



# yolo v8 用のラベル付きデータの作成 それによる学習の実行及び 学習結果の表示

株式会社フュージョンシス

文書バージョン	日付	変更点
ver. 1.0	2023 年 7 月 14 日	作成

# 目次

1.	今回のご相談の内容と本文書の目的	1
2.	前提とされる知識や能力	1
3.	プログラム類	2
3.1	labelimg と yolo v8	2
3.2	python 環境	2
3.2.1	python の version	2
3.2.2	python にインストールした module たち	2
3.2.3	labelimg	2
3.2.4	yolo v8	3
4.	言葉の説明	4
4.1	略語	4
4.2	NN の能力	4
4.3	NN の 3 つのフェイズ	4
4.4	学習時のラベリング	4
4.5	annotation ツール	4
5.	流れ	5
6.	annotation 用ツール	7
7.	labelimg	8
7.1	visual studio code を立ち上げ、open folder を選びます	8
7.2	labelimg の起動します。	9
7.3	labelimg.py を選んで、右上の▶を押します。	10
7.4	labelimg の起動画面	11
7.5	labelimg でラベル付を行う対象の画像群を選んだ	12
7.6	ラベル付を行う	13
7.7	「保存する」&「次の画像」	14
8.	yolo v8 が読める annotation ファイルの形式	16
9.	yolo v8 による、サンプル課題の学習	17
9.1	yolo v8 の起動	17
9.2	for_floor_image を選ぶ	18
9.3	yolo v8 の起動直後の画面	19
9.4	フォルダやファイルの配置が正しいことを確認します	20
9.5	annotation ファイル	21
9.6	サンプル課題で学習する内容	22
9.6.1	区別する内容	22
9.6.2	ある画像の中の sperm の位置	22
9.6.3	ある画像の中の gomi の位置	22
9.7	learn_corners.py を実行します	23
9.8	結果の確認	24
9.8.1	学習の過程 (sperm と gomi)	24
9.8.2	学習の結果 (sperm)	26
9.9	モデル (学習の結果できた NN の重みの集合) のコピー	27
10.	yolo v8 による、サンプル課題の認識	28

10.1	認識の対象となる画像たちを用意する.....	28
10.2	認識の結果.....	29

1

2 1. 今回のご相談の内容と本文書の目的

3

4 2023/6/16 に以下のご相談がありました。

5

6 >先日お話しさせていただいた学習プログラム(pt ファイル)の作成方法のマニュアル作成につ  
7 いてお見積もりをいただいてもよろしいでしょうか？

8

9 yolo の\*.pt のファイルにはニューラルネットワークの重みの集合が入っています。この文書では

10

11 1) annotation ツールを使って、yolo v8 用の学習用のデータを作成する方法

12 2) 学習を行う方法

13 3) その結果できた\*.pt(pytorch)ファイルを使って、認識を行う方法

14

15 を説明します。

16

17

18

19 2. 前提とされる知識や能力

20

21 1) windows の基本的な操作。

22 2) プログラミングは必要ないが、python に関する最小限の運用能力。

23

24

25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42  
43  
44  
45  
46  
47  
48  
49  
50  
51  
52  
53  
54  
55  
56  
57  
58  
59  
60  
61

### 3. プログラム類

#### 3.1 labeling と yolo v8

labelimg および yolo v8 を使います。labelimg および yolo v8 は、github にあるオープンソースプログラムであり、貴校の場合特に何もなくても自由に使えます。弊社の著作権の発生する部分は、今回ないので、labelimg と yolo v8 の動くプログラムを貴校に一式お送りいたします。

#### 3.2 python 環境

##### 3.2.1 python の version

```
PS E:¥labellmg-master> python --version  
Python 3.9.5  
PS E:¥labellmg-master>
```

##### 3.2.2 python にインストールした module たち

一方、python の module たちは貴校で構築していただく必要がございます。この部分は1番時間と手間のかかる部分です。

##### 3.2.3 labeling

QT 関係のモジュールが不可欠です。ただし以下が全部必要なわけではないです。他のモジュールが必要な可能性もあります。

PyQt5	5.15.7
PyQt5-Qt5	5.15.2
PyQt5-sip	12.11.0
PyQtWebEngine	5.15.6
PyQtWebEngine-Qt5	5.15.2



62  
63  
64  
65  
66  
67  
68  
69  
70  
71  
72  
73  
74  
75  
76  
77

### 3.2.4 yolo v8

torch 関係のモジュールが不可欠です。ただし以下が全部必要なわけではありません。他のモジュールが必要な可能性もあります。

torch	1.13.1
torchaudio	0.13.1
torchfile	0.1.0
torchmetrics	0.11.4
torchvision	0.14.1

78

79 4. 言葉の説明

80

81 4.1 略語

82 ✓ NN ニューラルネットワークの略です。

83 ✓ yolo v8 yolo (you only look once) の version8 です。

84

85 4.2 NN の能力

86 認識する(detect)、判別する(distinguish)、分類する(classify)、予測する(predict)などの能  
87 力があります。NN にそれらの能力を発揮させるためには、適切な学習を予め施しておくことが  
88 必要です。

89

90 4.3 NN の 3 つのフェイズ

91

92 ✓ train  
93 学習する、訓練する。

94

95 ✓ val or valid  
96 確認する。validate または validation を略しています。

97

98 ✓ テスト  
99 学習が完了した NN に、detect、distinguish、classify、predict などを行わせます。

100

101

102 4.4 学習時のラベリング

103 NN に認識(detect)あるいは判別(distinguish)してほしい、画像中のオブジェクトにラベリング  
104 をします。この行為は、ラベリング、タグ付け、annotation などと呼ばれます。

105

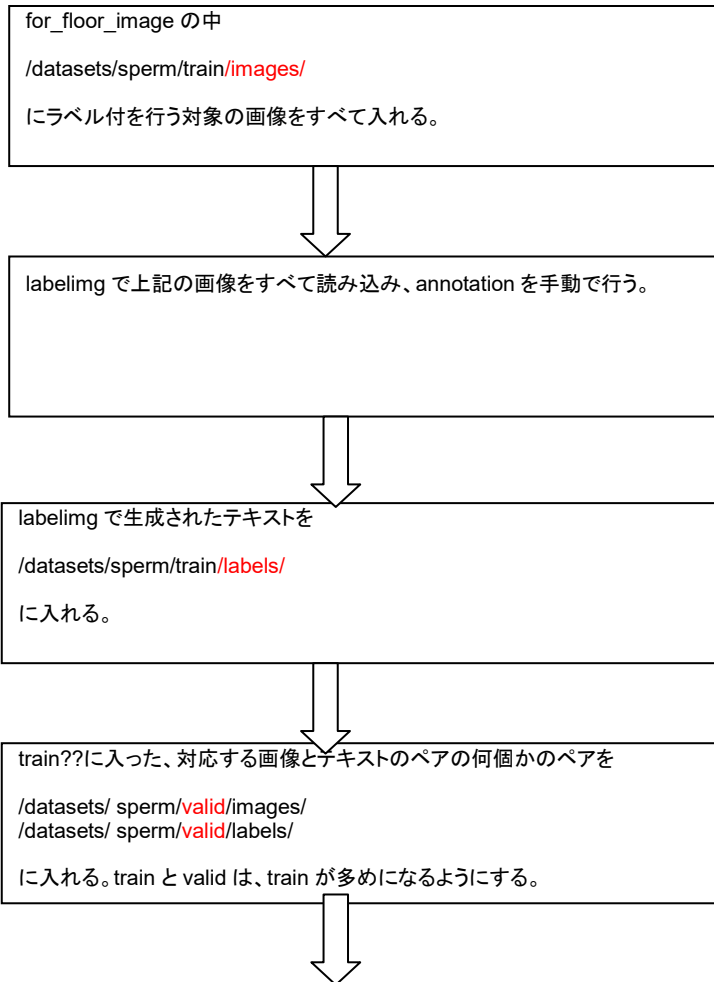
106 4.5 annotation ツール

107 annotation のためのソフトウェアです。

108

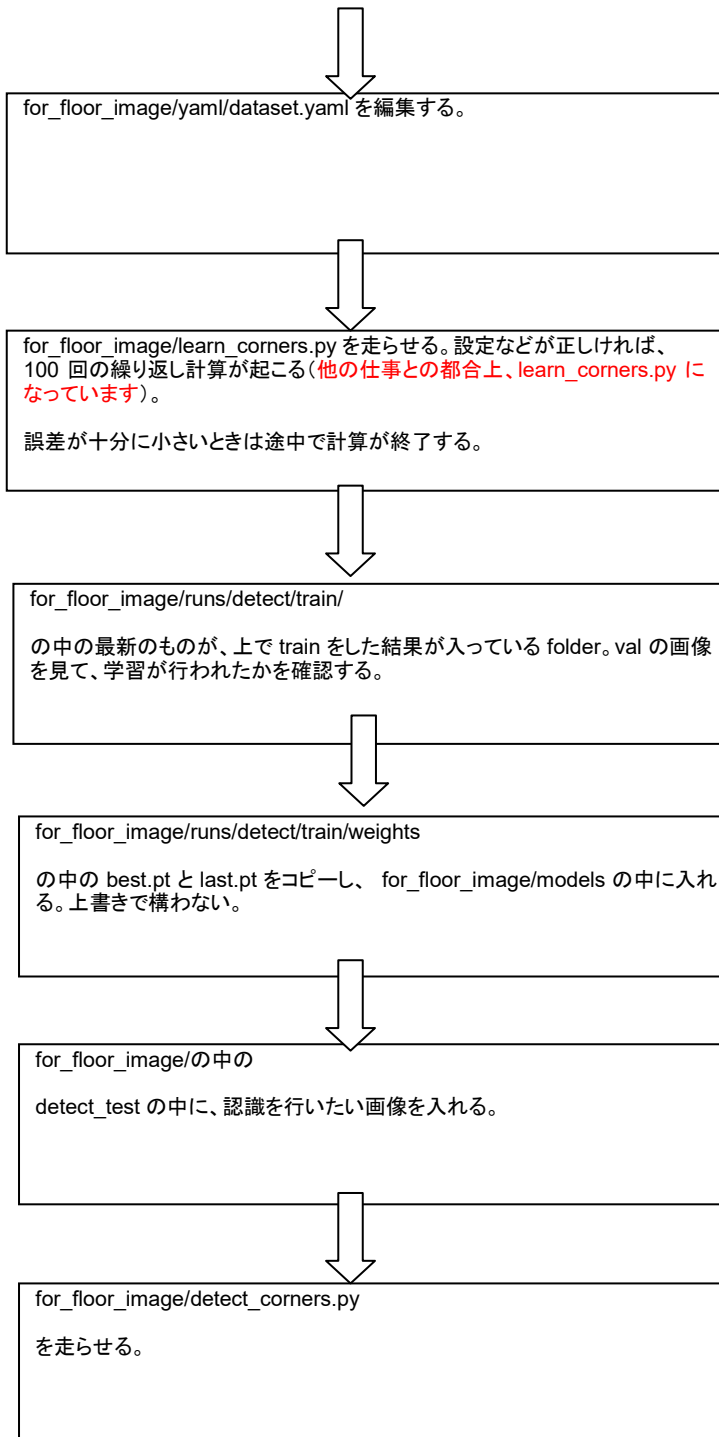
109

110  
111 5. 流れ  
112  
113





134  
135  
136  
137  
138  
139  
140  
141  
142  
143  
144  
145  
146  
147  
148  
149  
150  
151  
152  
153  
154  
155  
156  
157  
158  
159  
160  
161  
162  
163  
164  
165  
166  
167  
168  
169



6. annotation 用ツール

	labelimg	coco annotator	vott
url	<a href="https://github.com/hearterlabs/labelimg">https://github.com/hearterlabs/labelimg</a>	<a href="https://github.com/jsbroks/coco-annotator">https://github.com/jsbroks/coco-annotator</a>	<a href="https://github.com/microsoft/VoTT">https://github.com/microsoft/VoTT</a>
開発状況。	アクティブ。	アクティブ。	開発終了。
長所	使いやすい。ソフトウェアとしての完成度は高い。	サーバベースなので、複数人で共同作業ができる。	windows の standalone アプリとして実装されている。
短所	python についての若干の運用知識がいる。	ソフトウェアとしての完成度が非常に低い。	言葉の使い方が独特。開発が終了している。
yolo v8 が読める形で出力するか？	はい。	はい。	いいえ。ただし簡単に変換はできる。
向いている局面の例。	精子やゴミが多数写っている動画があって、その中で NN に精子だけを認識させたい。  1000 個の粒子の中に、2 種類の、特異な形の粒子が 2,3 個ずつあって、NN にそれらを認識させたい。	一人ではタグ付けできないほどの多数の画像があるなど。	精子やゴミが多数写っている動画があって、その中で NN に精子だけを認識させたい。
多角形を切り出せるか？	いいえ。	はい。	はい。

今回 1 番ソフトウェアとしての完成度の高い labelimg を使います。

179

180

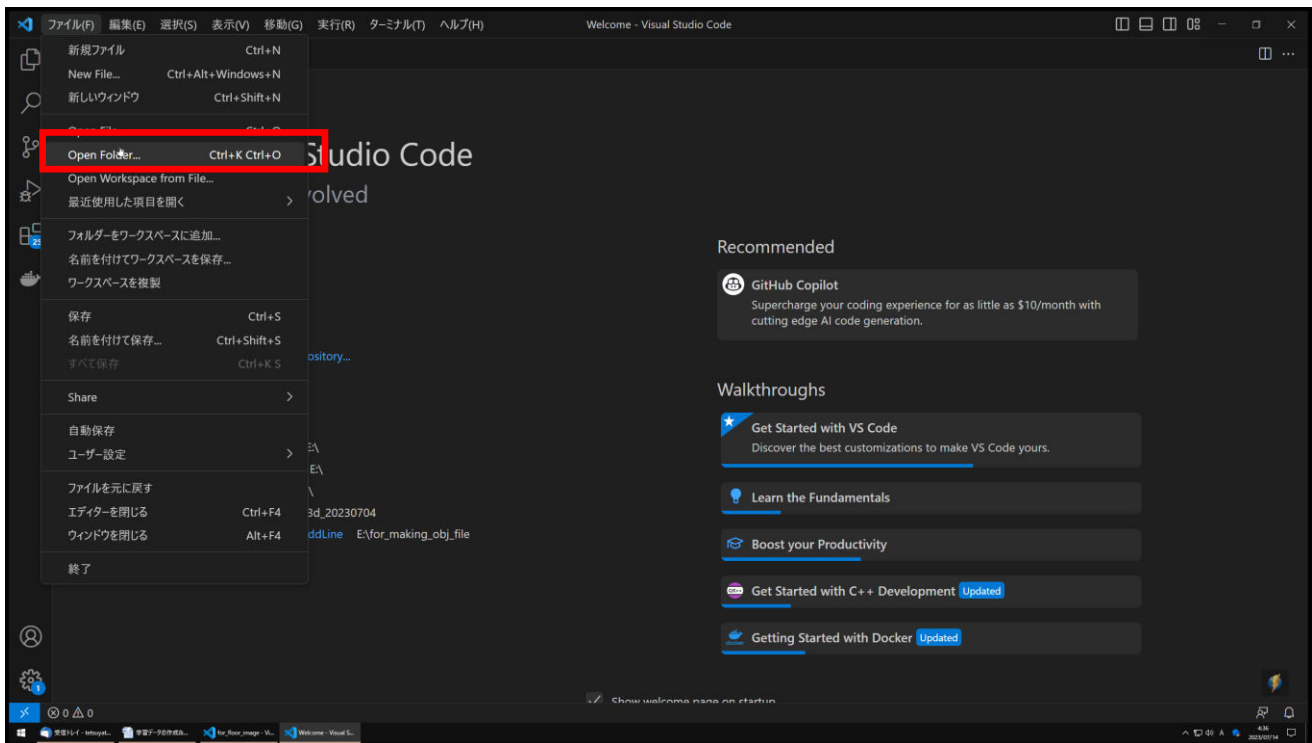
181

182

183

## 7. labeling

### 7.1 visual studio code を立ち上げ、open folder を選びます

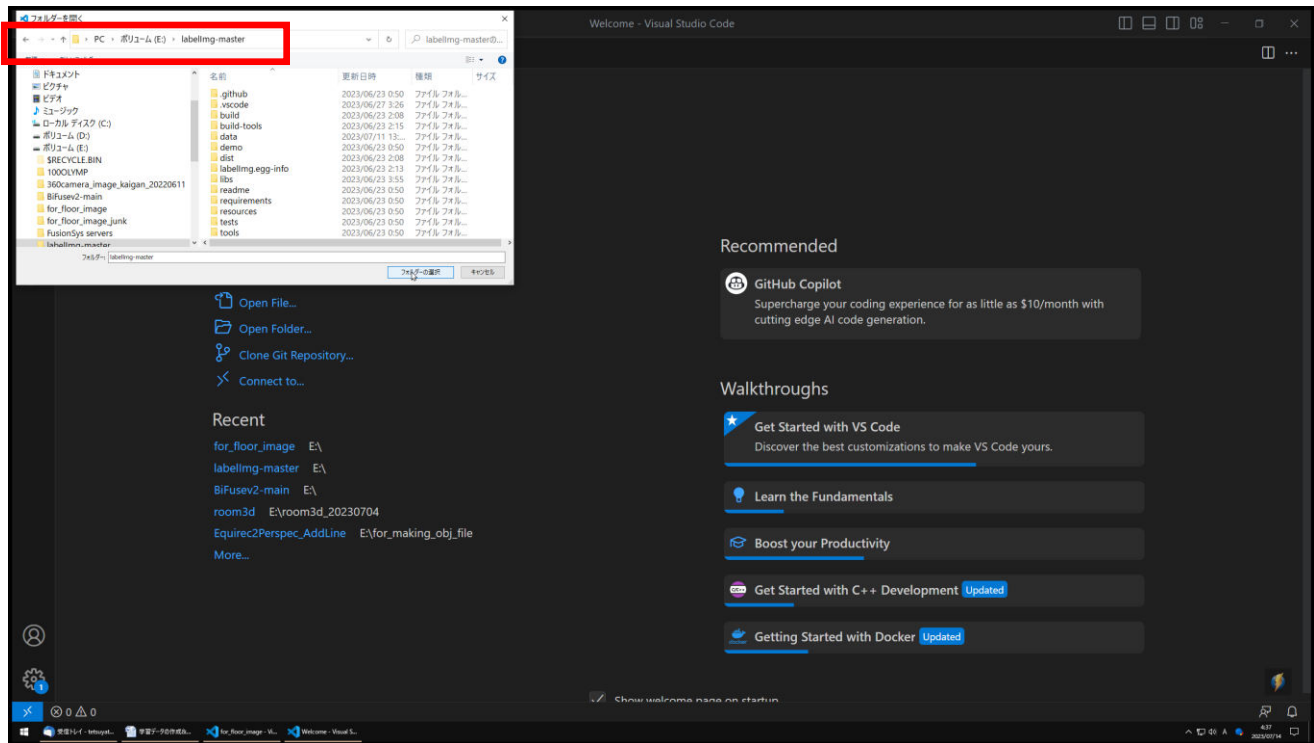


184

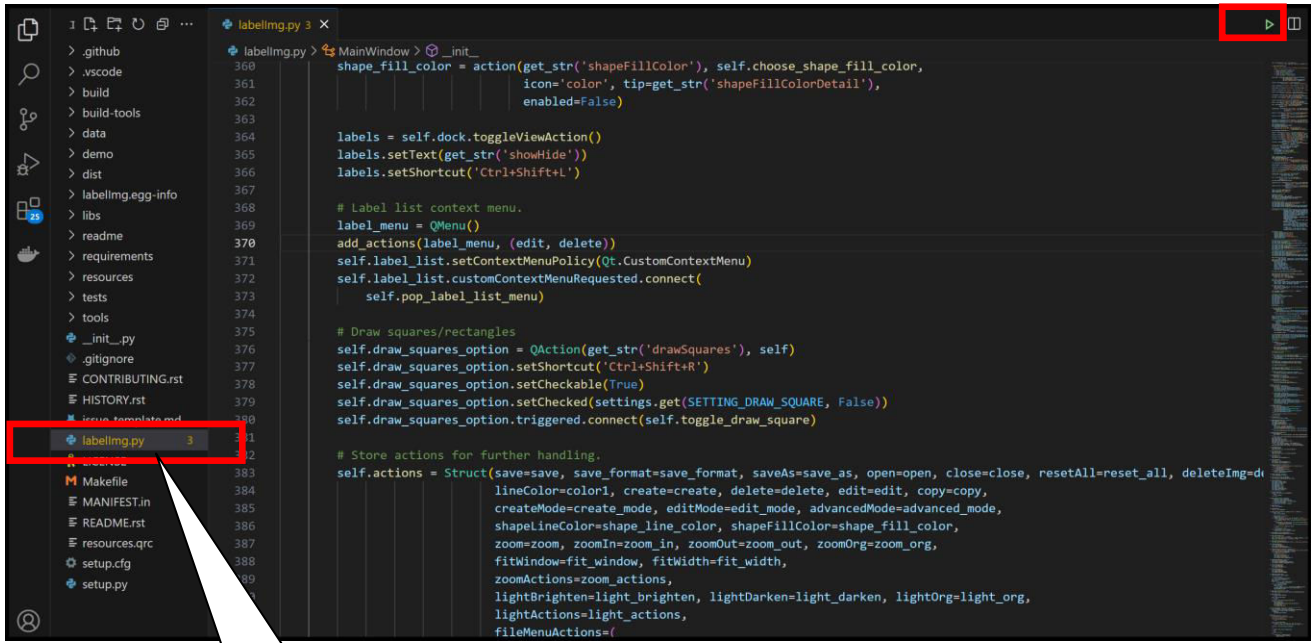
185

186

7.2 labeling の起動します。



7.3 labeling.py を選んで、右上の▶を押します。



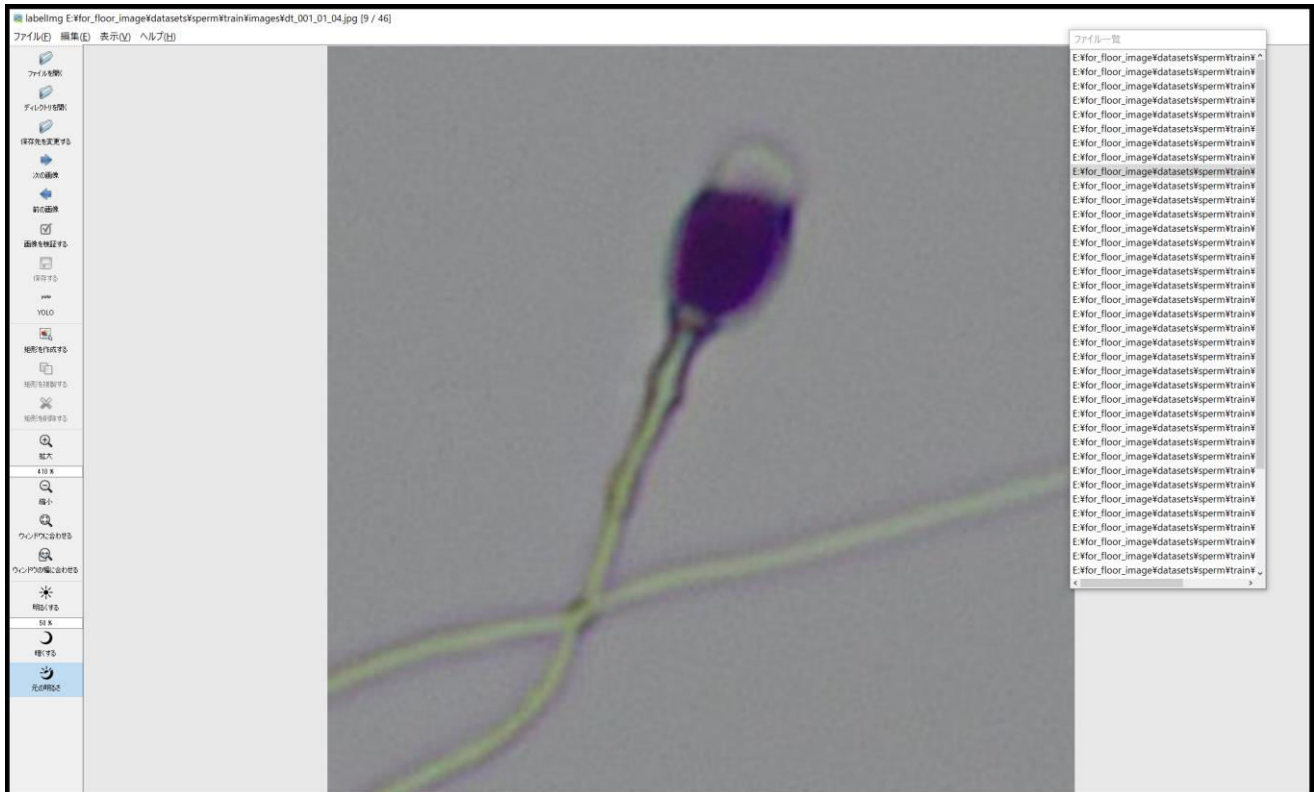
labelimg のメインのプログラ  
ム。

7.4 labeling の起動画面

labeling が起動するのに必要な module が入っていれば、次のような画面がでます。

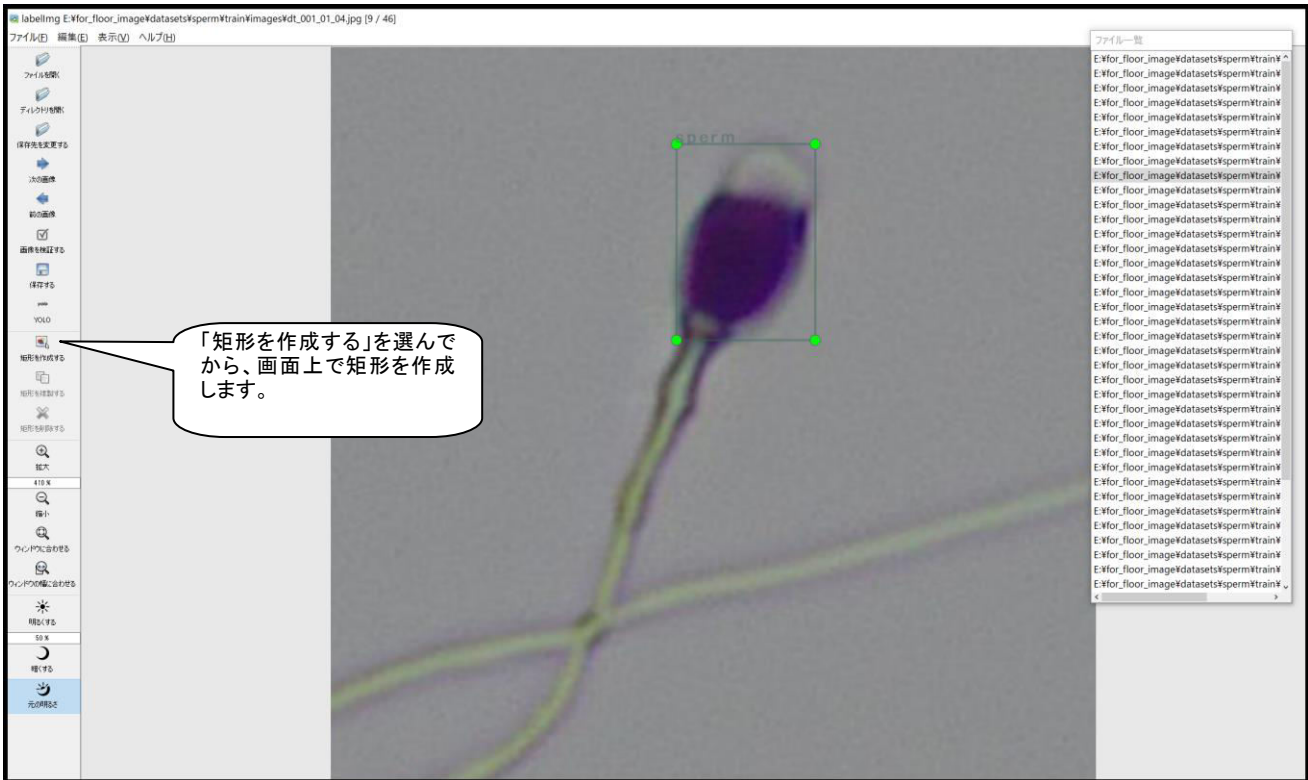


7.5 labeling でラベル付を行う対象の画像群を選んだ



7.6 ラベル付を行う

「矩形を作成する」を選んでから、画面上で矩形を作成します。



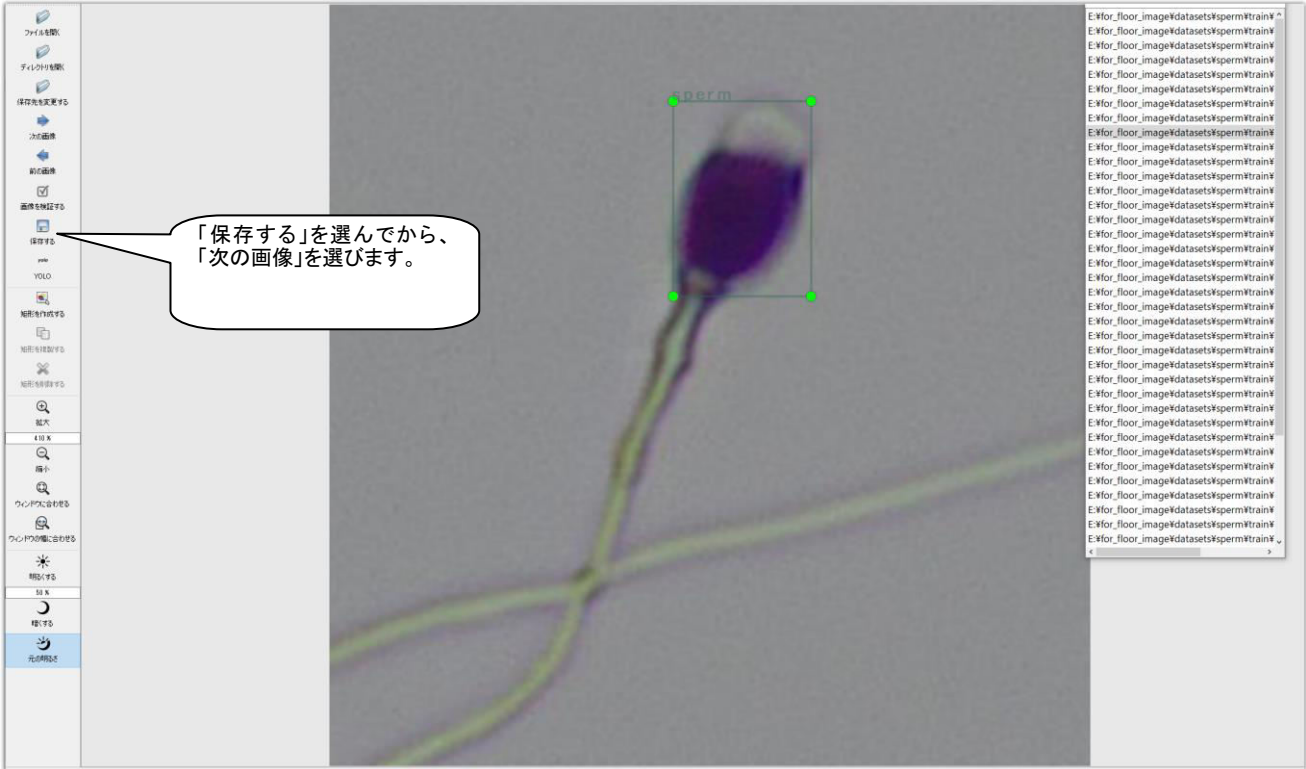
215  
216  
217

218  
219  
220  
221  
222  
223  
224  
225  
226  
227  
228  
229  
230  
231  
232  
233  
234



7.7 「保存する」&「次の画像」

「保存する」を選んでから、  
 「次の画像」を選びます。



235  
 236  
 237

238  
 239  
 240  
 241  
 242  
 243  
 244  
 245  
 246  
 247  
 248  
 249  
 250  
 251  
 252  
 253  
 254



株式会社フュージョンシステム  
〒105-0023 東京都港区芝浦 1-14-8-901  
<http://fusionsys.com/> | [info@fusionsys.com](mailto:info@fusionsys.com)

- 255
- 256
- 257
- 258
- 259
- 260
- 261
- 262
- 263
- 264
- 265
- 266
- 267
- 268
- 269
- 270
- 271

272  
273 8. yolo v8 が読める annotation ファイルの形式  
274  
275  
276

```
277 0 0.462500 0.565278 0.019444 0.022222  
278 0 0.549074 0.565972 0.020370 0.031944  
279 0 0.462037 0.332639 0.018519 0.029167  
280 1 0.554167 0.486111 0.021296 0.025000  
281 1 0.584259 0.325694 0.031481 0.040278
```

282 1 番先頭の **0, 1** は、ラベルの種類を  
283 表している。ここでは、2 種類のオブ  
284 ジェクトを、NN に学習させようとして  
285 いる。

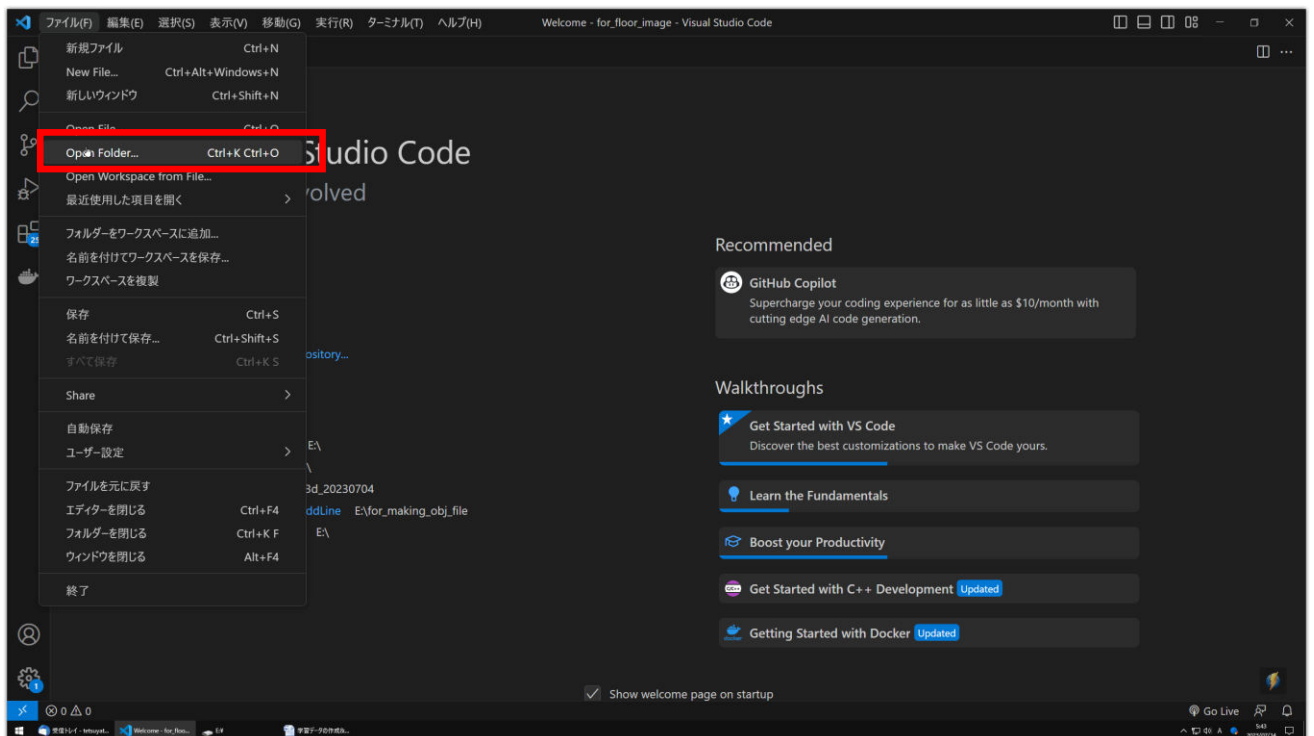
286  
287  
288  
289  
290  
291  
292  
293  
294  
295  
296  
297  
298  
299  
300  
301  
302  
303  
304  
305  
306  
307

4 つの数字は、長方形の左上の点の (x,y) 座標 および右下の (x,y) 座標。

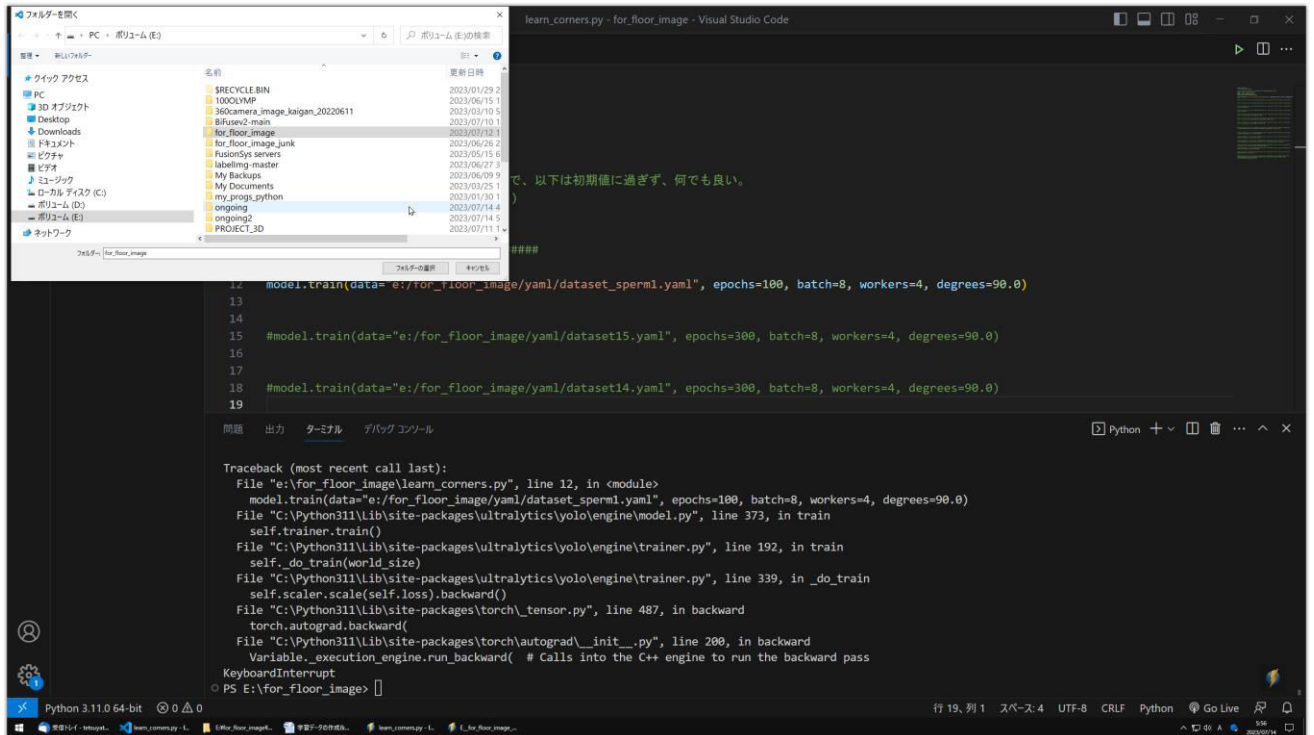
## 9. yolo v8 による、サンプル課題の学習

ここでは、sperm と gomi の 区別を学習するというサンプル課題を扱います。

### 9.1 yolo v8 の起動

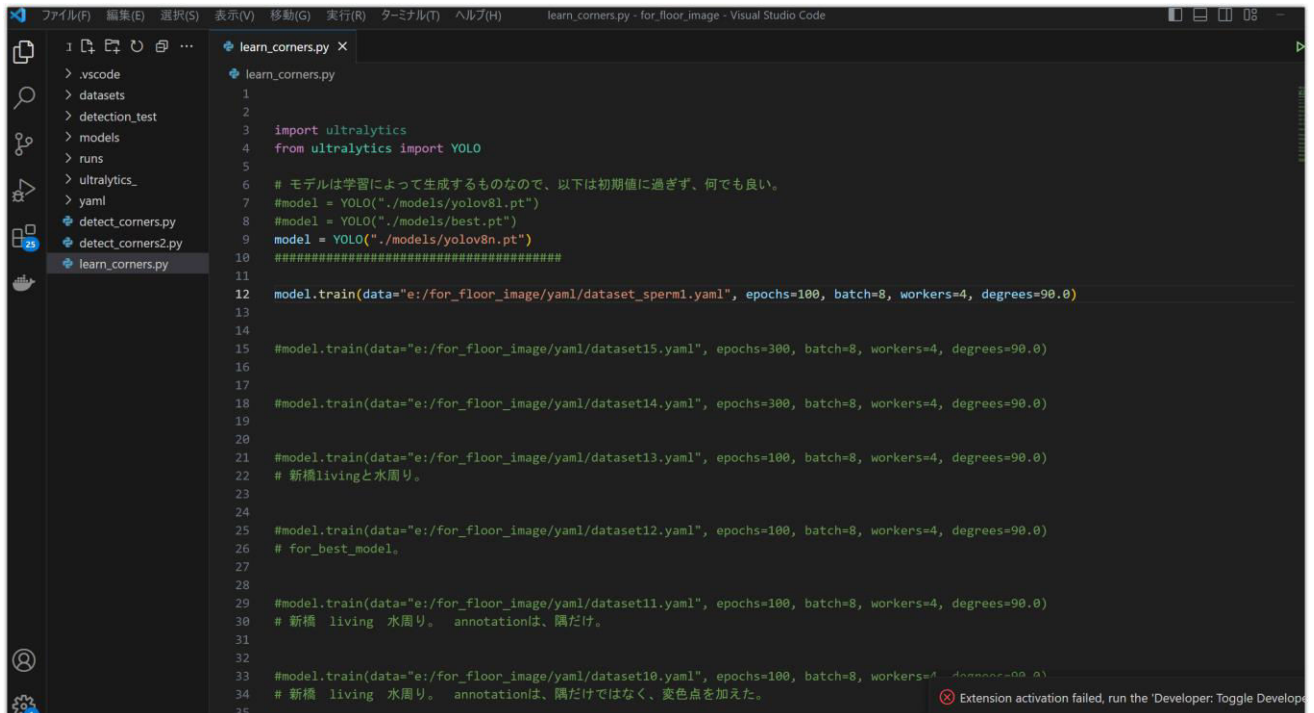


## 9.2 for\_floor\_image を選ぶ



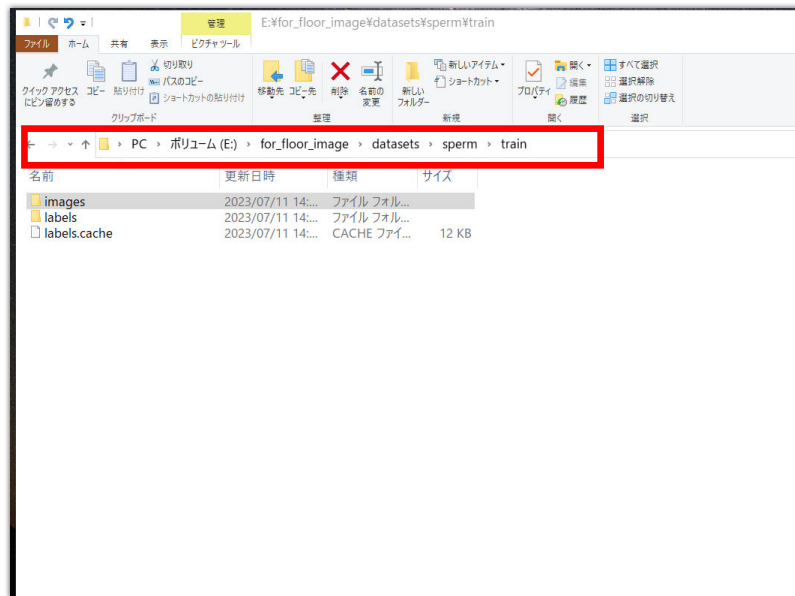
### 9.3 yolo v8 の起動直後の画面

設定や、プロジェクトや、その他都合によって色々ですが、弊社の場合の画面は以下になります。



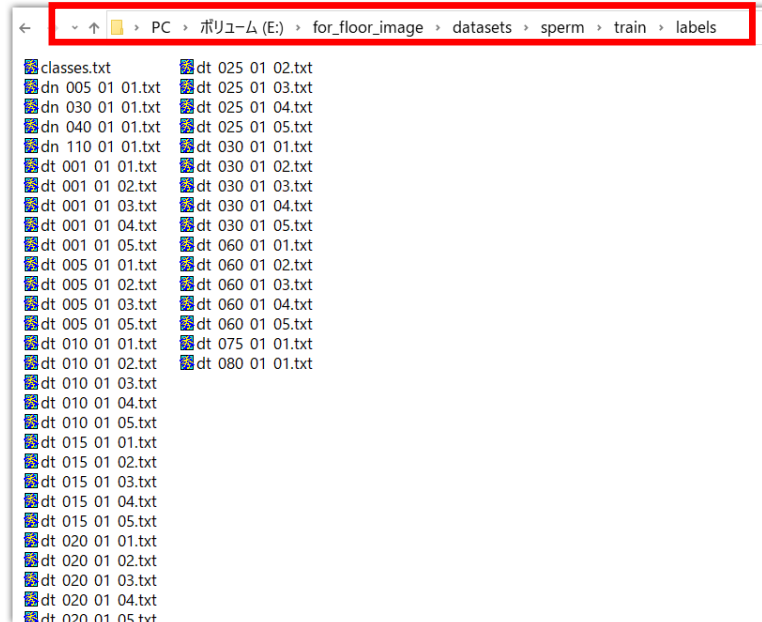
```
learn_corners.py
1
2
3 import ultralytics
4 from ultralytics import YOLO
5
6 # モデルは学習によって生成するものなので、以下は初期値に過ぎず、何でも良い。
7 #model = YOLO("./models/yolov8l.pt")
8 #model = YOLO("./models/best.pt")
9 model = YOLO("./models/yolov8n.pt")
10 #####
11
12 model.train(data="e:/for_floor_image/yaml/dataset_sperml.yaml", epochs=100, batch=8, workers=4, degrees=90.0)
13
14 #model.train(data="e:/for_floor_image/yaml/dataset15.yaml", epochs=300, batch=8, workers=4, degrees=90.0)
15
16 #model.train(data="e:/for_floor_image/yaml/dataset14.yaml", epochs=300, batch=8, workers=4, degrees=90.0)
17
18 #model.train(data="e:/for_floor_image/yaml/dataset13.yaml", epochs=100, batch=8, workers=4, degrees=90.0)
19
20 # 新橋 living と水周り。
21 #model.train(data="e:/for_floor_image/yaml/dataset12.yaml", epochs=100, batch=8, workers=4, degrees=90.0)
22 # for_best_model。
23
24
25 #model.train(data="e:/for_floor_image/yaml/dataset11.yaml", epochs=100, batch=8, workers=4, degrees=90.0)
26 # 新橋 living 水周り。 annotationは、隅だけ。
27
28
29 #model.train(data="e:/for_floor_image/yaml/dataset10.yaml", epochs=100, batch=8, workers=4, degrees=90.0)
30 # 新橋 living 水周り。 annotationは、隅だけではなく、変色点を加えた。
31
32
33
34
35
```

9.4 フォルダやファイルの配置が正しいことを確認します



382  
383  
384

### 9.5 annotation ファイル



385  
386



387  
388 9.6 サンプル課題で学習する内容  
389  
390

391 9.6.1 区別する内容  
392

393 sperm  
394 gomi  
395  
396  
397

398 9.6.2 ある画像の中の sperm の位置  
399

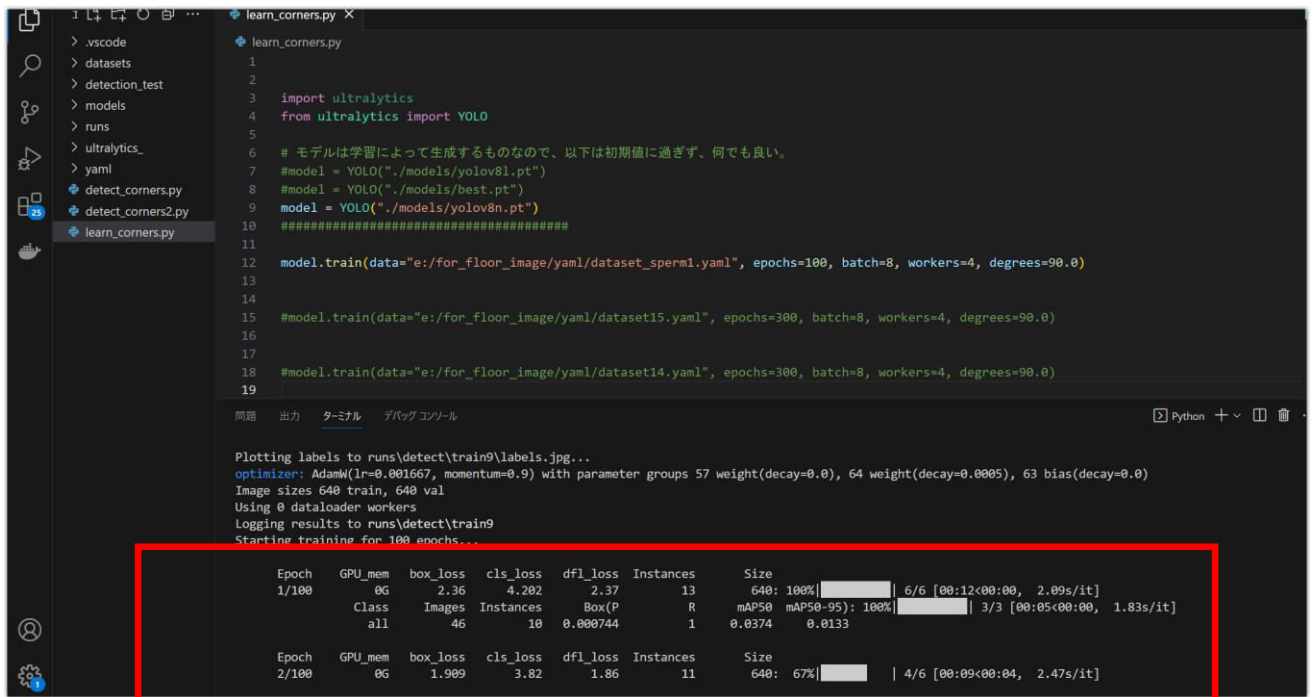
400 0 0.504687 0.475000 0.240625 0.231250  
401  
402  
403  
404  
405

406 9.6.3 ある画像の中の gomi の位置  
407

408 1 0.748958 0.150000 0.168750 0.166667  
409  
410  
411  
412  
413  
414

9.7 learn\_corners.py を実行します

ファイル類が正しい位置にあれば、以下のように iteration (繰り返しの計算) が起こります。



```

1
2
3 import ultralytics
4 from ultralytics import YOLO
5
6 # モデルは学習によって生成するものなので、以下は初期値に過ぎず、何でも良い。
7 #model = YOLO("./models/yolov8l.pt")
8 #model = YOLO("./models/best.pt")
9 model = YOLO("./models/yolov8n.pt")
10 #####
11
12 model.train(data="e:/for_floor_image/yaml/dataset_sperml.yaml", epochs=100, batch=8, workers=4, degrees=90.0)
13
14
15 #model.train(data="e:/for_floor_image/yaml/dataset15.yaml", epochs=300, batch=8, workers=4, degrees=90.0)
16
17
18 #model.train(data="e:/for_floor_image/yaml/dataset14.yaml", epochs=300, batch=8, workers=4, degrees=90.0)
19
  
```

Plotting labels to runs\detect\train9\labels.jpg...

optimizer: AdamW(lr=0.001667, momentum=0.9) with parameter groups 57 weight(decay=0.0), 64 weight(decay=0.0005), 63 bias(decay=0.0)

Image sizes 640 train, 640 val

Using 0 dataloader workers

Logging results to runs\detect\train9

Starting training for 100 epochs...

Epoch	GPU_mem	box_loss	cls_loss	dfl_loss	Instances	Size
1/100	0G	2.36	4.282	2.37	13	640: 100%   6/6 [00:12<00:00, 2.09s/it]
	Class	Images	Instances	Box(P)	R	mAP50 mAP50-95): 100%   3/3 [00:05<00:00, 1.83s/it]
	all	46	10	0.000744	1	0.0374 0.0133
Epoch	GPU_mem	box_loss	cls_loss	dfl_loss	Instances	Size
2/100	0G	1.909	3.82	1.86	11	640: 67%   4/6 [00:09<00:04, 2.47s/it]

## 9.8 結果の確認

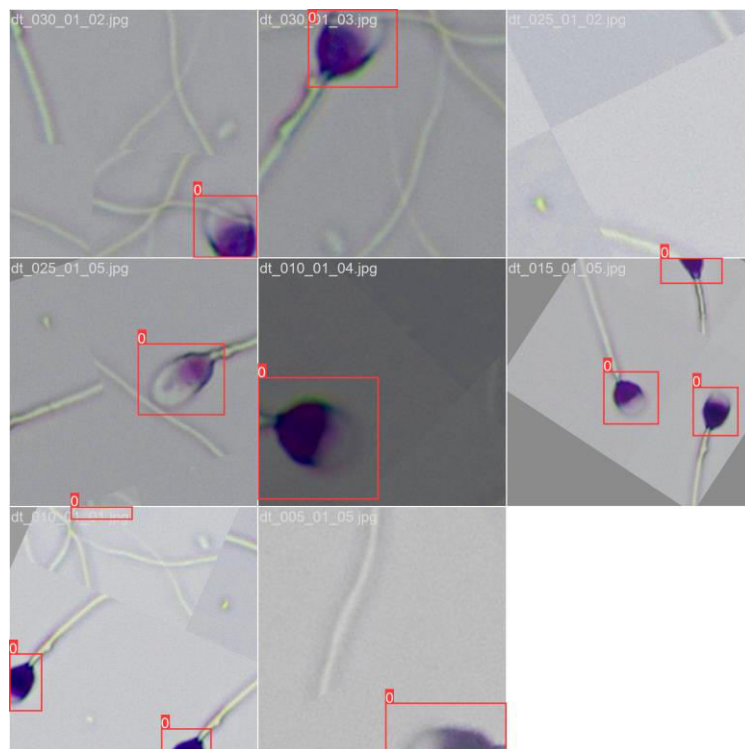
for\_floor\_image¥runs¥detect¥train(ここでは設定に依存する文字列)のなかを見て、学習(train)と確認(validation)が起きているかを確認します

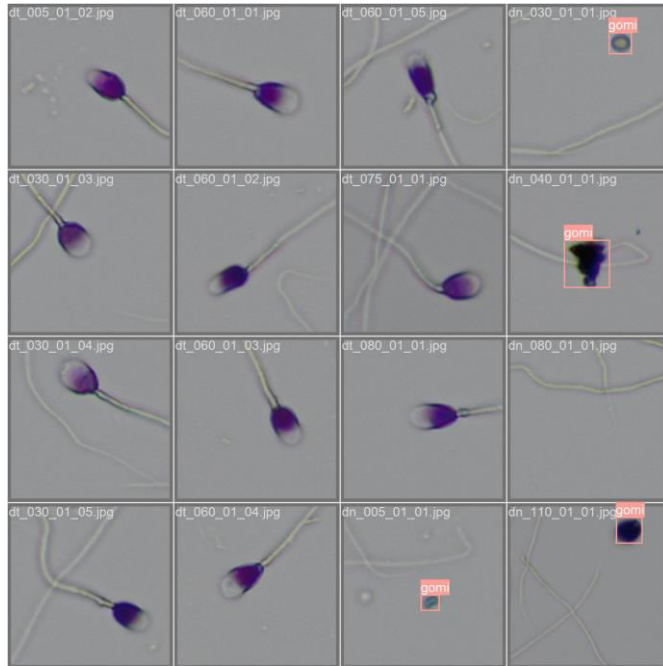
### 9.8.1 学習の過程(sperm と gomi)

この例では、

0 sperm  
1 gomi

を学習しています。



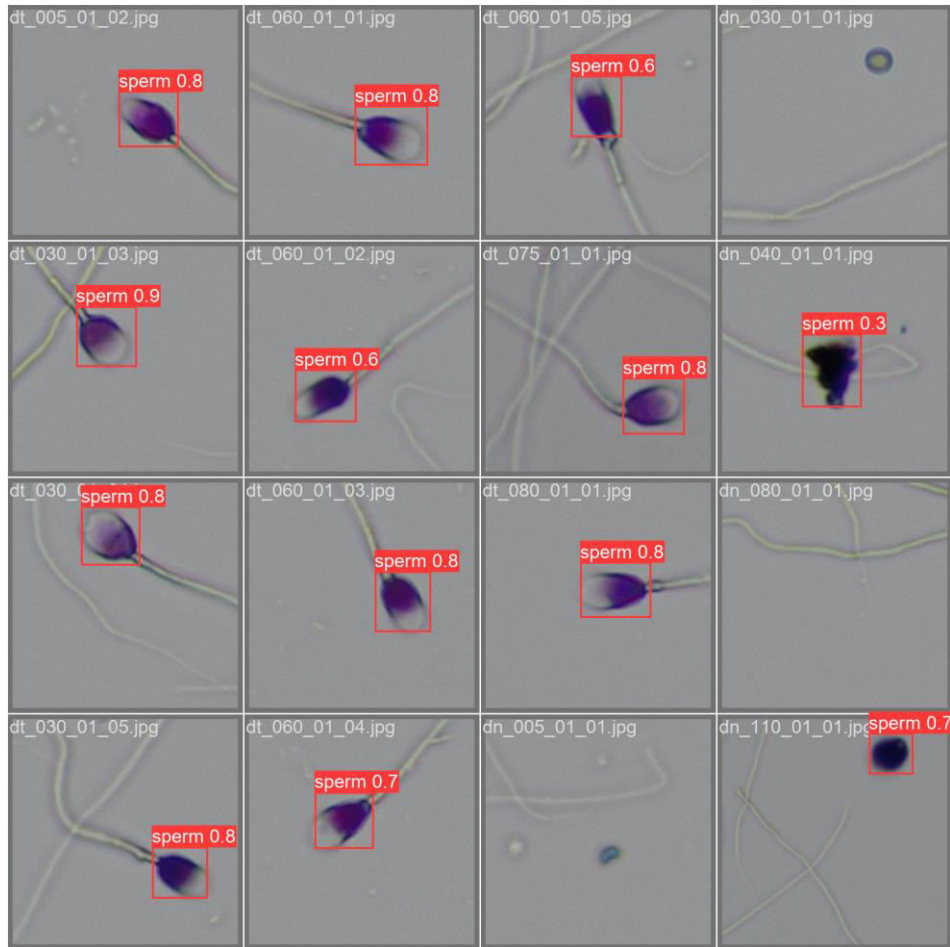


456

457  
 458  
 459  
 460  
 461

### 9.8.2 学習の結果 (sperm)

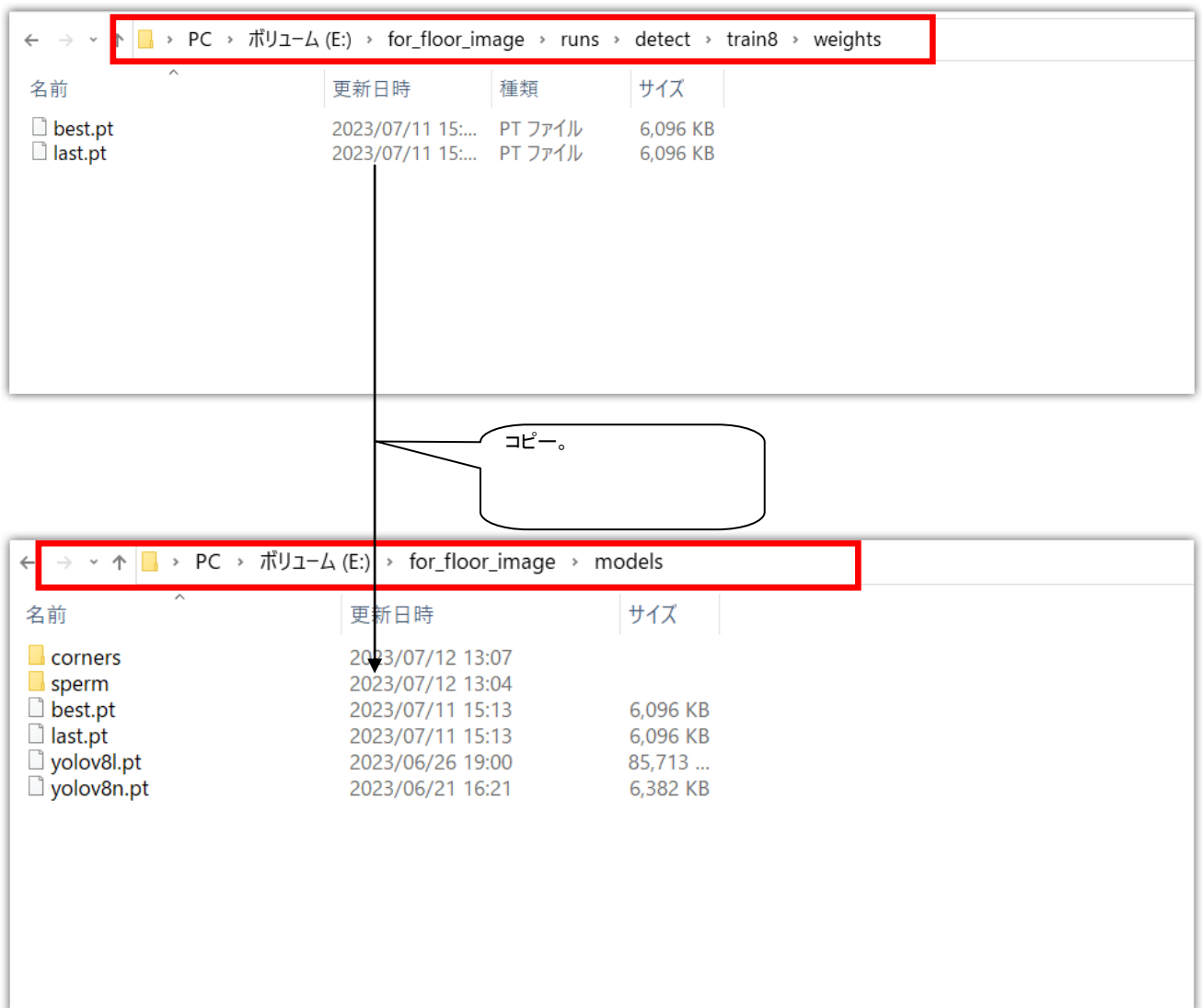
確信度とともに、結果が表示されます。



462  
 463  
 464  
 465  
 466  
 467  
 468  
 469  
 470

9.9 モデル(学習の結果できた NN の重みの集合)のコピー

モデル(学習の結果できた NN の重みの集合)をコピーします。



487  
488  
489  
490  
491 10. yolo v8 による、サンプル課題の認識  
492  
493

494 10.1 認識の対象となる画像たちを用意する  
495

496 detection\_test の中に、認識の対象となる画像たちを入れます。  
497  
498



501  
502  
503 10.2 認識の結果  
504

505 folder の名前や、ついている番号は設定に依存します。  
506

