

AniLength

version 1.2

User Manual

Last update: 6/21/2023

Contact: Sang-Kyu Jung (skjung@hongik.ac.kr)

System requirements

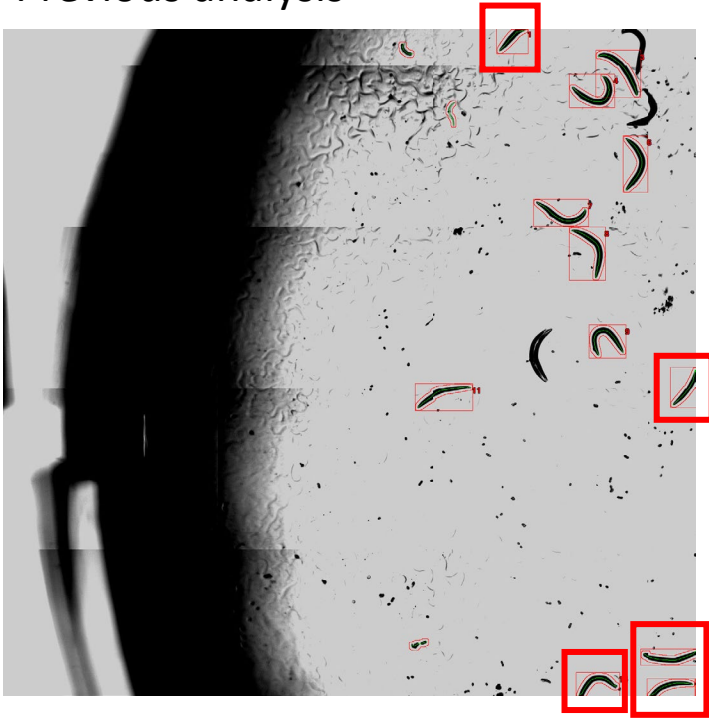
- AniLength runs on Microsoft Windows OS with .NET 5 installed.
 - Users will have to install ".NET 5 (x64)" and "Microsoft Visual C++ 2015-2022 Redistributable (X64)" on their computer.
 - CPU must support [Intel Advanced Vector Extensions \(AVX\)](#) to run AniLength.
 - Users can check AVX support using the [Intel Processor Identification Utility](#).
- AniLength supports both CPU and GPU processing.
- GPU processing greatly reduces processing time.
- To enable GPU processing,
 - Nvidia graphics card with CUDA cores must be installed in the computer.
 - Users must install both [CUDA 10.0](#) and [cuDNN 7.6](#) software.
- * Simply installing CUDA 10.0 and cuDNN 7.6 does not complete the settings for GPU CUDA processing. See other CUDA/cuDNN installation tutorials.

Version history

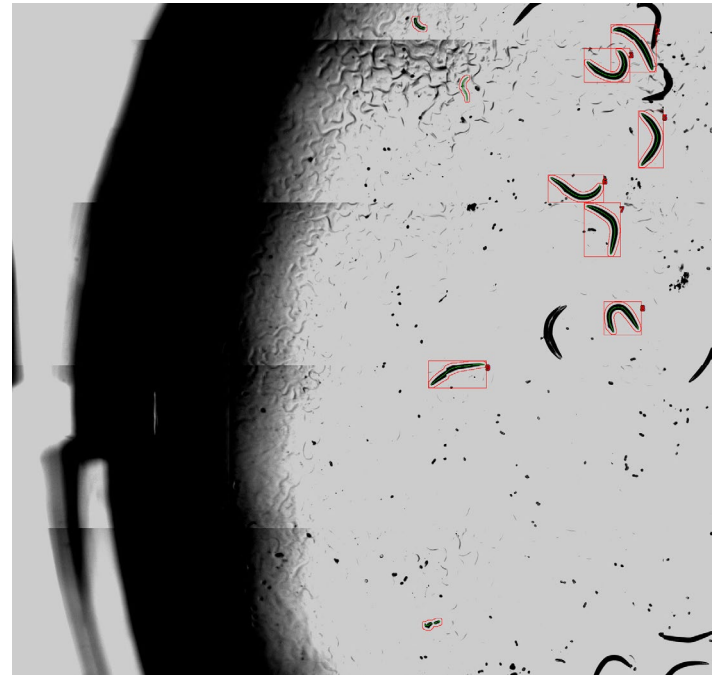
- **Version 1.1 (9/8/2022)**

Users can exclude animals that have been cut near the borders of the image.

Previous analysis



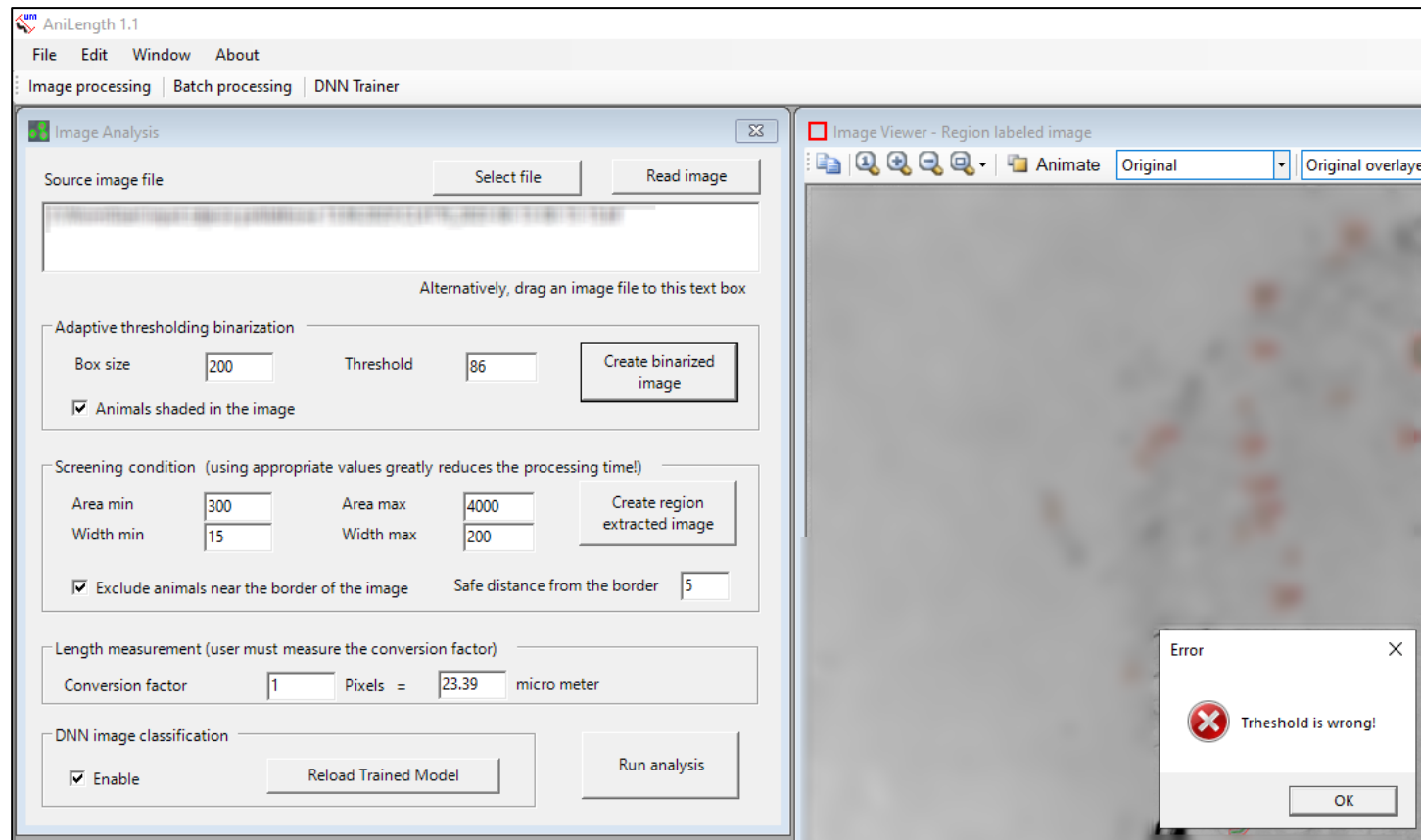
Improved analysis



Version history

- **Version 1.2 (6/20/2023)**

Fixed unexpected program errors caused by regional settings in some countries on Windows.



User interface

Viewer window

Menu

Main window for analysis

Image Viewer - Region labeled image

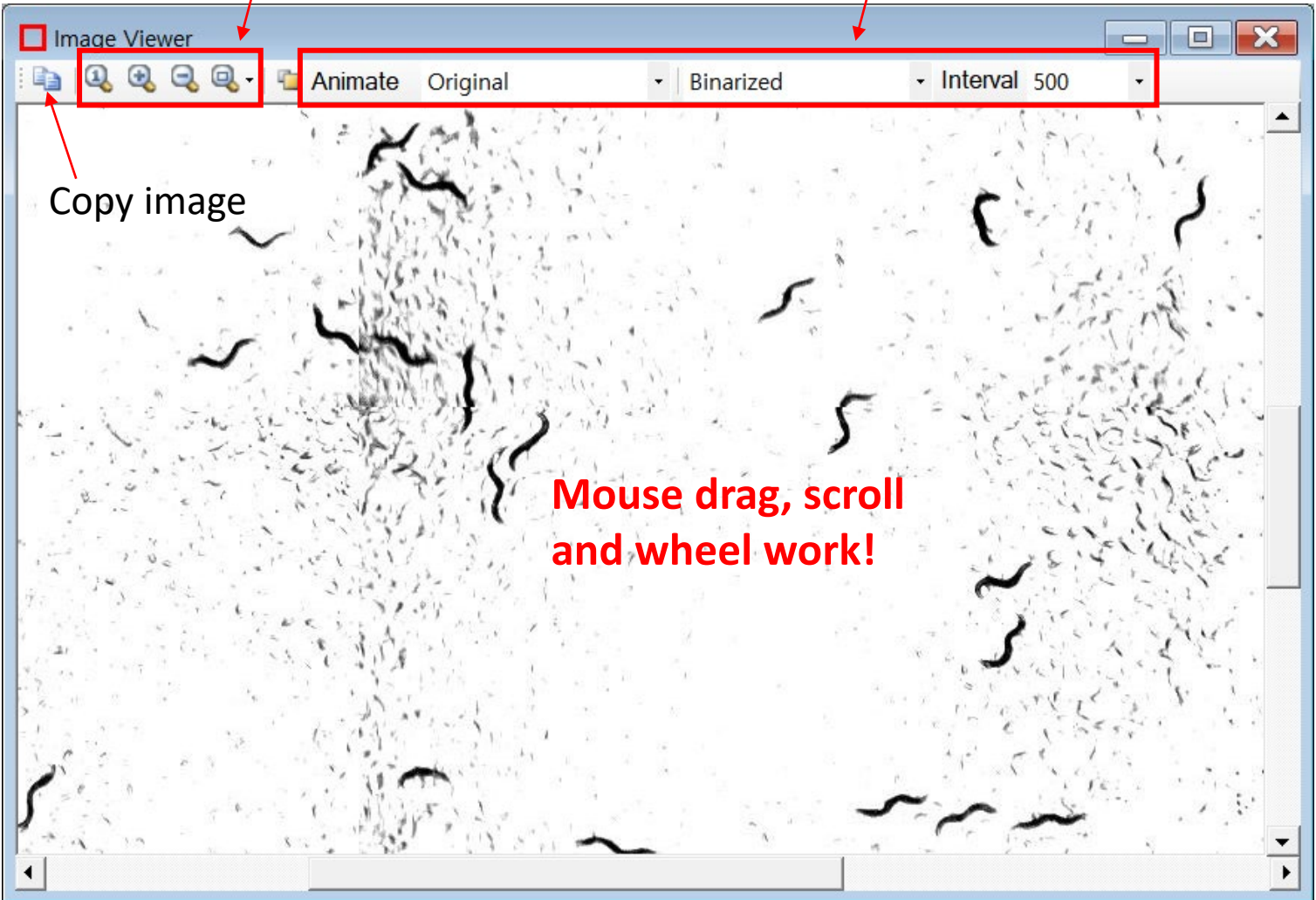
No animation Original Original overlaid Interval 500

5

Image Viewer window

Zoom in and out

Enable or disable animation between two images



Copy image

**Mouse drag, scroll
and wheel work!**

Length measurement in Image Analysis window

The screenshot shows the 'Image Analysis' window with the following components:

- Source image file:** A text box containing the path `D:\WAniLength Source training images\WVL40\WVL2_40.jpeg`. A red box highlights the **Select file** button next to it.
- Parameter Adjustment Section:** A large red box encompasses the 'Adaptive thresholding binarization' and 'Screening condition' sections. The 'Adaptive thresholding binarization' section includes a 'Box size' of 200, a 'Threshold' of 86, and a 'Create binarized image' button. The 'Screening condition' section includes 'Area min' (300), 'Area max' (4000), 'Width min' (15), 'Width max' (200), and a 'Create region extracted image' button. There is also a checkbox for 'Animals shaded in the image' and 'Exclude animals near the border of the image' with a 'Safe distance from the border' of 5.
- Length measurement:** A section with a 'Conversion factor' of 1 Pixel = 23.39 micro meter.
- DNN image classification:** A section with an 'Enable' checkbox and a 'Reload Trained Model' button.
- Run analysis:** A red box highlights the **Run analysis** button at the bottom right.

1. Select an image file.

2. Adjust parameters (default values should work for normal use).

3. Click on the 'Run analysis' button to start the analysis.

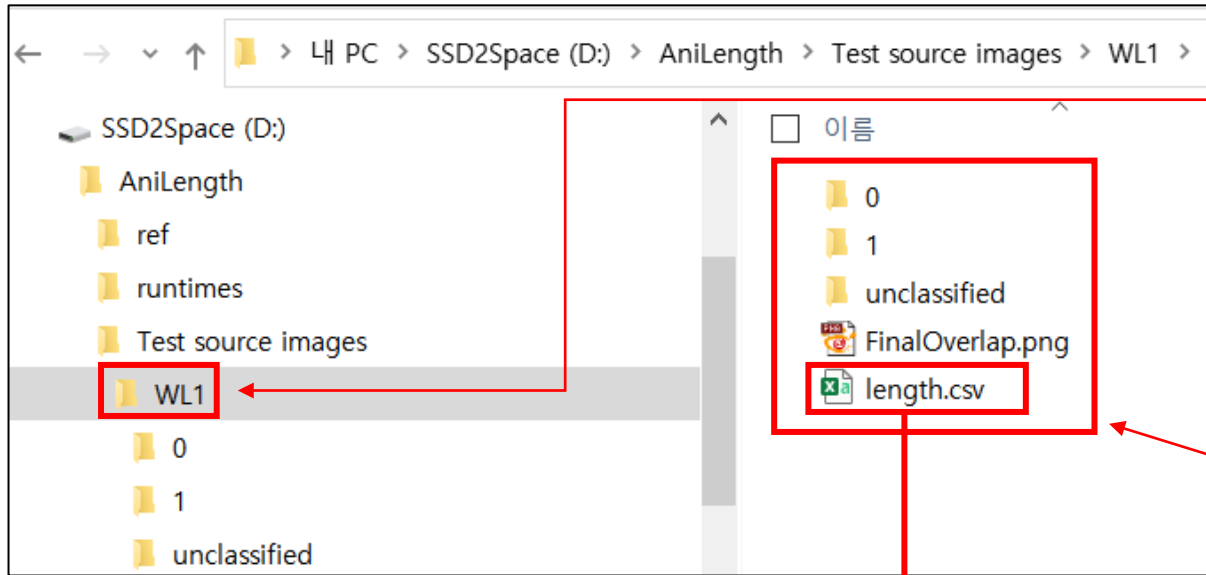
Length measurement in Image Analysis window

The screenshot shows the 'Image Analysis' window with the following settings:

- Source image file: D:\WAniLength Source training images\WL40\WL2_40.jpeg
- Adaptive thresholding binarization: Box size 200, Threshold 86, Animals shaded in the image (checked).
- Screening condition: Area min 300, Area max 4000, Width min 15, Width max 200, Exclude animals near the border of the image (checked), Safe distance from the border 5.
- Length measurement: Conversion factor 1 Pixels = 23.39 micro meter.
- DNN image classification: Enable (checked).

- Box size: Set a number larger than the worm thickness at least.
- Threshold: Detect blurry worms by adjusting sensitivity.
- Area min, Area max, width(height) min, width(height) max: Objects outside the range are excluded from analysis.
- Safe distance from the border: Animals close to borders are excluded from detection.
- Conversion factor: Users must measure the conversion factor before analysis.

Length measurement in Image Analysis window

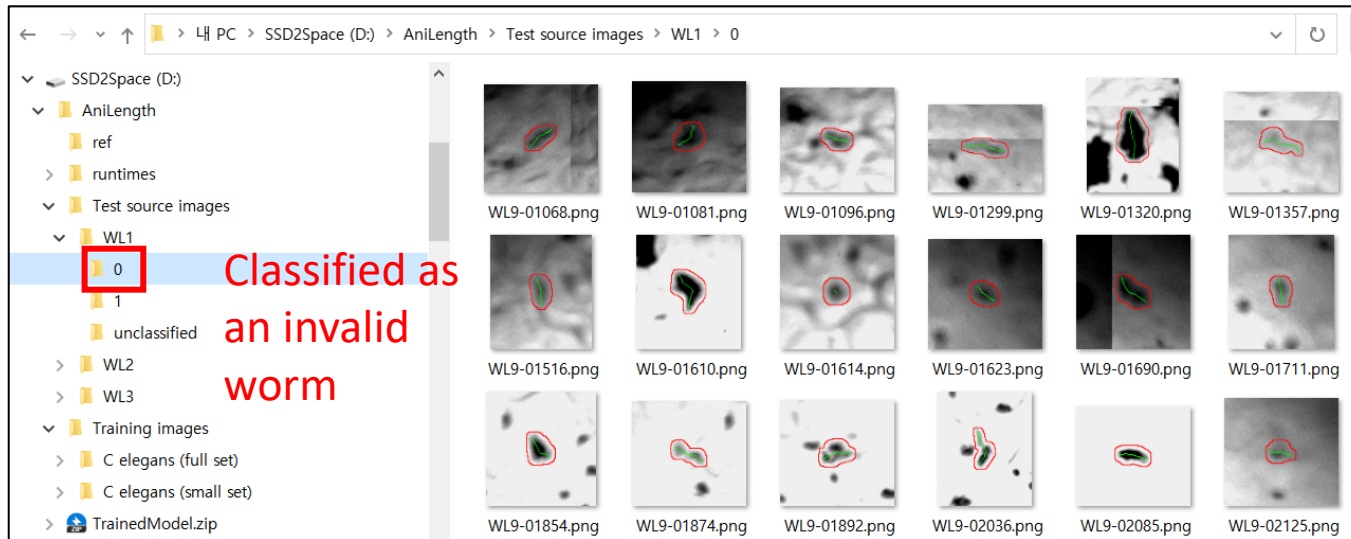


AniLength creates a new subfolder after image analysis.

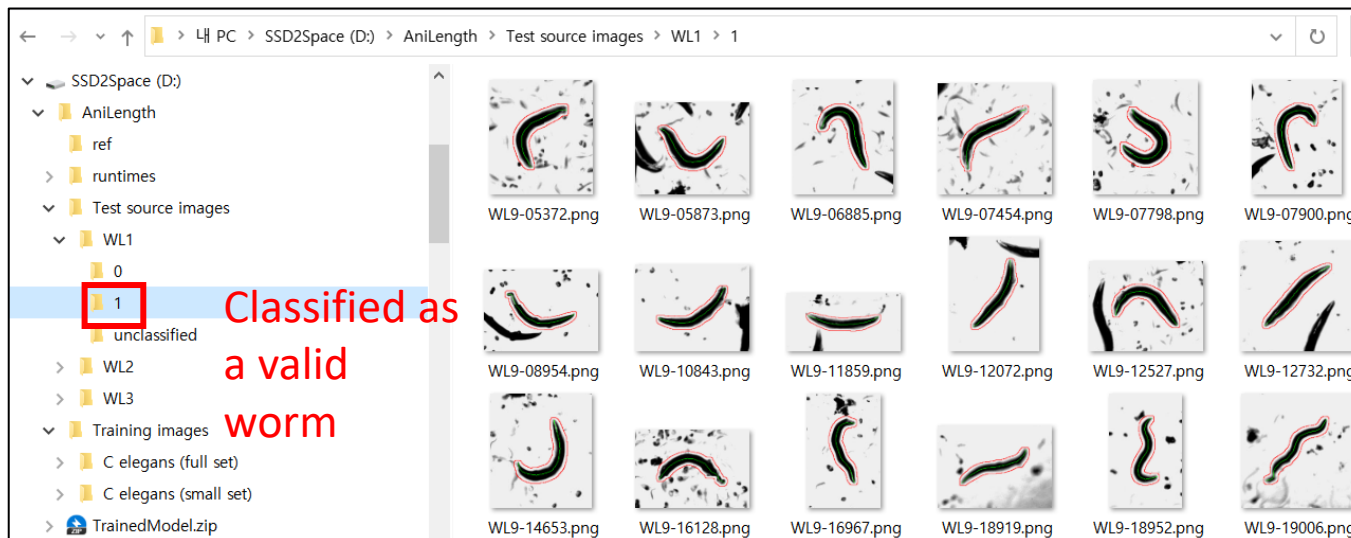
The subfolder contained the output files.

	A	B	C	D	E	F
1	AnimalID	length(um)	Area(pixels)	ROI_Width	ROI_Height	ROI_ID
2	97	1076.03	3465	99	120	5372
3	111	1123.25	3575	128	84	5873
4	123	1138.09	3364	92	118	6885
5	136	1220.88	3581	132	127	7454
6	139	1070.88	3222	81	97	7798
7	144	1190.63	3164	81	130	7900
8	145	998.38	2663	153	34	7972
9	152	1134.82	3181	125	72	8357
10	164	1151.97	2815	149	79	8954

Length measurement in Image Analysis window



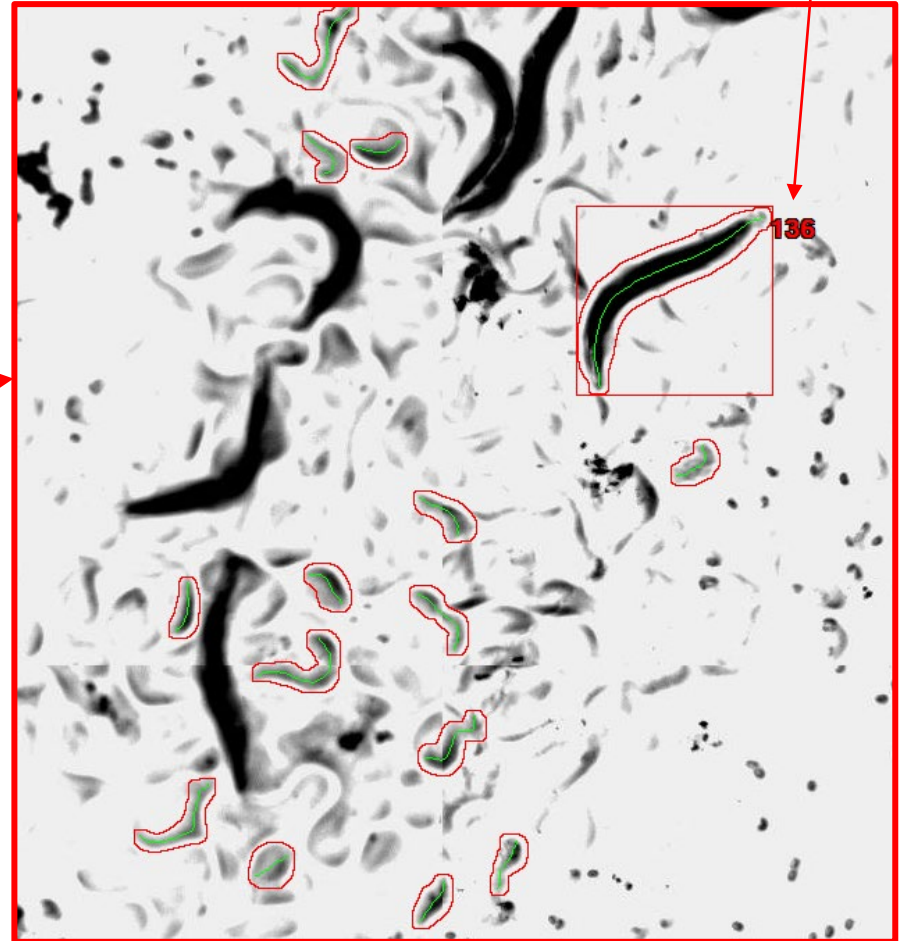
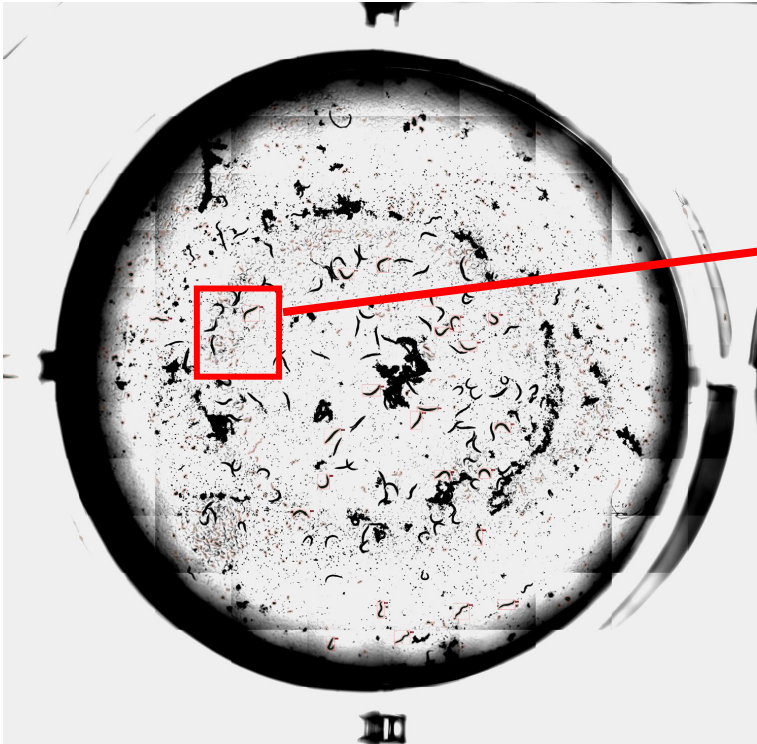
Sample contents of output folders marked "0" and "1"



Length measurement in Image Analysis window

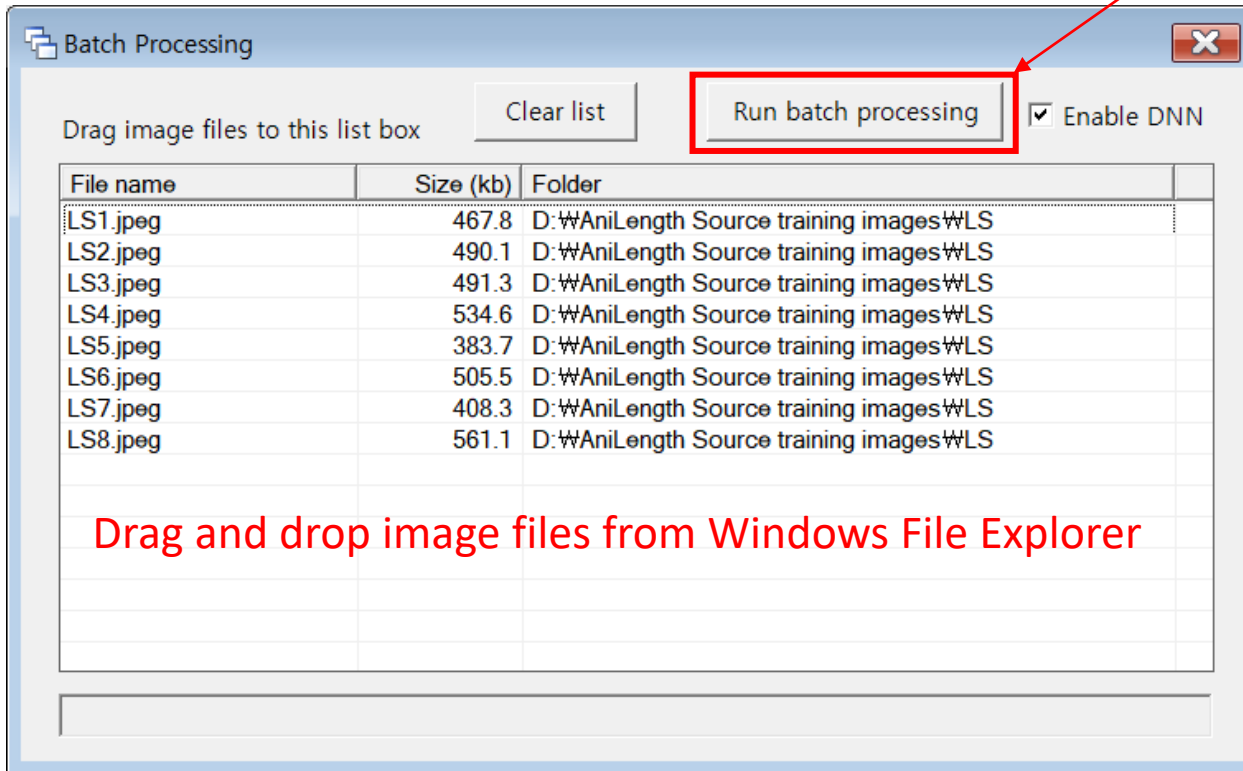
Only valid worms are marked with numbers.

Sample "FinalOverlap.png"

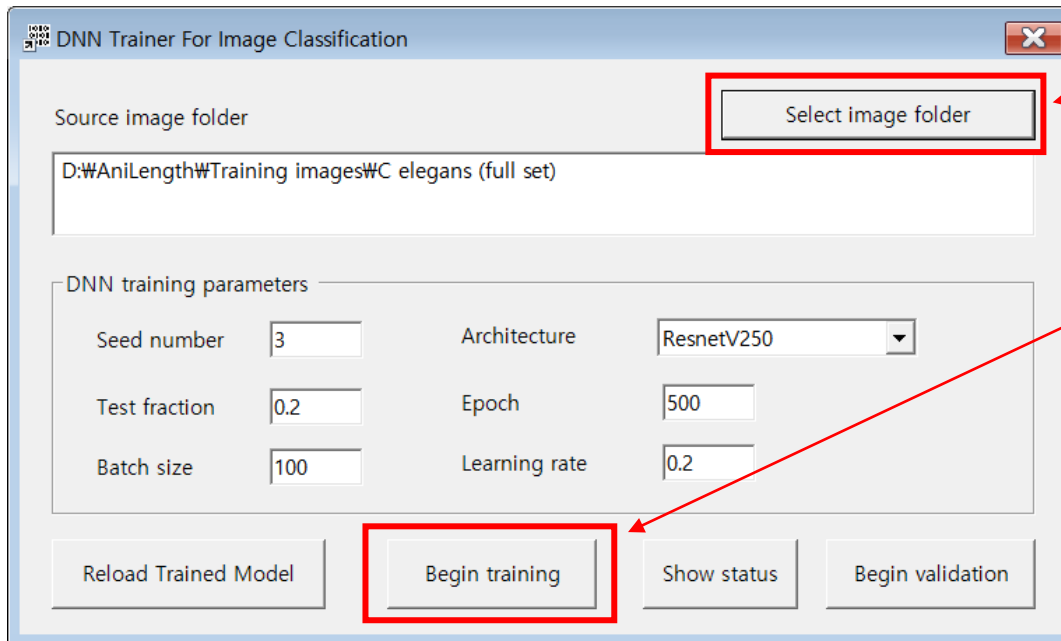


Batch processing

Click the button to start the batch processing.



DNN image classification training

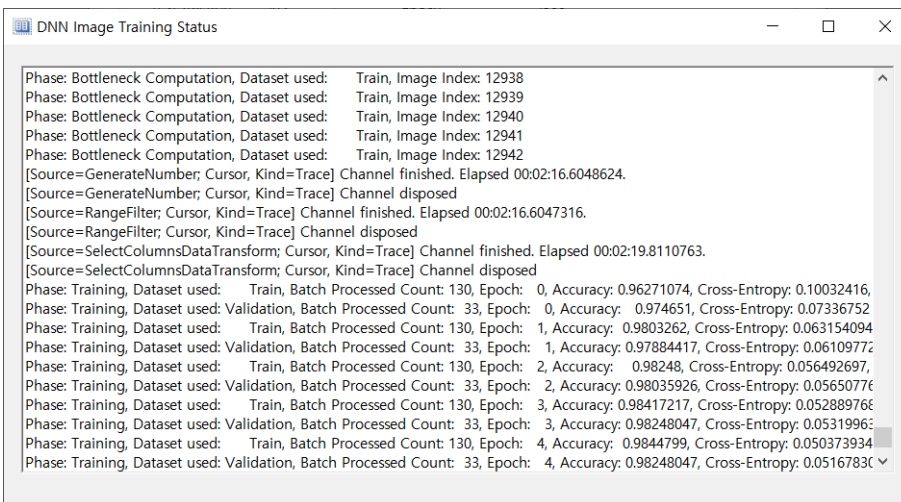


1. Select source image folder containing training images.

2. Click on the 'Begin training' button to start the training.

3. When the training is completed, "TrainedModel.zip" is created in the source image folder.

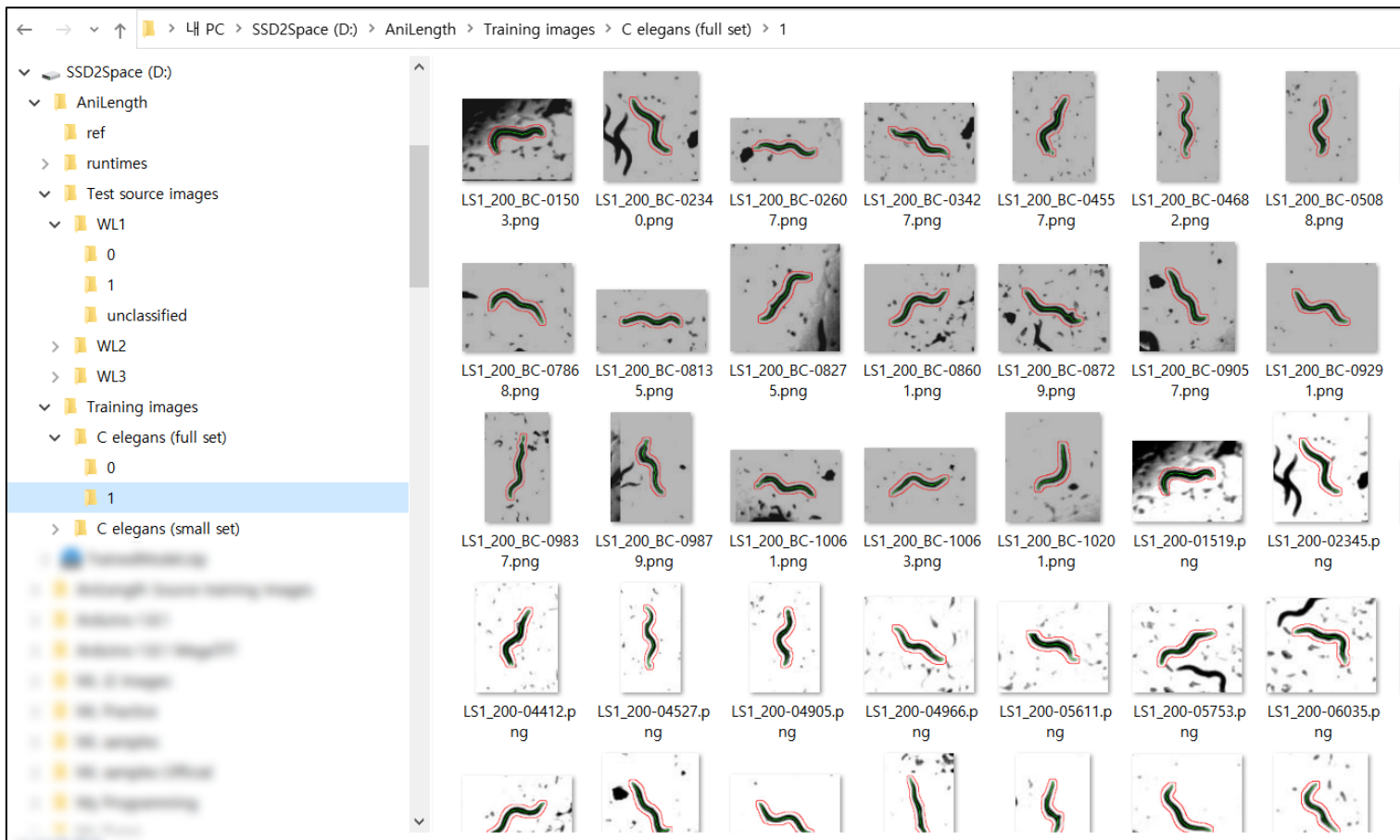
4. Users can replace the original "TrainedModel.zip" with a new one. Find the original zip file in the folder where "AniLenth.exe" is located.



This training status window pops up

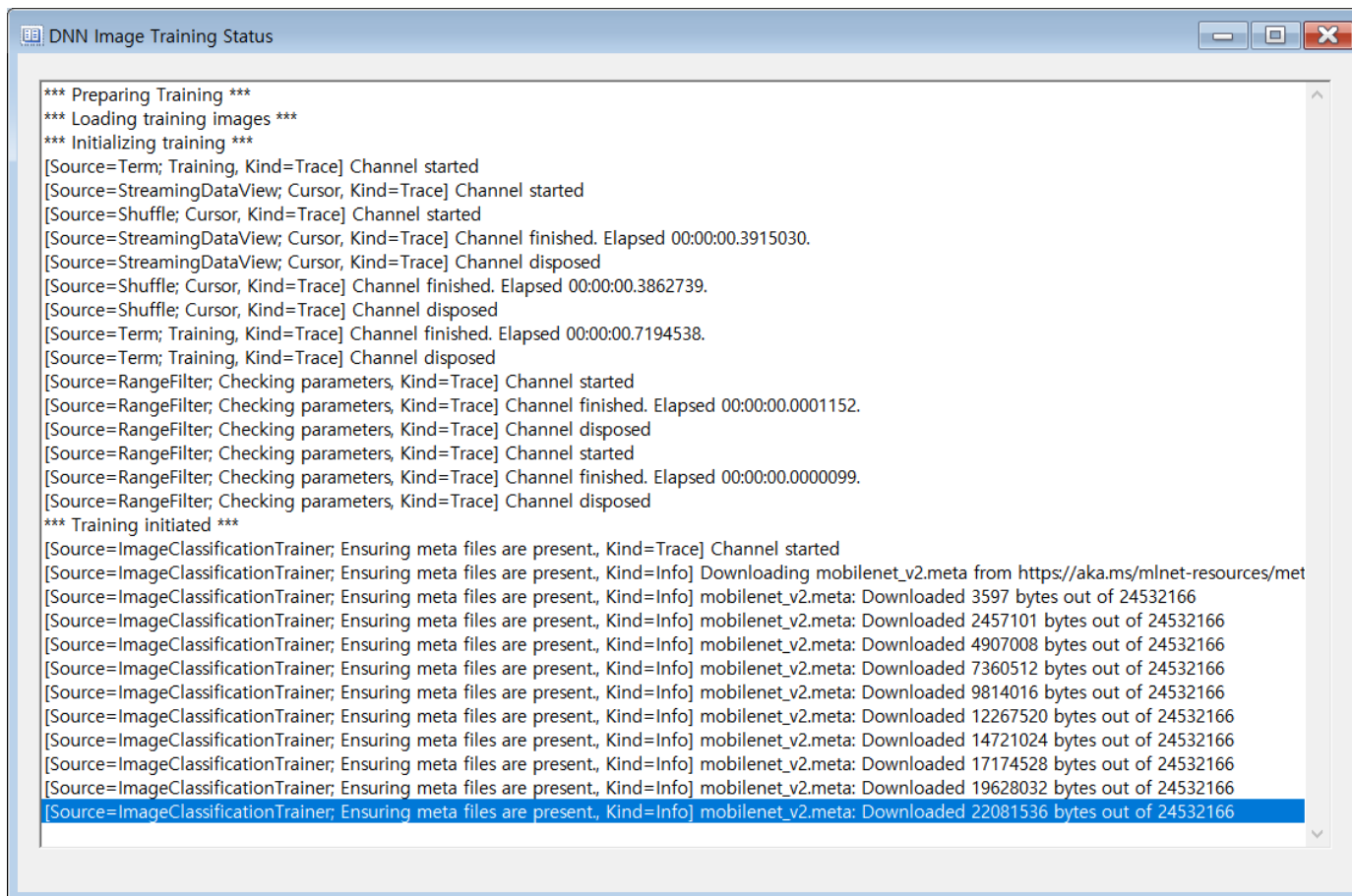
DNN image classification training

Sample training images are included in the "C elegans (full set)" and "C elegans (small set)" folders. Training images are easily created by analyzing source image in the "Image Analysis" window (See slide 8)



DNN image classification training

Pretrained DNN models (ResnetV250, ResnetV2101, InceptionV3, and MobilenetV2) must be downloaded from the Internet when used for the first time, so the computer must be connected to the Internet, and it takes time to download.



```
*** Preparing Training ***
*** Loading training images ***
*** Initializing training ***
[Source=Term; Training, Kind=Trace] Channel started
[Source=StreamingDataView; Cursor, Kind=Trace] Channel started
[Source=Shuffle; Cursor, Kind=Trace] Channel started
[Source=StreamingDataView; Cursor, Kind=Trace] Channel finished. Elapsed 00:00:00.3915030.
[Source=StreamingDataView; Cursor, Kind=Trace] Channel disposed
[Source=Shuffle; Cursor, Kind=Trace] Channel finished. Elapsed 00:00:00.3862739.
[Source=Shuffle; Cursor, Kind=Trace] Channel disposed
[Source=Term; Training, Kind=Trace] Channel finished. Elapsed 00:00:00.7194538.
[Source=Term; Training, Kind=Trace] Channel disposed
[Source=RangeFilter; Checking parameters, Kind=Trace] Channel started
[Source=RangeFilter; Checking parameters, Kind=Trace] Channel finished. Elapsed 00:00:00.0001152.
[Source=RangeFilter; Checking parameters, Kind=Trace] Channel disposed
[Source=RangeFilter; Checking parameters, Kind=Trace] Channel started
[Source=RangeFilter; Checking parameters, Kind=Trace] Channel finished. Elapsed 00:00:00.0000099.
[Source=RangeFilter; Checking parameters, Kind=Trace] Channel disposed
*** Training initiated ***
[Source=ImageClassificationTrainer; Ensuring meta files are present., Kind=Trace] Channel started
[Source=ImageClassificationTrainer; Ensuring meta files are present., Kind=Info] Downloading mobilenet_v2.meta from https://aka.ms/mlnet-resources/met
[Source=ImageClassificationTrainer; Ensuring meta files are present., Kind=Info] mobilenet_v2.meta: Downloaded 3597 bytes out of 24532166
[Source=ImageClassificationTrainer; Ensuring meta files are present., Kind=Info] mobilenet_v2.meta: Downloaded 2457101 bytes out of 24532166
[Source=ImageClassificationTrainer; Ensuring meta files are present., Kind=Info] mobilenet_v2.meta: Downloaded 4907008 bytes out of 24532166
[Source=ImageClassificationTrainer; Ensuring meta files are present., Kind=Info] mobilenet_v2.meta: Downloaded 7360512 bytes out of 24532166
[Source=ImageClassificationTrainer; Ensuring meta files are present., Kind=Info] mobilenet_v2.meta: Downloaded 9814016 bytes out of 24532166
[Source=ImageClassificationTrainer; Ensuring meta files are present., Kind=Info] mobilenet_v2.meta: Downloaded 12267520 bytes out of 24532166
[Source=ImageClassificationTrainer; Ensuring meta files are present., Kind=Info] mobilenet_v2.meta: Downloaded 14721024 bytes out of 24532166
[Source=ImageClassificationTrainer; Ensuring meta files are present., Kind=Info] mobilenet_v2.meta: Downloaded 17174528 bytes out of 24532166
[Source=ImageClassificationTrainer; Ensuring meta files are present., Kind=Info] mobilenet_v2.meta: Downloaded 19628032 bytes out of 24532166
[Source=ImageClassificationTrainer; Ensuring meta files are present., Kind=Info] mobilenet_v2.meta: Downloaded 22081536 bytes out of 24532166
```

DNN image classification training

Optionally, the user can generate a training analysis report

DNN Trainer For Image Classification

Source image folder

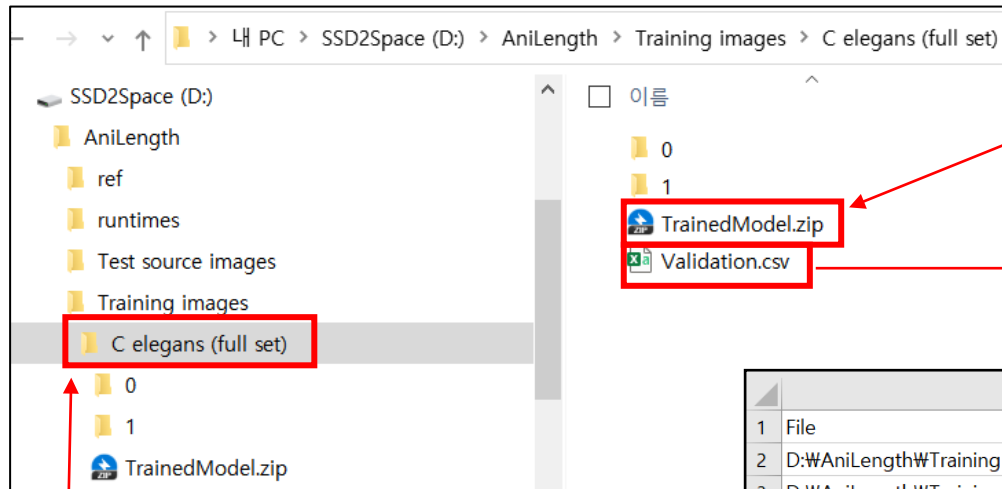
D:\#AniLength#Training images#C elegans (full set)

DNN training parameters

Seed number	<input type="text" value="3"/>	Architecture	<input type="text" value="ResnetV250"/>
Test fraction	<input type="text" value="0.2"/>	Epoch	<input type="text" value="500"/>
Batch size	<input type="text" value="100"/>	Learning rate	<input type="text" value="0.2"/>

Click on the 'Begin validation' button to generate a validation report named "Validation.csv" for the current loaded trained model.

DNN image classification training



Newly created trained model file

Sample "Validation.csv"

Source training image folder

	A	B	C	D
1	File	real	predicted	correct
2	D:\AniLength\Training images\C elegans (full set)\W0WLS1-00303.png	0	0	1
3	D:\AniLength\Training images\C elegans (full set)\W0WLS1-00388.png	0	0	1
4	D:\AniLength\Training images\C elegans (full set)\W0WLS1-00474.png	0	1	0
5	D:\AniLength\Training images\C elegans (full set)\W0WLS1-00502.png	0	0	1
6	D:\AniLength\Training images\C elegans (full set)\W0WLS1-00567.png	0	0	1
7	D:\AniLength\Training images\C elegans (full set)\W0WLS1-00632.png	0	0	1
8	D:\AniLength\Training images\C elegans (full set)\W0WLS1-00638.png	0	0	1
9	D:\AniLength\Training images\C elegans (full set)\W0WLS1-00870.png	0	0	1
10	D:\AniLength\Training images\C elegans (full set)\W0WLS1-00910.png	0	0	1
16225	D:\AniLength\Training images\C elegans (full set)\W1WLS12-06069.png	0	1	1
16226	D:\AniLength\Training images\C elegans (full set)\W1WLS12-06085.png	0	1	1
16227	D:\AniLength\Training images\C elegans (full set)\W1WLS12-06106.png	0	1	1
16228	D:\AniLength\Training images\C elegans (full set)\W1WLS12-06147.png	0	1	1
16229	D:\AniLength\Training images\C elegans (full set)\W1WLS12-06175.png	0	1	1
16230	-----			
16231	True negative : 8611 of 8763Accuracy of 98.265%			
16232	False negative : 152 of 8763Accuracy of 1.735%			
16233	True positive : 7416 of 7465Accuracy of 99.344%			
16234	False positive : 49 of 7465Accuracy of 0.656%			
16235	-----			
16236	Correct: 16027 of 16228Accuracy of 98.761%			
16237	Wrong : 201 of 16228Accuracy of 1.239%			

License info

LGPL v3.0

ANILENGTH IS DISTRIBUTED 'AS IS'. NO WARRANTY OF ANY KIND IS EXPRESSED OR IMPLIED. YOU USE THE PROGRAM AT YOUR OWN RISK.

AniLength uses the following compiled libraries:

Microsoft.ML (MIT license)

Microsoft.ML.Vision (MIT license)

Microsoft.ML.ImageAnalytics (MIT license)

SharpZipLib (MIT license)

SciSharp.TensorFlow.Redist-Windows-GPU (Apache License)

Queens_ImageControl ("you are free to redistribute it and/or modify it")

Some of the images used for this project were generated from a previous nanotoxicity study.

Jung, Sang-Kyu, Xiaolei Qu, Boanerges Aleman-Meza, Tianxiao Wang, Celeste Riepe, Zheng Liu, Qilin Li, and Weiwei Zhong. "Multi-endpoint, high-throughput study of nanomaterial toxicity in *Caenorhabditis elegans*." *Environmental science & technology* 49, no. 4 (2015): 2477-2485

Alternative license terms are available upon request.