

AI4EVER, design, implementation and operations

Yang Hu



AI4EVER purpose

- Everyone can have their own AI models
- Integrate image label, AI model construction, training, and validation
- GUI for model construction and training parameter attune.
- Visualize published model architecture and connections

AI4EVER Design



ObjC

Swift

Graphical User Interface (GUI)

Data access

Model access/editing



Python

Neural Network computation

Model component

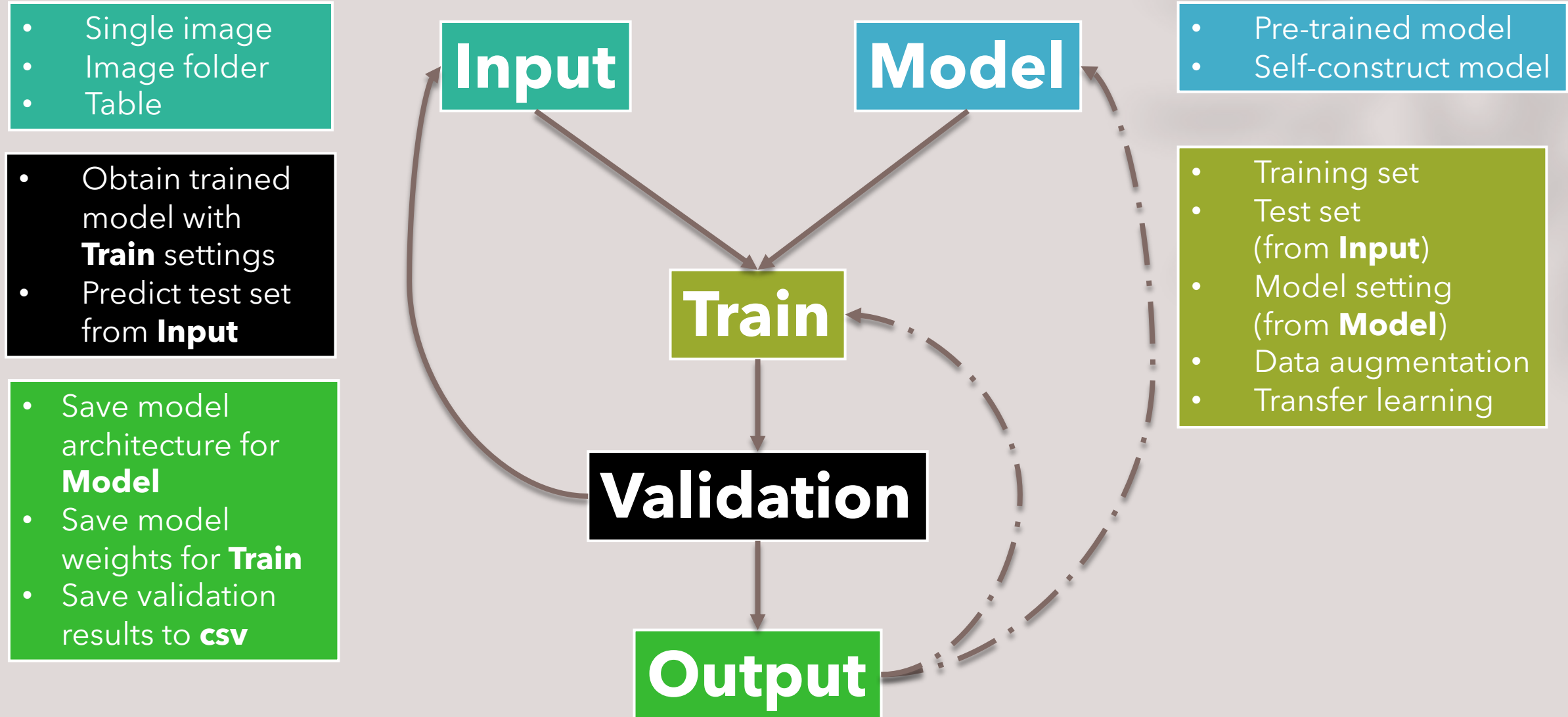
Data pre-process

Model training

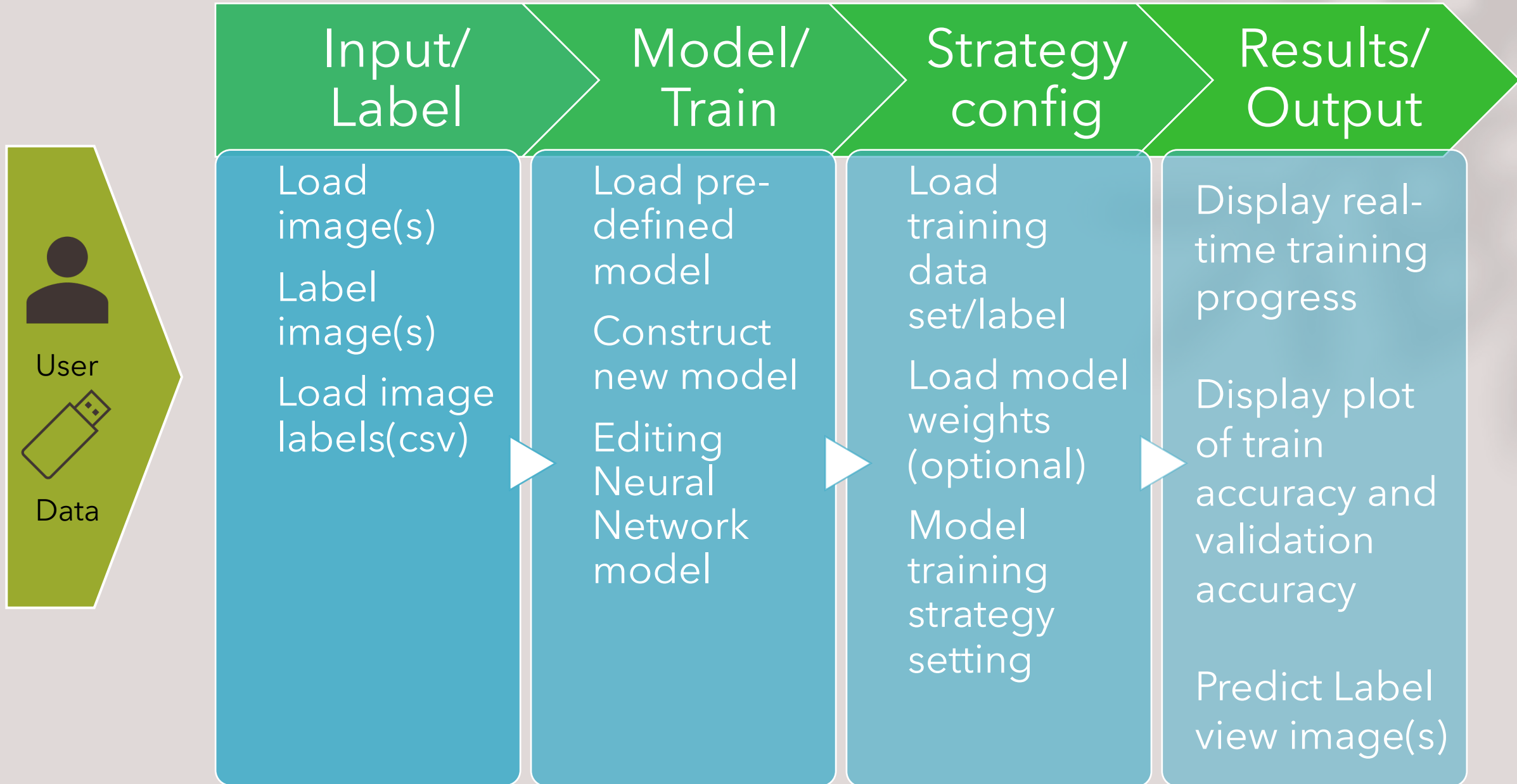


Keras

AI4EVER Architecture



AI4EVER Implementation



AI4EVER – All in one GUI

Input:

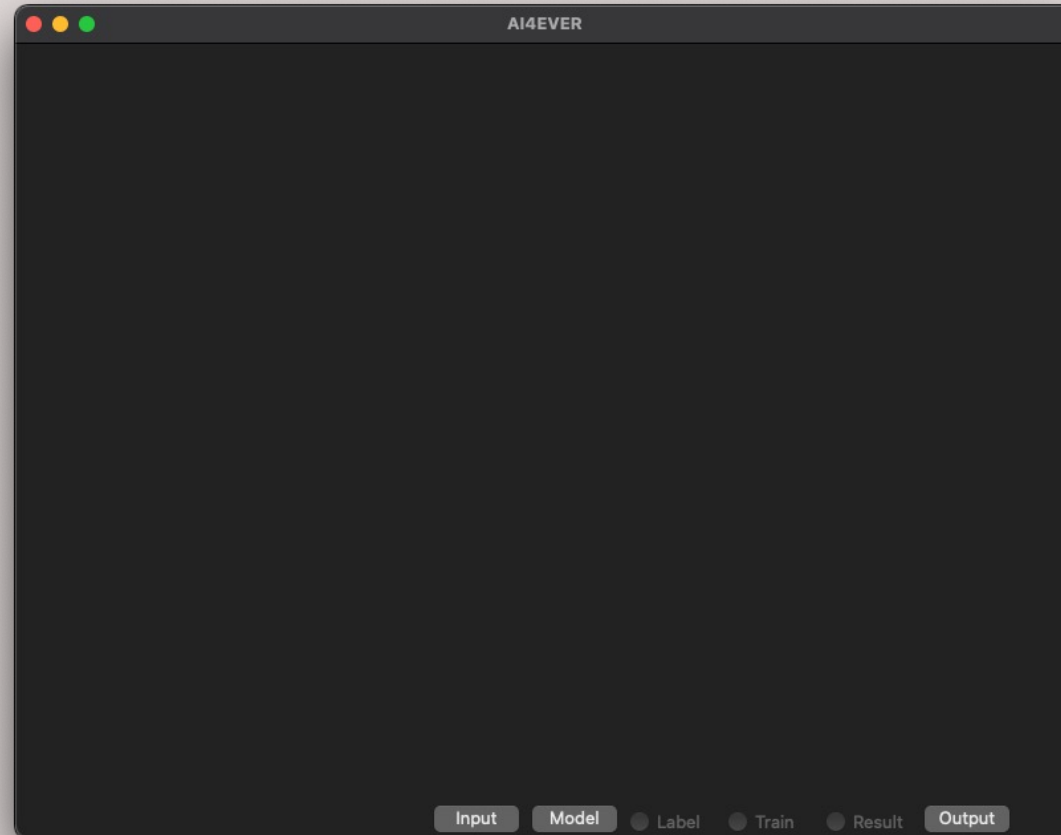
Load test dataset
image/image
folder/csv file
*cannot be empty

Model:

Load model (json,
pb,h5)
*empty option: need to
choose default models
or make new model

Label:

- Crop single image to grids
- Visualize images from folder
- Display table from csv
- Label images by mouse clicking



Result:

- Check real-time training progress
- Exam accuracy change of training/validation in all epochs

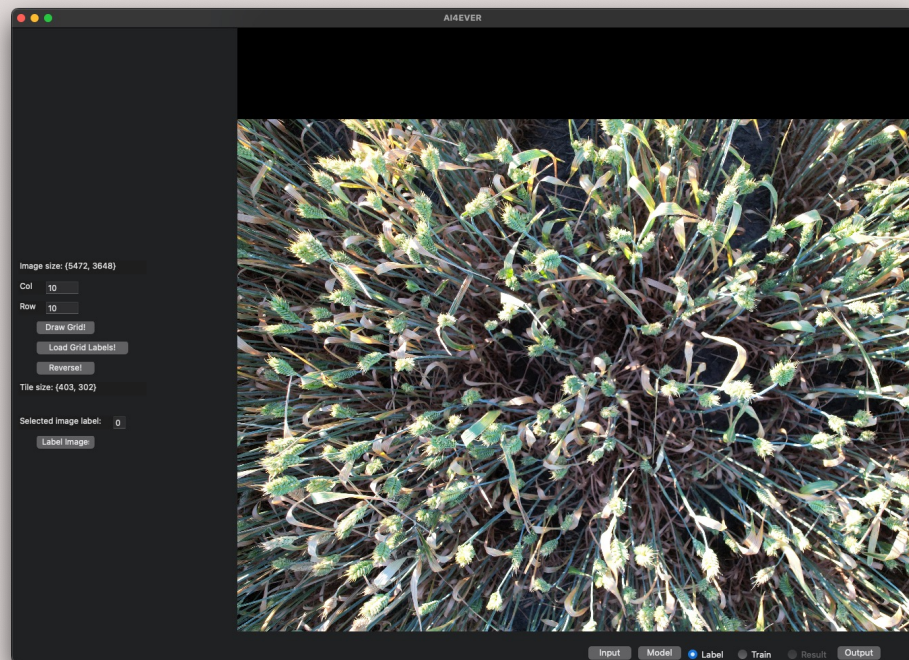
Output:

Select or create folder to save models, weights and predictions to disk (*Store model before load new image)
Or save labeled data as training dataset

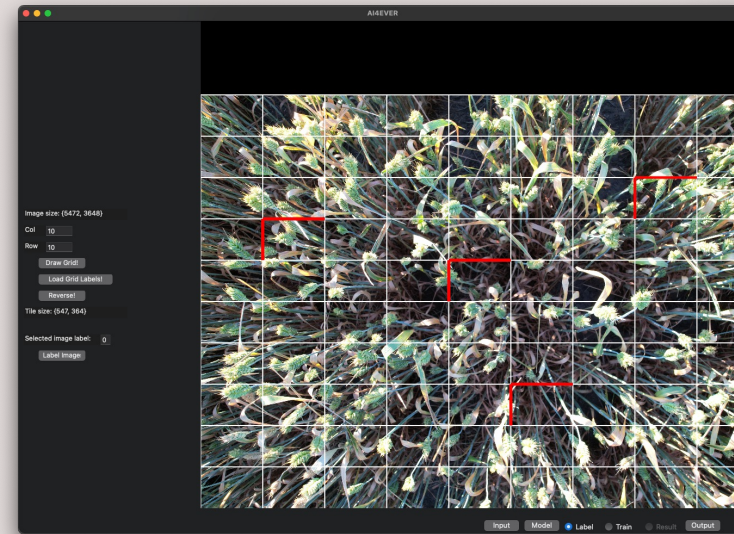
Train:

- Default model options
- Neural network layer options
- Strategy config to start model training setting

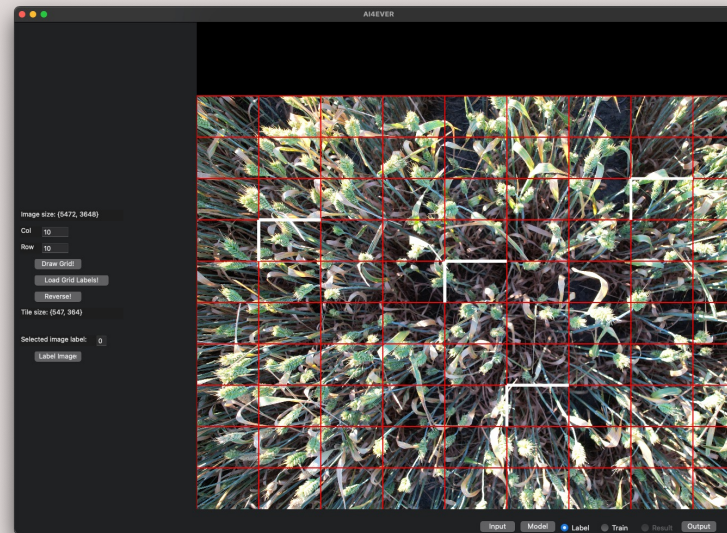
AI4EVER – Input



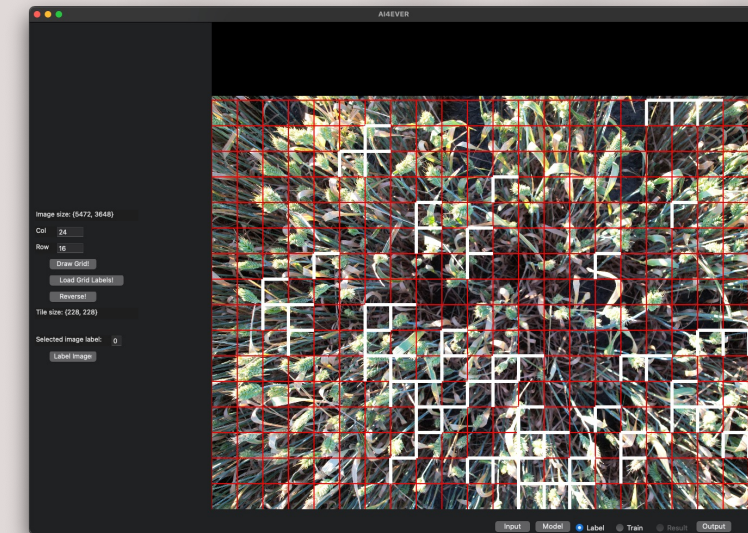
Click "Input" button load an image



Click "Draw Grid" and click the interested grids (white labeled as 0, red labeled as 1)

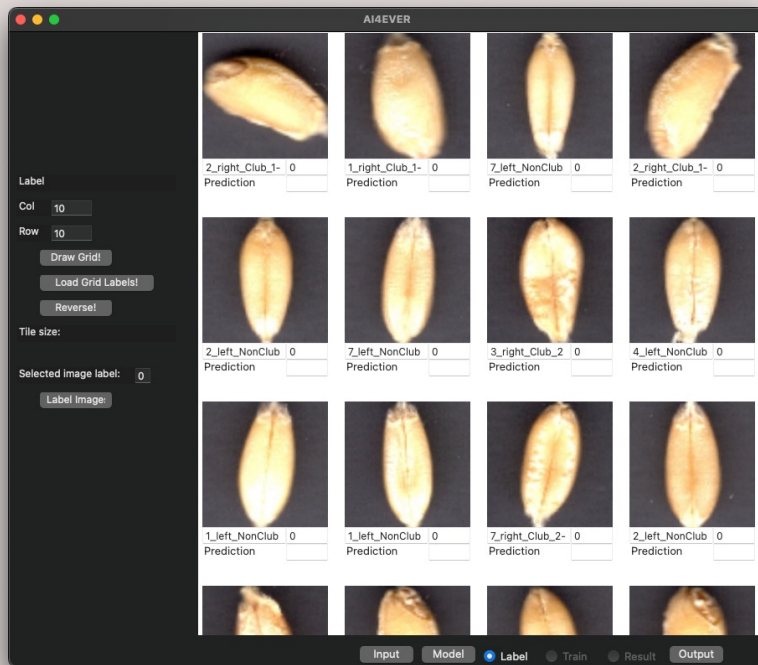


Click "Reverse!"
(Label value exchanged)

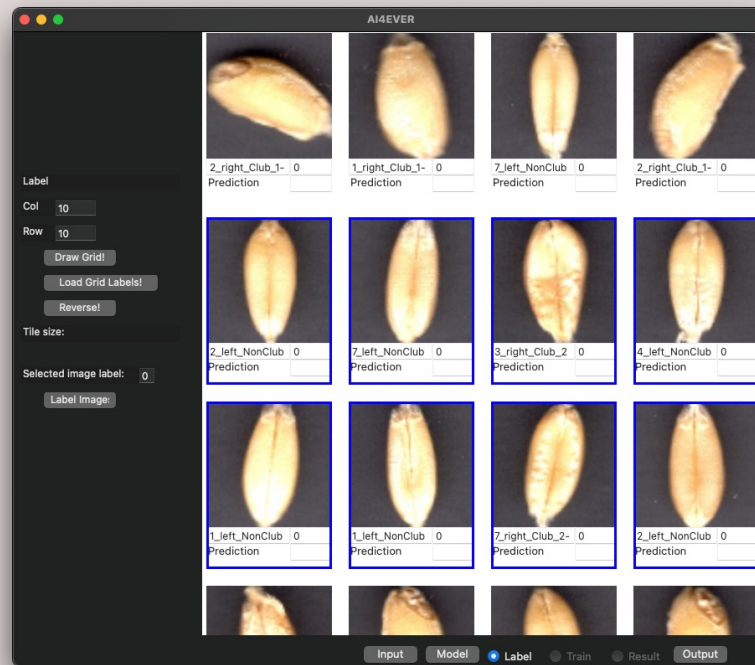


Click "Load Grid Labels!"
(load a csv label file)

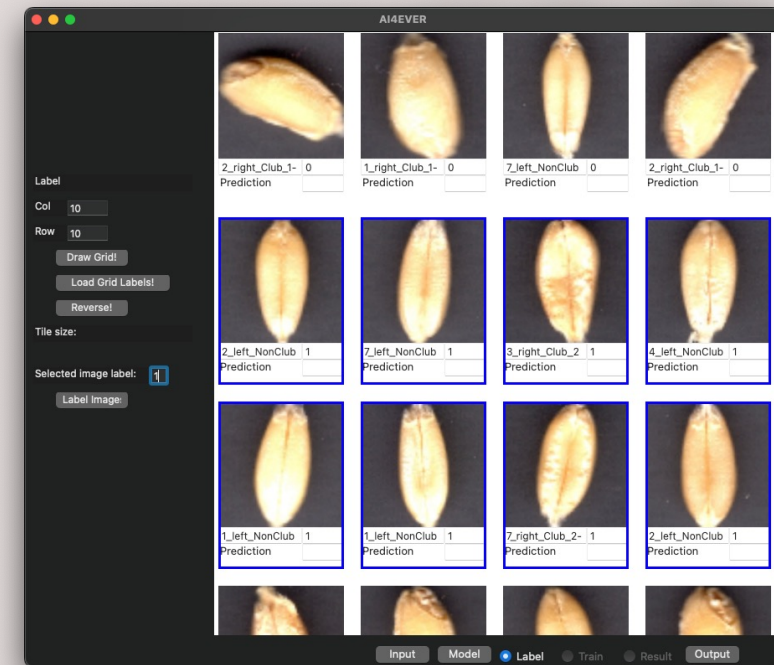
AI4EVER - Input



Click "Input" button load an image folder

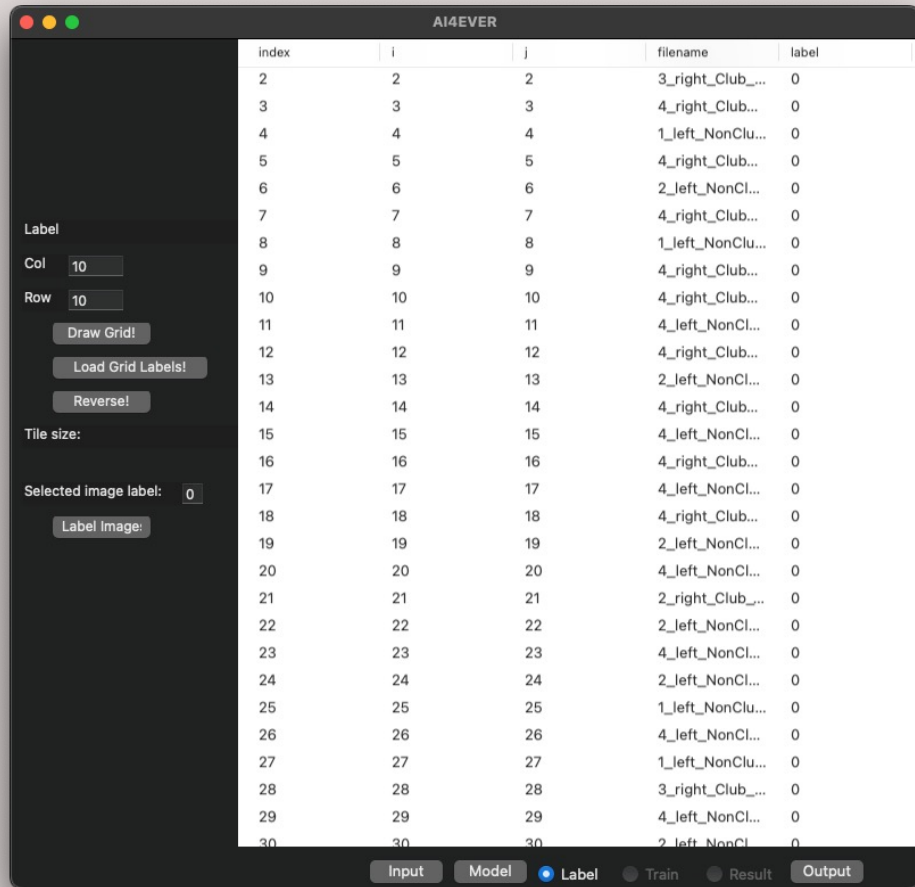


Drag the mouse to select multiple images

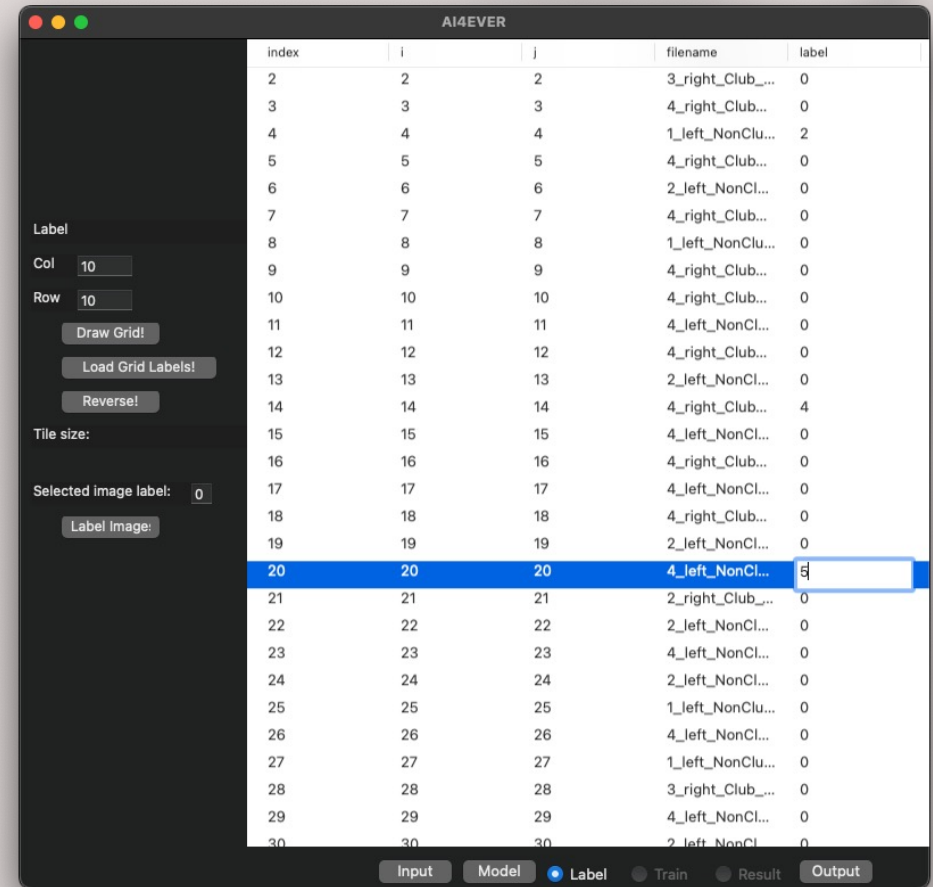


Type in label value to label the selected images and click "Label image"

AI4EVER - Input



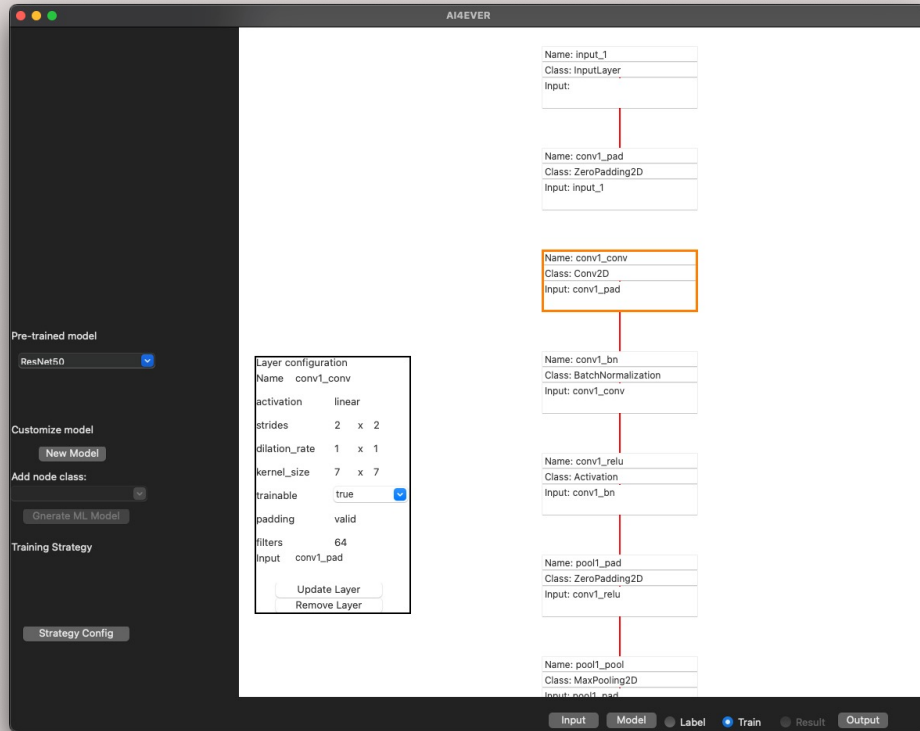
Click "Input" button load a csv document



Double click on cells to change values in the table

AI4EVER – Model Transfer Learning (Neural Networks / Deep learning)

AI4EVER pre-trained (ImageNet) neural-network models



Click "Model" click "OK" do not choose any model
Load pre-trained model "ResNet50"

Neural Network Model editing:

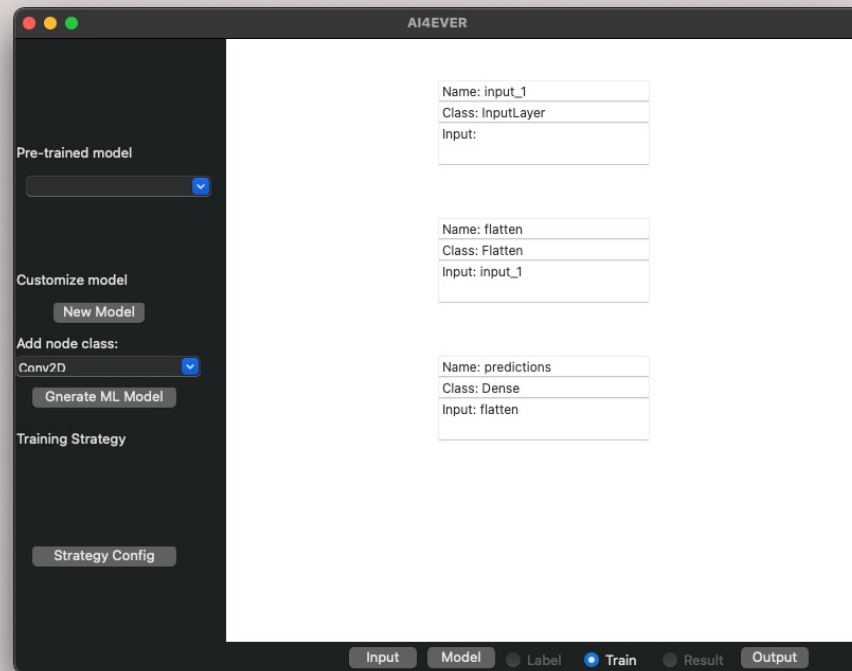
Mouse click on layer component to display layer configuration

*Pre-trained model layers cannot be removed

Model	Size (MB)	Top-1 Accuracy	Top-5 Accuracy	Parameters	Depth	Time (ms) per inference step (CPU)	Time (ms) per inference step (GPU)
Xception	88	79.0%	94.5%	22.9M	81	109.4	8.1
VGG16	528	71.3%	90.1%	138.4M	16	69.5	4.2
VGG19	549	71.3%	90.0%	143.7M	19	84.8	4.4
ResNet50	98	74.9%	92.1%	25.6M	107	58.2	4.6
EfficientNetB0	29	77.1%	93.3%	5.3M	132	46.0	4.9
EfficientNetB1	31	79.1%	94.4%	7.9M	186	60.2	5.6
EfficientNetB2	36	80.1%	94.9%	9.2M	186	80.8	6.5
EfficientNetB3	48	81.6%	95.7%	12.3M	210	140.0	8.8
EfficientNetB4	75	82.9%	96.4%	19.5M	258	308.3	15.1
EfficientNetB5	118	83.6%	96.7%	30.6M	312	579.2	25.3
EfficientNetB6	166	84.0%	96.8%	43.3M	360	958.1	40.4
EfficientNetB7	256	84.3%	97.0%	66.7M	438	1578.9	61.6

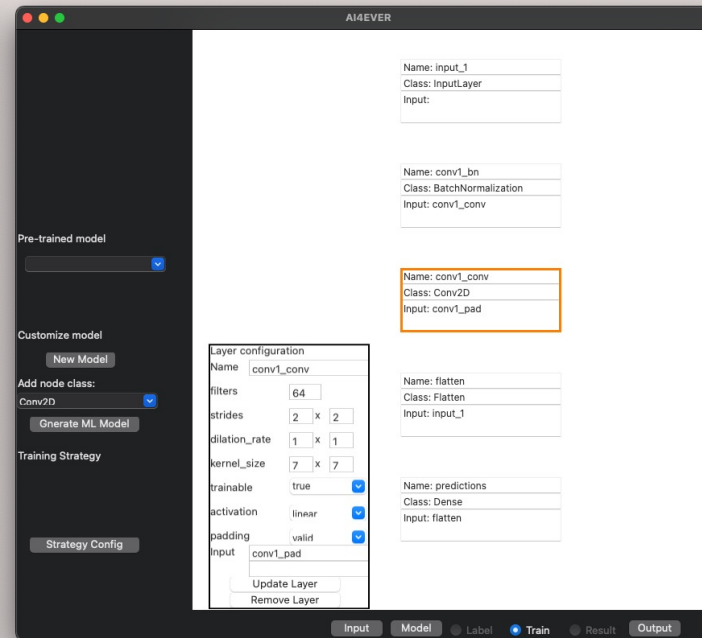
* Pre-trained models are from Keras and TensorFlow

AI4EVER – Model Construction



Click “New Model”

- Default three layers: input, flatten, dense(output)

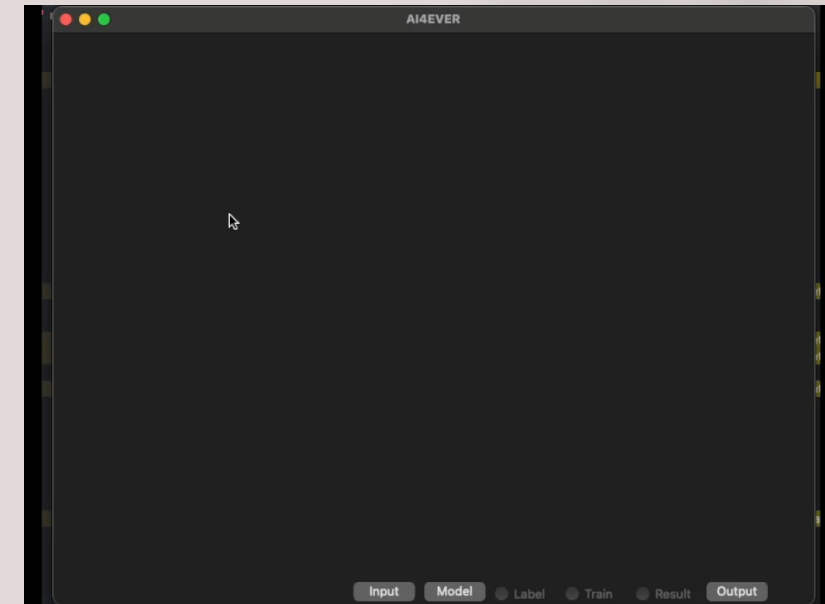


Select neural-network layers from “Add node class”

Click on layer component to trigger layer editor

*Drag layer component to change their positions

* Need to type layer names for input and output

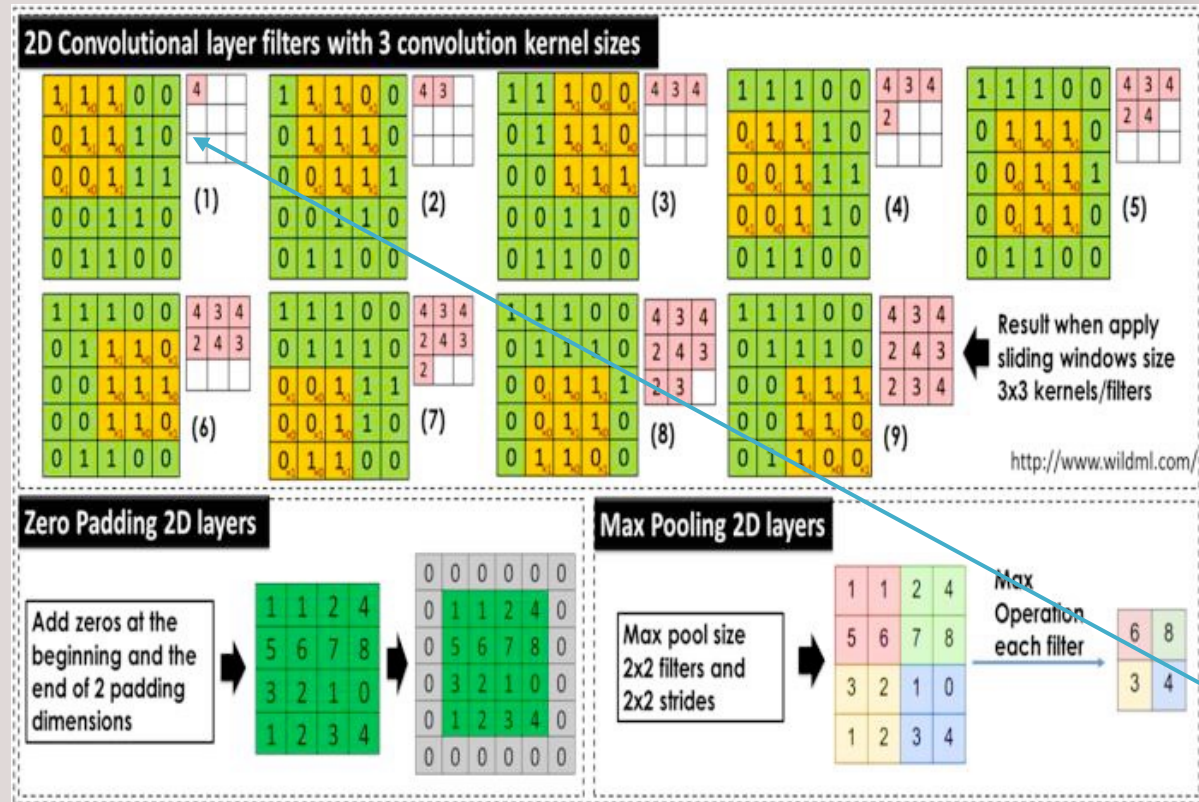


Screen-record of neural-network construction

AI4EVER – Model Construction

- Neural network layer operation supported by AI4EVER

- Conv2D
- ZeroPadding2D
- MaxPadding2D
- Batch Normalization
- Activation
- Flatten (before output layer)
- Dense (output layer)



Conv kernel

1	0	1
0	1	0
1	0	1

Apply

1x1	1x0	1x1
0x0	1x1	1x0
0x1	0x0	1x1

Sum

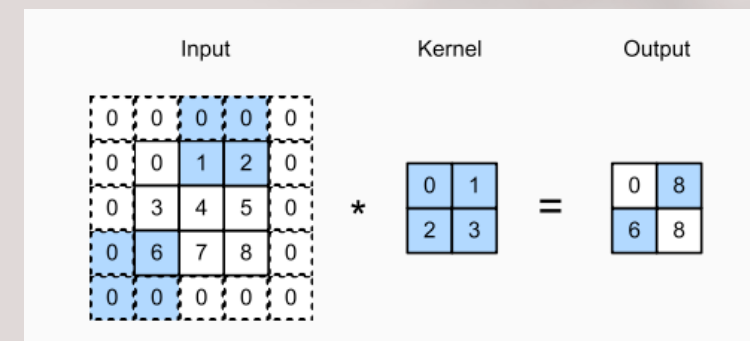
$$1+0+1+0+1+0+0+0+1=4$$

AI4EVER – Model Construction

- Conv2D layer editor

- Stride: height x width
- Kernel size: height x width
- Activation
 - Functions add non-linearity to neural-networks
- Padding
 - Valid: No padding
 - Same: use zero padding to keep layer size the same
- Dilation rate
 - Sparse conv kernels
- Filters
 - Number of kernels
- Trainable
 - True: update layer parameters during training
 - False: do not update layer parameters

The screenshot displays the 'Layer configuration' panel for a Conv2D layer named 'conv1_conv'. The configuration includes: strides (2 x 2), kernel_size (7 x 7), activation (linear), padding (valid), trainable (true), dilation_rate (1 x 1), and filters (64). The input is set to 'conv1_pad'. To the right, a list of layers shows 'conv1_conv' as the selected layer, with 'conv1_pad' as its input. Other layers include 'conv1_relu' (Activation) and another 'conv1_conv' (Conv2D) with input 'conv1_pad'.



Stride: 3x2
Kernel: 2x2

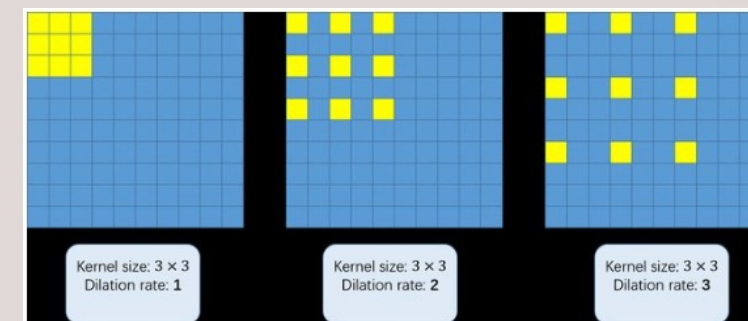
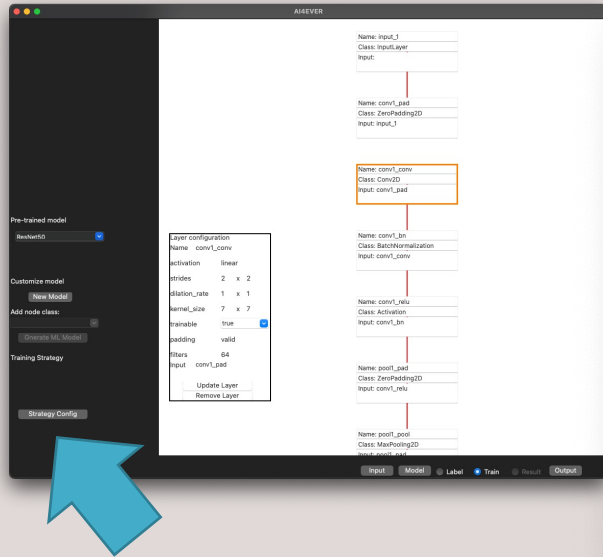


Image from: Li, Y., Zhang, X. and Chen, D., 2018. Csrnet: Dilated convolutional neural networks for understanding the highly congested scenes. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1091-1100).

AI4EVER - Training



Button to trigger training strategy

AI4
EVER

Training Strategy Panel

Test data

Training data

Training doc

Weights

Batch size

4

Training Epoch

3

Optimizer

Adam

Loss Function

mean_squared_error

Adjust Training & Validation Ratio

Start!

Script!

Cancel

AI4
EVER

Training Strategy Panel

Train Ratio

0.60

Valid Ratio

0.40

-----Image Data Augmentation-----

☐ Image Transfer (up,down,left,right)

☐ Image Random Crops x 4

☐ Image Rotate in degrees x 11

☐ Image distorte x 4

Start!

Script!

Cancel

AI4
EVER

Training Strategy Panel

☐ Image Rotate in degrees x 11

☐ Image distorte x 4

-----Transfer learning (TF) with extra layers-----

☐ TF fine-tuning (to activate all the following settings)

☐ Freeze model

Dropout rate

0.2

Predict Class

2

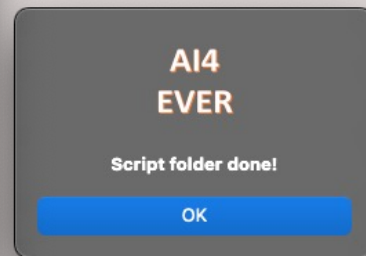
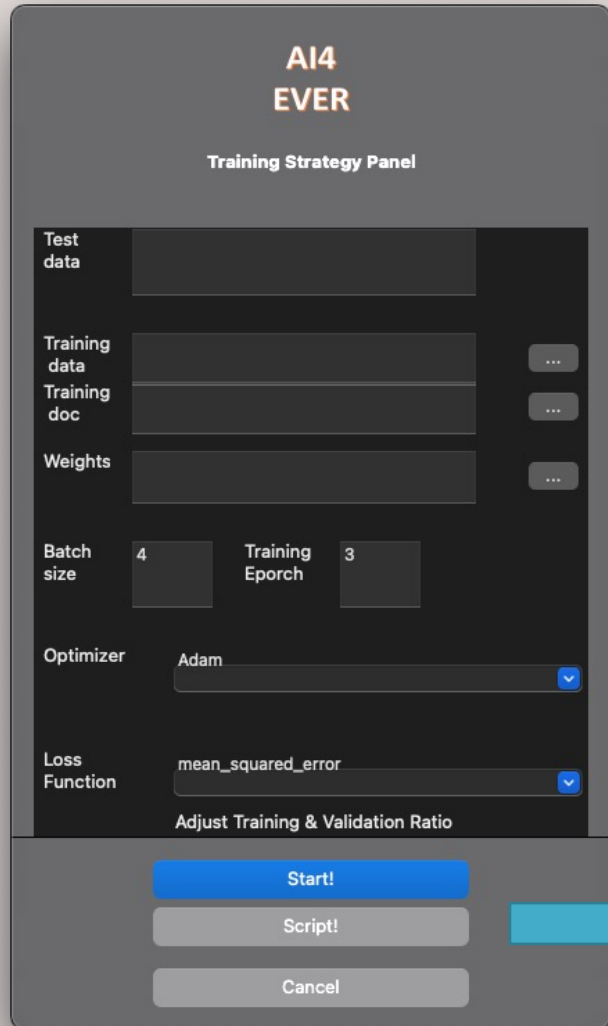
Start!

Script!

Cancel

- Start: To train at the background
- Script: To generate command line script and all files to run on server or other computer

AI4EVER - Training



Test dataset

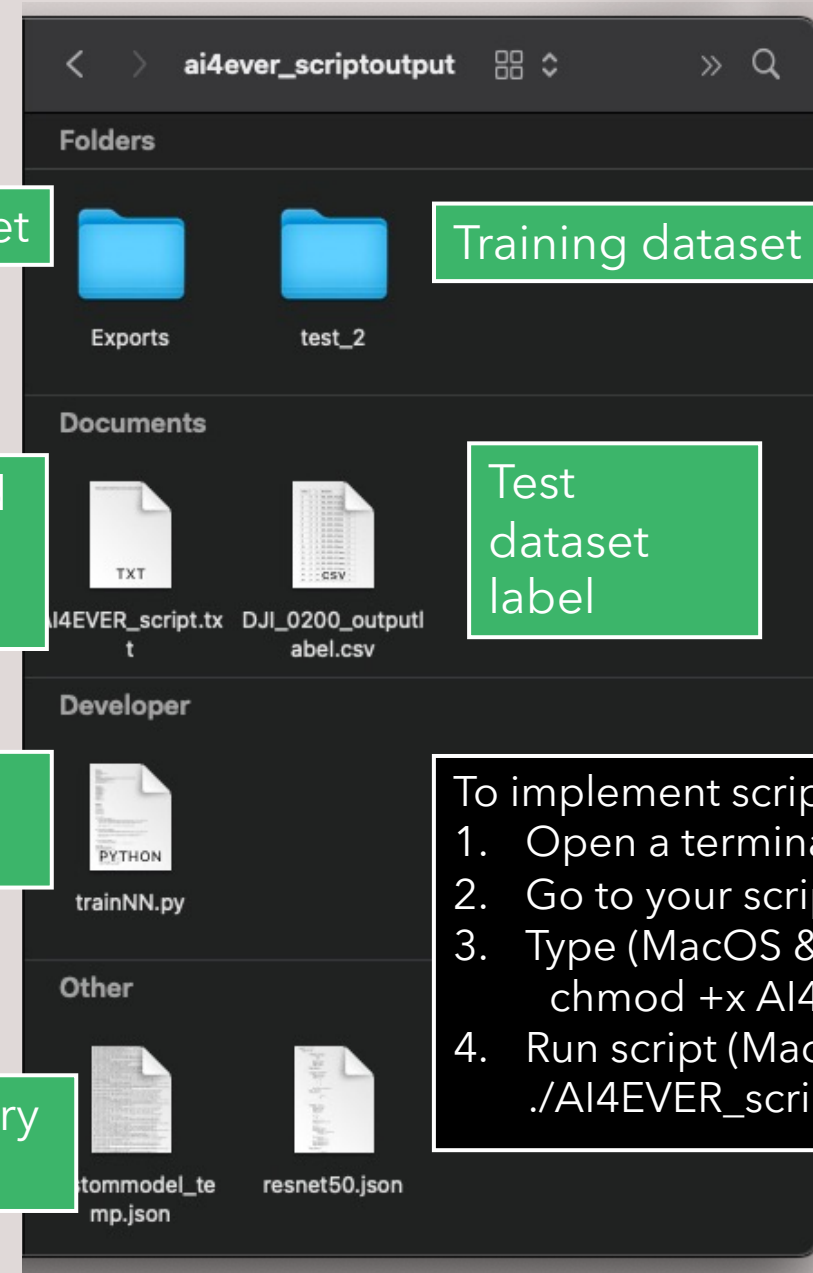
Training dataset

Command
Line script
file

Test
dataset
label

Python
package

Temporary
model

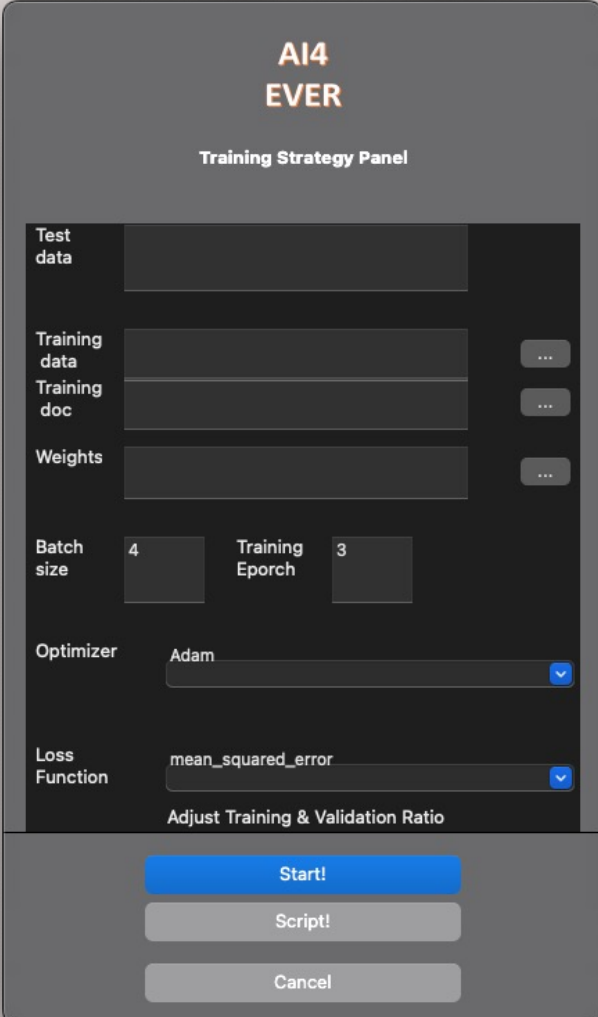


To implement script:

1. Open a terminal
2. Go to your script output folder
3. Type (MacOS & Linux):
`chmod +x AI4EVER_script.txt`
4. Run script (MacOS & Linux):
`./AI4EVER_script.txt`

AI4EVER - Training

- Test data: Path of input data
- Training data: Select a folder with your training dataset
- Training doc: Select your label file (*.csv) for your training dataset
- Weights: neural-network parameters (*.hdf5) could be empty
- Batch size: number of images in a batch
- Training epoch: number of iterations of model training
- Optimizer: Gradient method to find the minimum errors
- Loss function: Error measurement

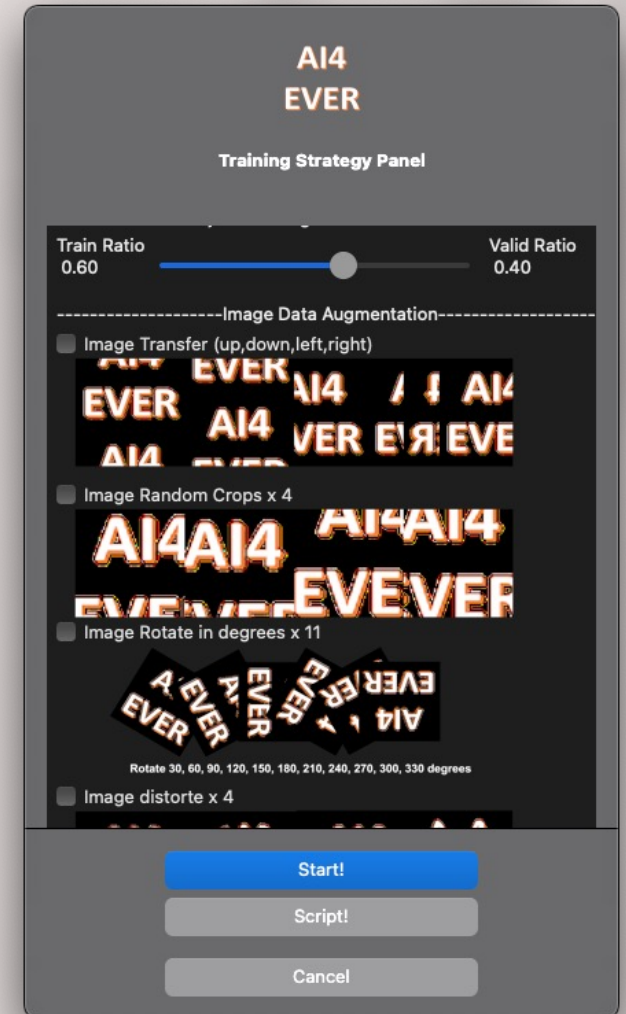


The image shows a software interface titled "AI4EVER" with a subtitle "Training Strategy Panel". The interface is dark-themed and contains several input fields and buttons. At the top, the title "AI4EVER" is in orange, and "Training Strategy Panel" is in white. Below the title, there are four rows of input fields: "Test data", "Training data", "Training doc", and "Weights". Each of these fields has a corresponding button with three dots (a file selector icon) to its right. Below these fields, there are two more input fields: "Batch size" with the value "4" and "Training Epoch" with the value "3". Below these, there is a dropdown menu for "Optimizer" with "Adam" selected. Below that, there is a dropdown menu for "Loss Function" with "mean_squared_error" selected. At the bottom of the panel, there are three buttons: "Start!" (blue), "Script!" (gray), and "Cancel" (gray). The text "Adjust Training & Validation Ratio" is visible below the "Loss Function" dropdown.

AI4EVER	
Training Strategy Panel	
Test data	
Training data	...
Training doc	...
Weights	...
Batch size	4
Training Epoch	3
Optimizer	Adam
Loss Function	mean_squared_error
Adjust Training & Validation Ratio	
Start!	
Script!	
Cancel	

AI4EVER - Training

- Train/Valid Ratio:
 - Proportion of training and validation of training dataset
 - E.g. 1,000 images, Train ratio 60%, Valid ratio 40%, then 600 will be used for training, and 400 will be used for validation after each epoch. Training model cannot access to validation data
- Image augmentation
 - To produce more images for training dataset for neural networks training
 - Image transfer: produce x 4 per image by transfer image content
 - Random Crops: produce x 4 per image by randomly cut image content
 - Rotation: produce x11 per image by image rotation of every 30 degrees
 - Distortion: produce x 4 per image by change image content shapes



AI4EVER - Training

- Transfer learning with extra layers
 - Automatically add:
 - A globalaverage2D layer
 - A dropout layer
 - A flatten layer
 - A dense layer (output = predict class)
- Freeze model
 - Chceked: Only update parameters to the extra layers
 - Unchecked: Update parameters of all layers
- Dropout rate (%)
 - To randomly remove parameters of % from dropout layer
 - Dropout layer is common for EfficientNets



The image shows a screenshot of the 'AI4EVER Training Strategy Panel'. At the top, the text 'AI4 EVER' is displayed in orange. Below it, the title 'Training Strategy Panel' is shown. The main area contains two image augmentation settings: 'Image Rotate in degrees x 11' with a preview of rotated 'AI4 EVER' text, and 'Image distorte x 4' with a preview of distorted 'AI4 EVER' text. Below these, a dashed line separates the settings from the 'Transfer learning (TF) with extra layers' section. This section includes checkboxes for 'TF fine-tuning (to activate all the following settings)' and 'Freeze model'. At the bottom, there are three buttons: 'Start!' (blue), 'Script!' (grey), and 'Cancel' (grey). The 'Dropout rate' is set to 0.2 and the 'Predict Class' is set to 2.

AI4
EVER

Training Strategy Panel

Image Rotate in degrees x 11

Rotate 30, 60, 90, 120, 150, 180, 210, 240, 270, 300, 330 degrees

Image distorte x 4

-----Transfer learning (TF) with extra layers-----

☐ TF fine-tuning (to activate all the following settings)

☐ Freeze model

Dropout rate 0.2

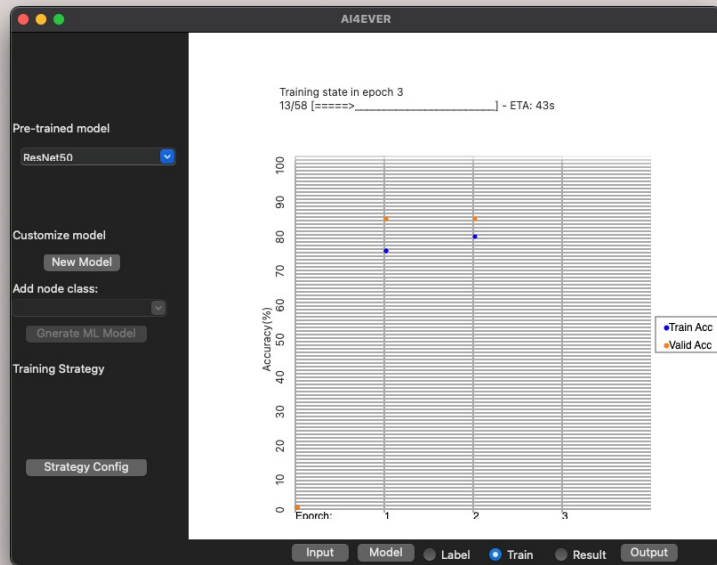
Predict Class 2

Start!

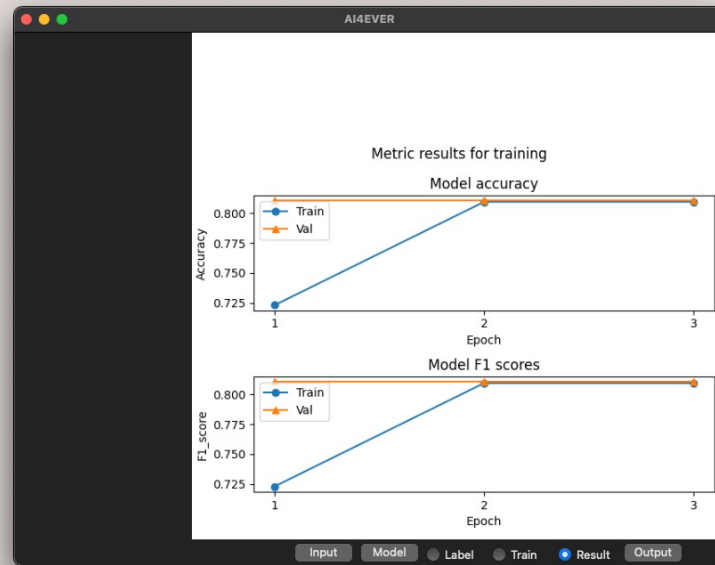
Script!

Cancel

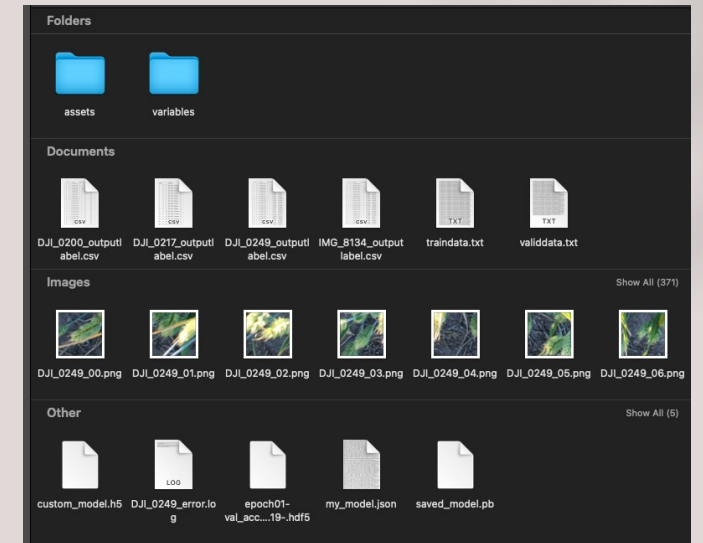
AI4EVER – Results



Real-time training progress monitor



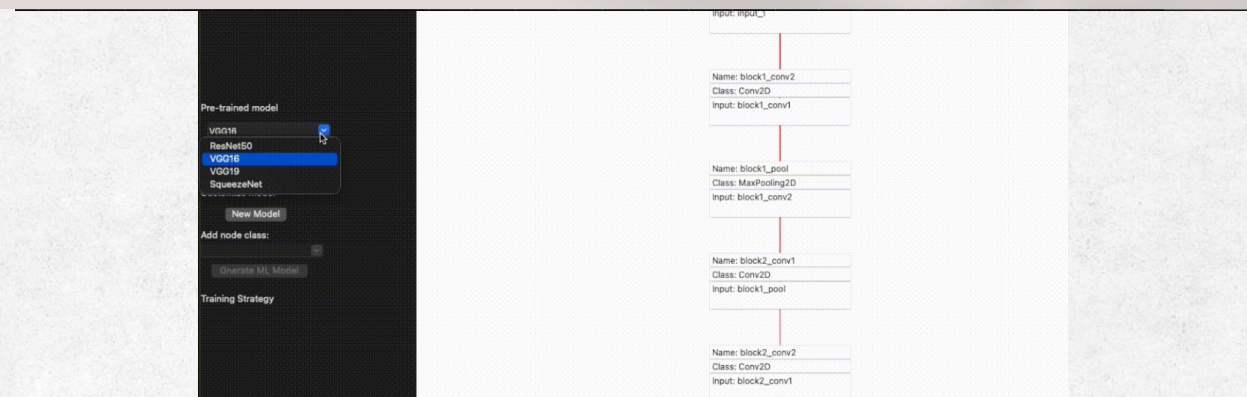
All epoch training results plots



After training output

- Images & labels
- my_model.json: Neural network model
- *.hdf5: model weight

AI4EVER link



Neural networks (NN) are a subset of machine learning that has attracted great attention. However, the prerequisites of computer coding skills cause the resources for learning NN to be unequally distributed, with interested learners who do not have the background underrepresented in the subject. We propose AI4EVER to fill the gap using a hybrid of an interactive online learning environment and offline software that teaches and assists the underrepresented individuals to construct machine learning programs from scratch with no computer programming required. The learning environment has ten projects from a range of different topics that users can choose to learn. Each project has an interactive teaching module that determines if learners perform all the necessary steps and gives hints on how to proceed. Existing online resources supporting neural network customization usually have video illustrations but lack interactive learning processes. Users need to upload their data to the website, which causes data security concerns. Offline resources support users using existing models and have little support for users to construct a machine learning program from scratch. Data labeling is essential for the success of machine learning programs, but existing resources do not facilitate the process. If this project is successful, it will assist learners in understanding machine learning workflow and allow them to construct machine learning programs by drag-and-drop without prerequisite coding skills.

To perform the Neural Network Computation, please download Python 3.8.5 and the "installpkg(Mac)" shown below for the necessary Python libraries.

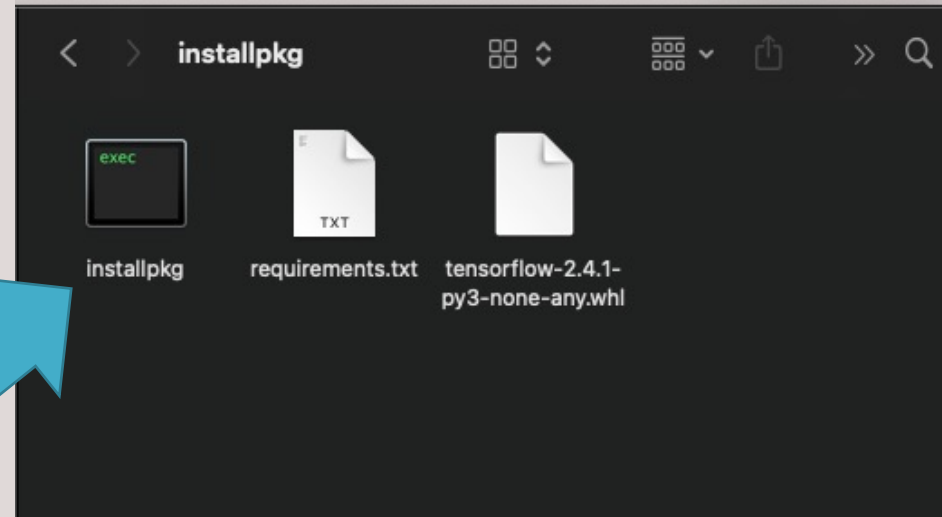
- [Download latest version](#)
- [Old Versions](#)
- [Download installpkg\(Mac\)](#)

<https://zzlab.net/software/>

<https://zzlab.net/AI4EVER/>

AI4EVER Installation

- Mac system
 - Install Python 3.8.5
 - Download installpkg(Mac)
Double click to install the packages



- Download the latest version AI4EVER

AI4EVER Demo



Five ingredients to succeed: CS-VMV

Culture: Trying to understand.

Strategy: Solve biological problems with analytical & computational challenges.

Vision: Genomic & phenomic stream data is stationary water for organisms.

Mission: You get data, we help with our analytical methods, tools, & expertise.

Value: Every idea makes sense.

Thank you! Any questions?