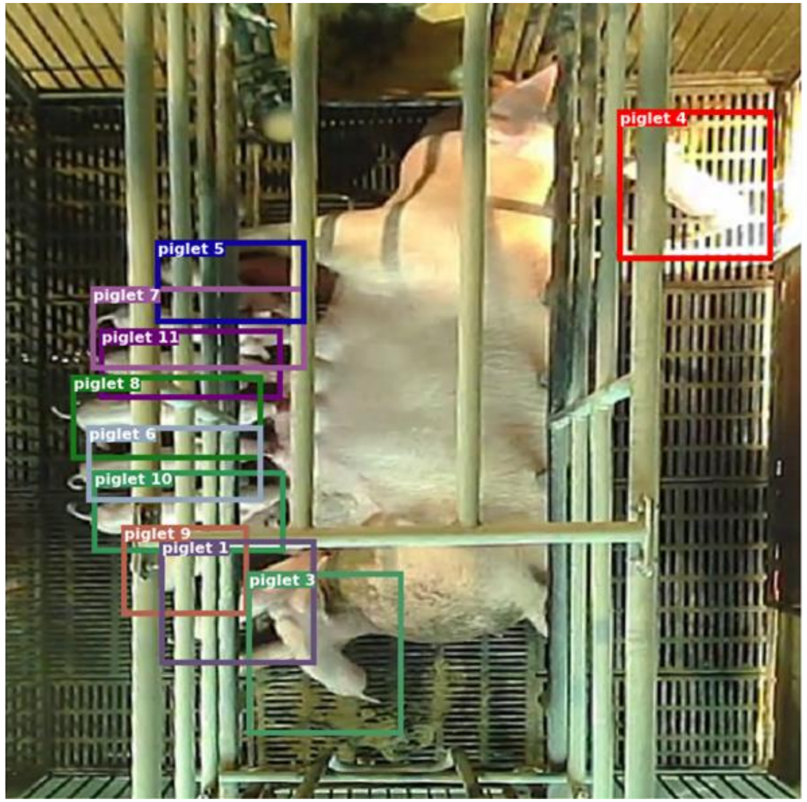
- Find unfed piglets by combining two models

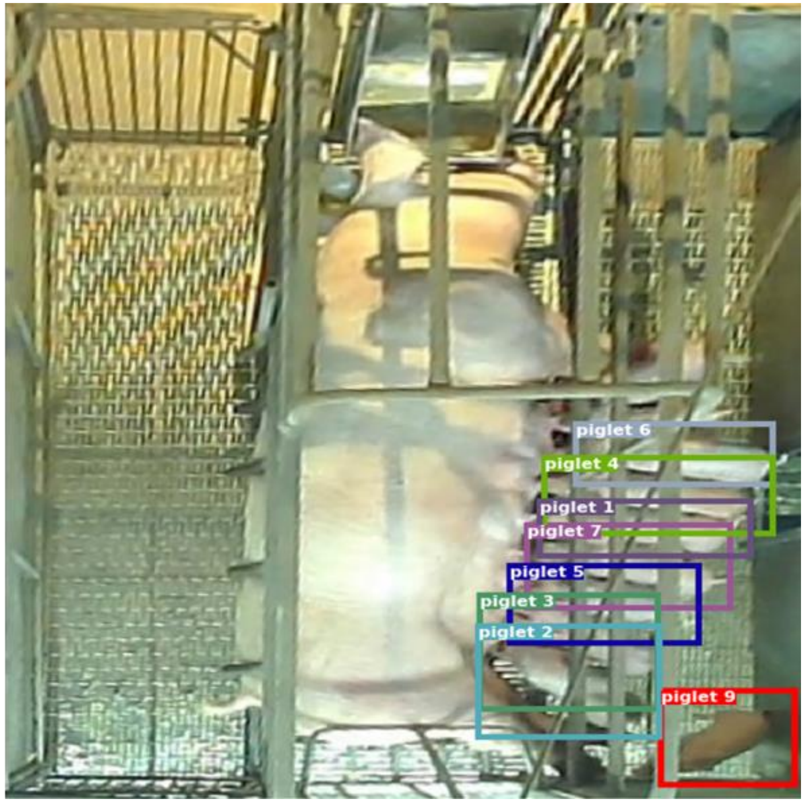


Yan-Fu Kuo
Dept. of Biomechatronics Engineering

# Unfed Piglets

# Acknowledgement

MOST 科技部
Ministry of Science and Technology

行政院農業委員會
COUNCIL OF AGRICULTURE, EXECUTIVE YUAN

行政院農業委員會
林務局

行政院農業委員會漁業署
FA·COA

立瑞畜產
LEADRAY
LIVESTOCK

MORNSUN

National Taiwan University
Biomechatronics Engineering

National Taiwan University X Council of Agriculture

# Thanks for listening

Yan-Fu Kuo | 27, June 2022