

簡易的画像分類器(AI)の作製と推論マニュアル

概要 (p2, 3)

1. Google Colaboratoryでの実行 (p4～)

- 1.1 画像の準備 (p5)
- 1.2 訓練用画像フォルダの準備(p6)
- 1.3 テスト用画像フォルダの準備(p7, 8)
- 1.4 Google driveにアップロードする (p9)
- 1.5 Colaboratoryで実行する (p10～p15)
- 1.6 出力結果を見る (p16～p19)
- 1.7 保存済みモデルを使って推論する(p20)
- 1.8 (補) 画像収集のグループワーク (p21～23)

2. ローカルPC上での実行 (p24～)

- 2.1 Spyderを使えるようにする (p25)
- 2.2 Spyderを起動する (p26, 27)
- 2.3 必要なパッケージのインストール (p28)
- 2.4 実行前の準備 (p29)
- 2.5 実行・結果 (p30)

概要-1

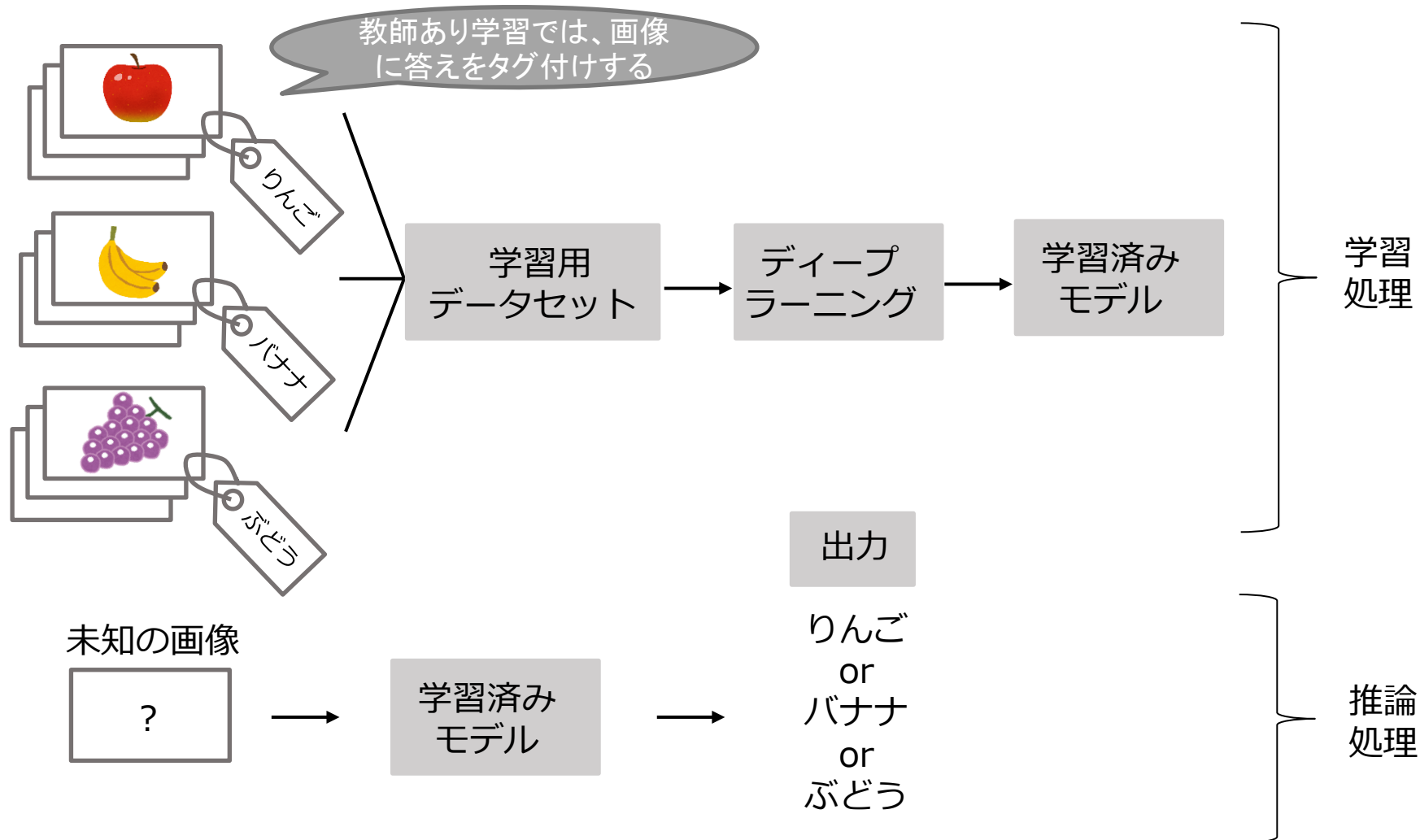
自分で収集した画像を分類し、フォルダに分けておくことで（教師データの作製）画像分類できるAIを簡易的に作製できる。

また、そのAIによって、推論を行うことができ、
(1) 答えをアノテーションしてあるものを正誤判定
(2) 画像に対して推論
の2通りが行える。

また、画像分類器（モデル）は保存しておくことがで、
推論のみを後から実行することも可能。

概要-2

例：果物の画像認識の学習器作成の流れ（教師あり学習）



1. Google Colaboratoryで実行する

<https://colab.research.google.com/?hl=ja>

Googleが提供しているブラウザ上でpythonを記述、実行できるツール。

メリットとして

- 自身のPCに環境を構築する必要がない
- クラウド上のGPUを利用できる
- 簡単に共有できる

ことが挙げられる。一方で、データの転送にはインターネット回線を使うため、時間がかかる。

1.1 画像の準備

- 分類したい画像を収集する

現状 **jpeg**と**png** 形式に対応

少ない画像数でも機能はするが、各分類**100枚以上**を推奨

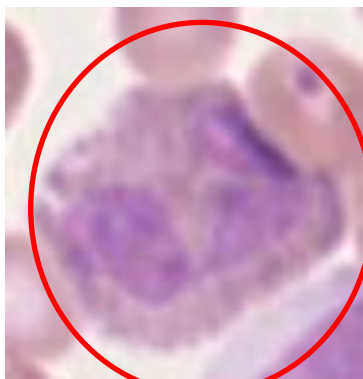
<ポイント>

できる限り、目的のもの以外は写り込まないようにする。

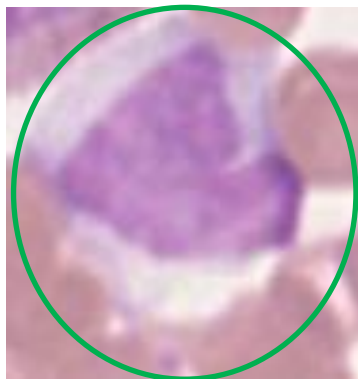
(おそらく) 正方形に近いほうがよい。

(例)

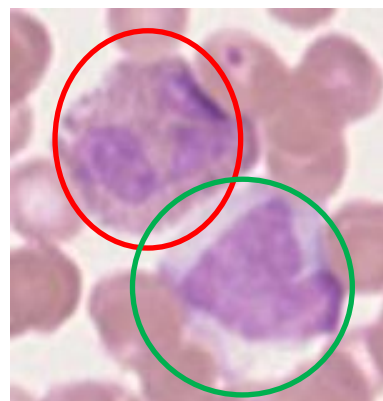
OK



OK



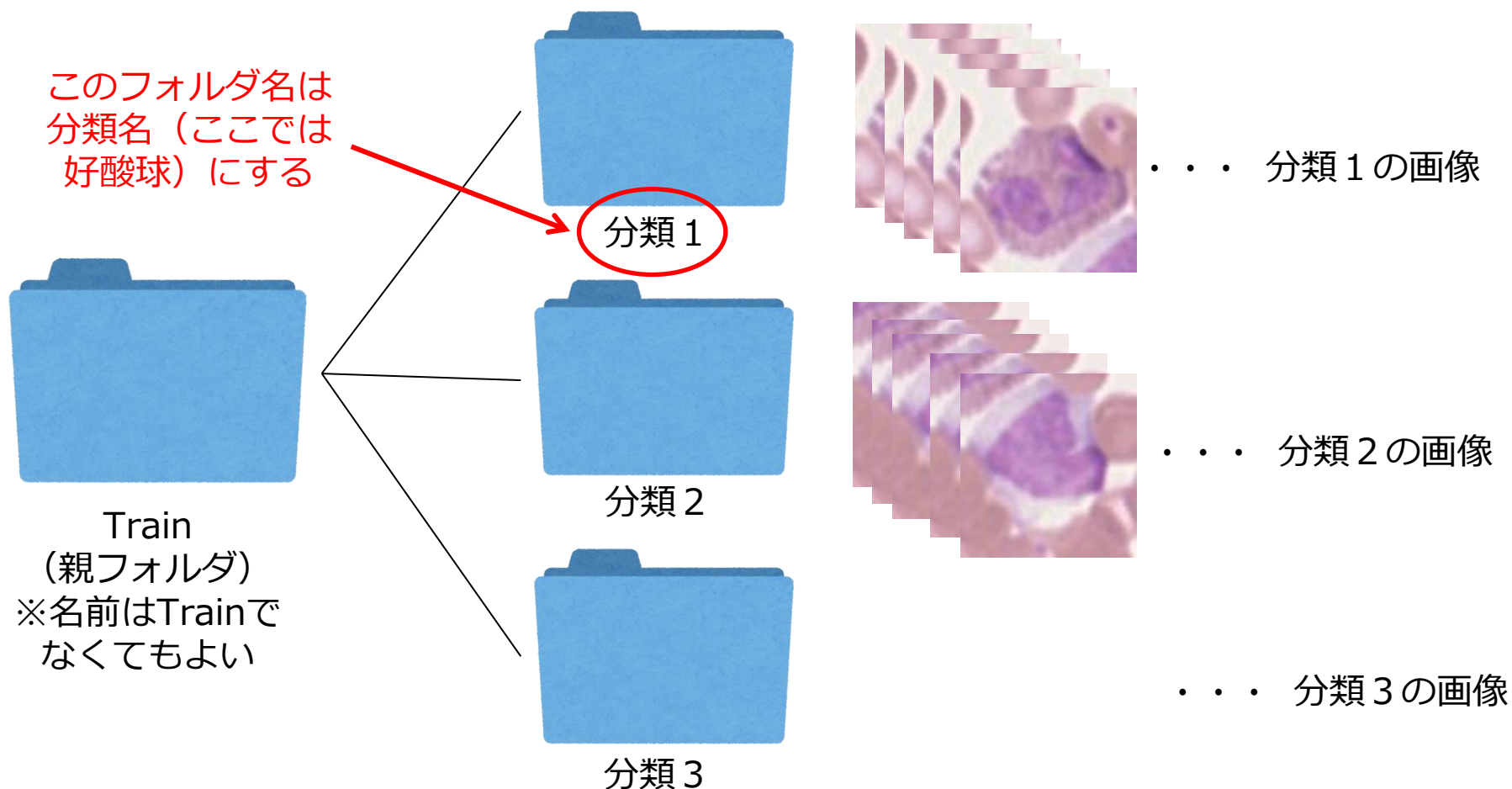
NG



赤は好酸球、緑は単球とよばれる共に白血球の仲間。後ろにみえるのは赤血球だが、これはどこでみられるので、入ってしまって構わない。

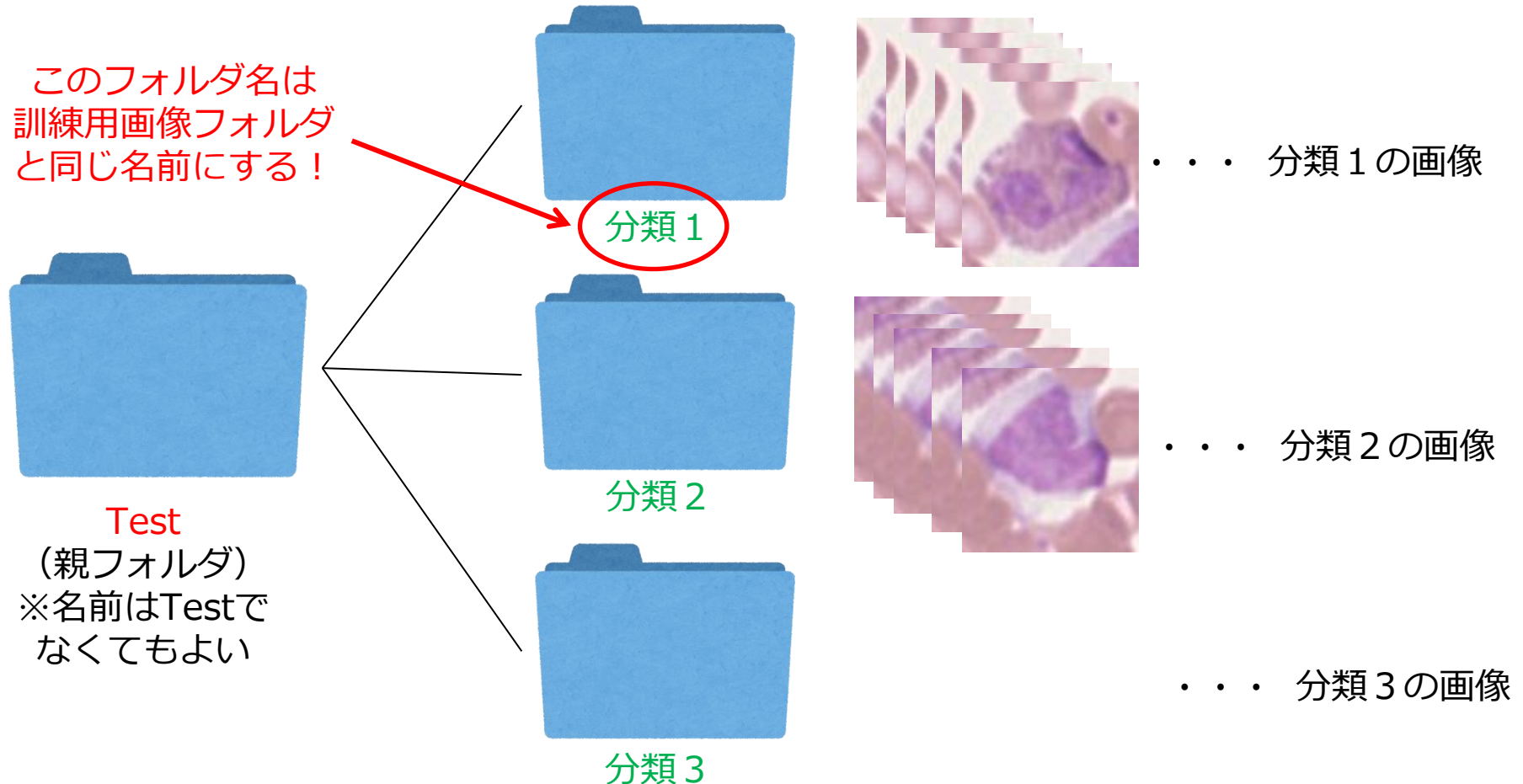
1.2 訓練用画像フォルダの準備

- ・ 自分で画像を分類してフォルダに分ける。
- ・ 各フォルダ100枚以上が好ましい



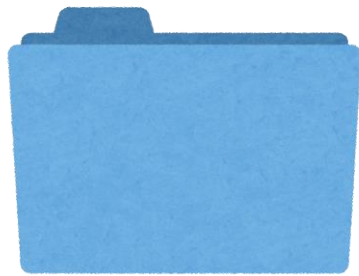
1.3.1 テスト用画像フォルダの準備（解答あり）

- ・ AIを新しい画像を見せ、正しく推論できるかを確認するため。各フォルダ、数枚。訓練用画像とは異なる画像を準備。

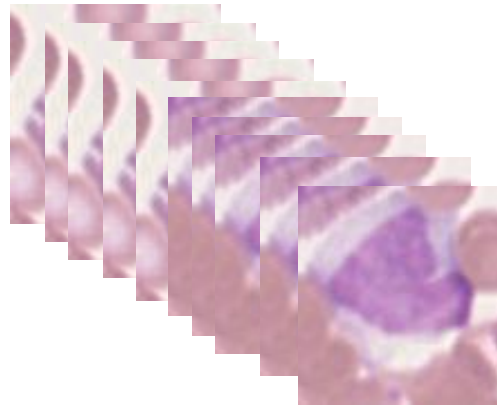


1.3.2 テスト用画像フォルダの準備（推論のみ）

- ・ AIを作ったあとに、新しい画像を見せて推論させる



Test
(親フォルダ)
※名前はTestでなくてもよい

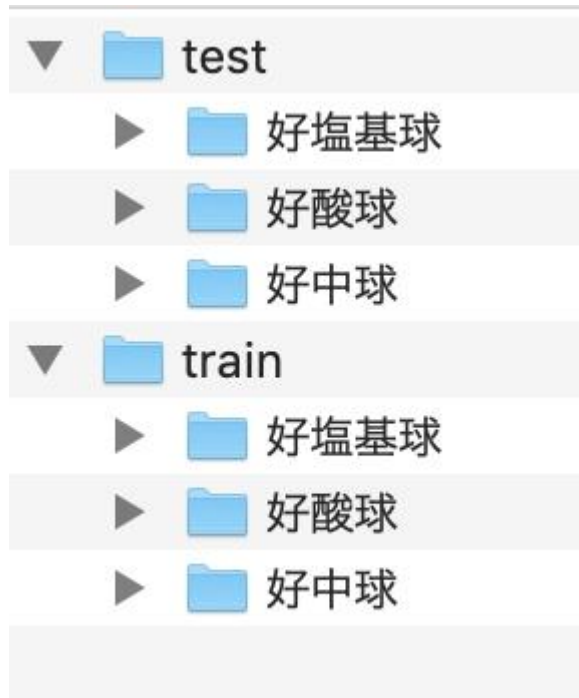


画像は分類せずに、親フォルダに
直接画像ファイルを入れる。

1.4 Google driveにアップロードする

PCの容量に余裕があるのであれば、ダウンロード版Googleドライブが使いやすくお勧め。

https://www.google.com/intl/ja_jp/drive/download/



アップロードしたらもう一度確認

画像分類器を作ったあとのAIによる推論が

(解答ありの場合)
こんな感じで、trainフォルダとtestフォルダに
同じ名前のフォルダがあればOK

(解答なしの場合)
testフォルダ内は直接画像ファイルになります。

1.5 Colaboratoryで実行する（1）

Google ColaboratoryでHistology.ipynbを開く

<https://colab.research.google.com/notebooks/intro.ipynb?hl=ja>

ファイル>ノートブックを開く



アップロードから
Histology.ipynbを選択

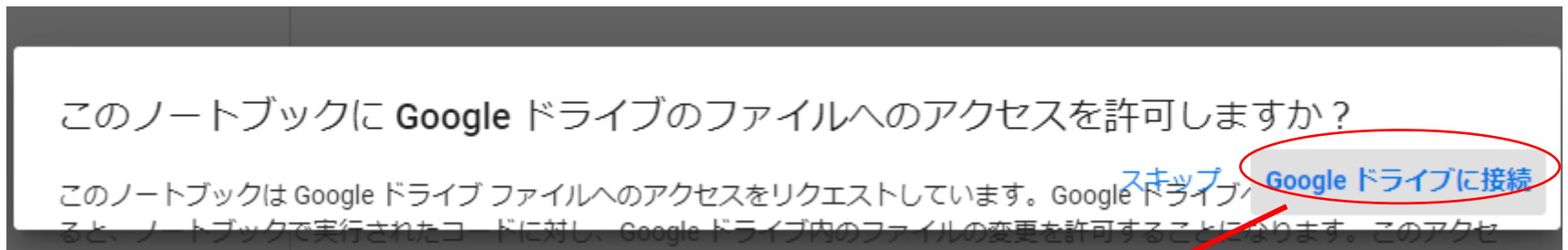
またはGoogle driveのColaboNotebook
のフォルダ内にファイルをいれておく

1.5 Colaboratoryで実行する(2)

自分のGoogle driveをcolaboratoryにマウントする



(1) 再生ボタンをクリックする



(2) Google ドライブに接続をクリック

1.5 Colaboratoryで実行する(3)

自分のGoogle driveをcolaboratoryにマウントする (続き)



Google Drive for desktop が
Google アカウントへのアクセス
をリクエストしています

Google Drive for desktop を信頼できることを確認

お客様の機密情報をこのサイトやアプリと共有することがあります。アクセス権の確認、削除は、[Google アカウント](#)でいつでも行えます。

Google で [データ共有を安全に行う方法](#) についての説明をご覧ください。

Google Drive for desktop の [プライバシー ポリシー](#) と [利用規約](#) をご覧ください。

キャンセル

許可

(3) アカウントを選択し、クリック

```
1 ##自分のgoogle driveをgoogle colaborylに接続
2 ##実行すると、別windowが開き、googleアカウントへのアクセスを許可するか表示される
3 ##許可すると、権限を与えるためのkeyが表示される。コピーし、この下に出てくるkeyを入れるところにペーストし
4 #Enterキーを押す。
5 #"/content/drive"が表示されれば成功
6
7 from google.colab import drive
8 drive.mount('/content/drive')
```

Mounted at /content/drive

(4) 許可をクリック

(5)これができれば成功

1.5 Colaboratoryで実行する(4)

フォルダ名など必要なパラメータを変更

20行目～36行目

```
20 #####
21 #汎用的に使えるようにするためにこの部分に課題ごとに変更が必要になるパラメータを集積する
22
23 train_data_dir = '/content/drive/MyDrive/train' #訓練用画像のフォルダdir
24
25 test_data_dir = '/content/drive/MyDrive/test' #作成したモデルによる推論を行いたい画像フォルダdir
26 Test_images_are_clasified = True #すでに人間が分類したテスト画像に対して推論する場合はTrue、画像に対して推論する場合はFalse
27
28 save_dir = '/content/drive/MyDrive/result' #最後に出力するjpegファイルを保存するdir
29 save_file_name = "prediction.jpg" #出力するjpegファイルの名前 #画像枚数が増えた場合 ***-2.jpg, ***-3.jpgとなる 形式の変更は不可
30
31 model_save = True #モデルを保存する場合はTrue、保存しない場合はFalseにする
32 only_make_model = False #モデルを作成するだけの場合はTrue、モデルによる推論まで行う場合はFalse
33
34 ##学習にかかわるパラメータ
35 epoch_num = 7 #学習のエポック数
36 #####
```

/content/drive/MyDrive/は自分のGoogle driveのパスとなるので、変更しなくてよい

train_data_dir: 訓練データの親フォルダの名前（1.2で作製したもの。ここではtrainとなっている）

test_data_dir: 推論したい画像フォルダ（1.3.1 or 1.3.2で作製したもの。ここではtest）

Test_images_are_clasified: 分類済み(1.3.1)の場合はTrueに。分類してない場合(1.3.2)はFalseにする

save_dir: 結果を保存するフォルダ（あらかじめ作ってある必要はない）

save_file_name: test内の画像を推論した結果を出力するファイル名。jpgのみ対応。

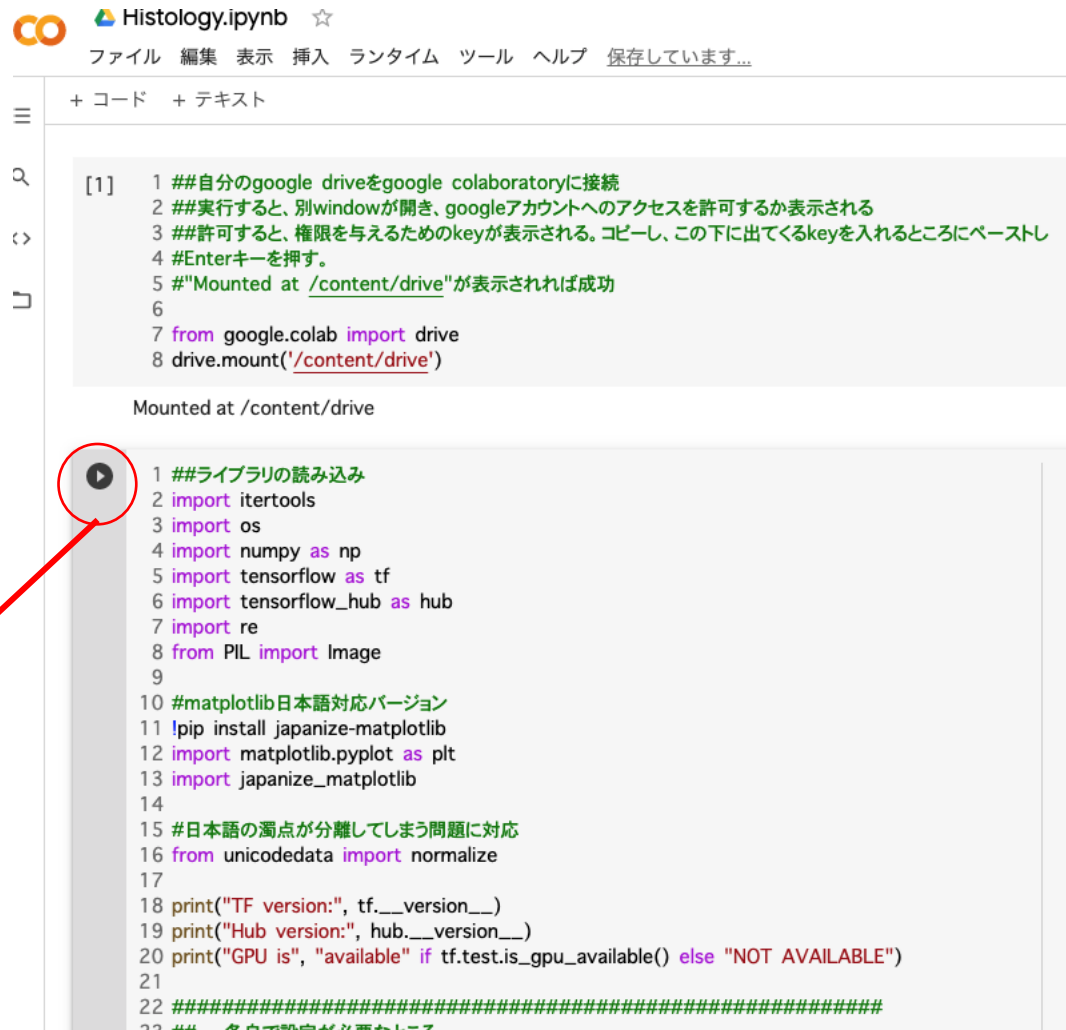
model_save : 作った画像分類器のモデルを保存したい場合True, 保存は必要ない場合はFalse

only_make_model : 画像分類器を作るだけで、推論が不要な場合はTrue, 推論する場合False

epoch_num: 機械学習の繰り返し数。よくわからなければとりあえず変えない。

1.5 Colaboratoryで実行する(5)

プログラムの実行（実行完了までに時間がかかります）



```
[1] 1 ##自分のgoogle driveをgoogle colaboratoryに接続
    2 ##実行すると、別windowが開き、googleアカウントへのアクセスを許可するか表示される
    3 ##許可すると、権限を与えるためのkeyが表示される。コピーし、この下に出てくるkeyを入れるところにペーストし
    4 #Enterキーを押す。
    5 #"Mounted at /content/drive"が表示されれば成功
    6
    7 from google.colab import drive
    8 drive.mount('/content/drive')
```

Mounted at /content/drive

```
[2] 1 ##ライブラリの読み込み
    2 import itertools
    3 import os
    4 import numpy as np
    5 import tensorflow as tf
    6 import tensorflow_hub as hub
    7 import re
    8 from PIL import Image
    9
   10 #matplotlib日本語対応バージョン
   11 !pip install japanize-matplotlib
   12 import matplotlib.pyplot as plt
   13 import japanize_matplotlib
   14
   15 #日本語の濁点が多分点になってしまう問題に対応
   16 from unicodedata import normalize
   17
   18 print("TF version:", tf.__version__)
   19 print("Hub version:", hub.__version__)
   20 print("GPU is", "available" if tf.test.is_gpu_available() else "NOT AVAILABLE")
   21
   22 #####
   23 ## 各自で設定が必要ケース
```

クリックする

1.5 Colaboratoryで実行する(6)

実行中、画面の下の方に行くと、進んでいる様子が見えます。

```
Using https://tfhub.dev/google/imagenet/mobilenet\_v3\_small\_100\_224/feature\_vector/5 with input size (224, 224)
Found 111 images belonging to 4 classes.
Found 457 images belonging to 4 classes.
Building model with https://tfhub.dev/google/imagenet/mobilenet\_v3\_small\_100\_224/feature\_vector/5
Model: "sequential"
```

Layer (type)	Output Shape	Param #
keras_layer (KerasLayer)	(None, 1024)	1529968
dropout (Dropout)	(None, 1024)	0
dense (Dense)	(None, 4)	4100

```
=====
Total params: 1,534,068
Trainable params: 4,100
Non-trainable params: 1,529,968
```

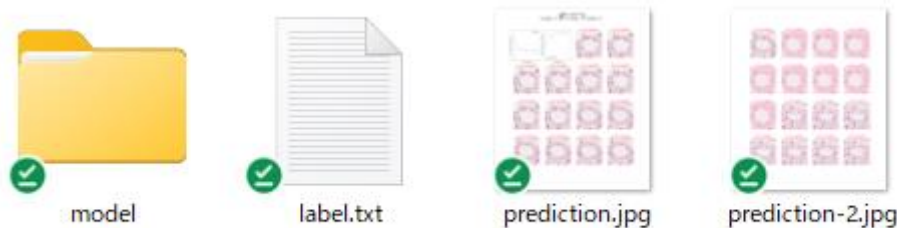
```
Epoch 1/7
3/14 [====>.....] - ETA: 15s - loss: 1.5713 - accuracy: 0.3425
```

この際に、何かエラーが出たらここに表示されます。
例えば、指定したフォルダ名が見つからない、などが出る場合があります。
エラーメッセージに従って、タイプミスなどないか、確認してください

1.6 出力結果を見る(1)

Resultフォルダの中に以下のようなファイルができています
(modelの保存を選んでいない場合はprediction.jpgのみ)

推論する画像が多い場合、jpgファイルは下記のように-2.jpg, -3.jpgのようにファイル数が増えます。



modelのフォルダ内には以下の4つが作られています。

 assets

 variables

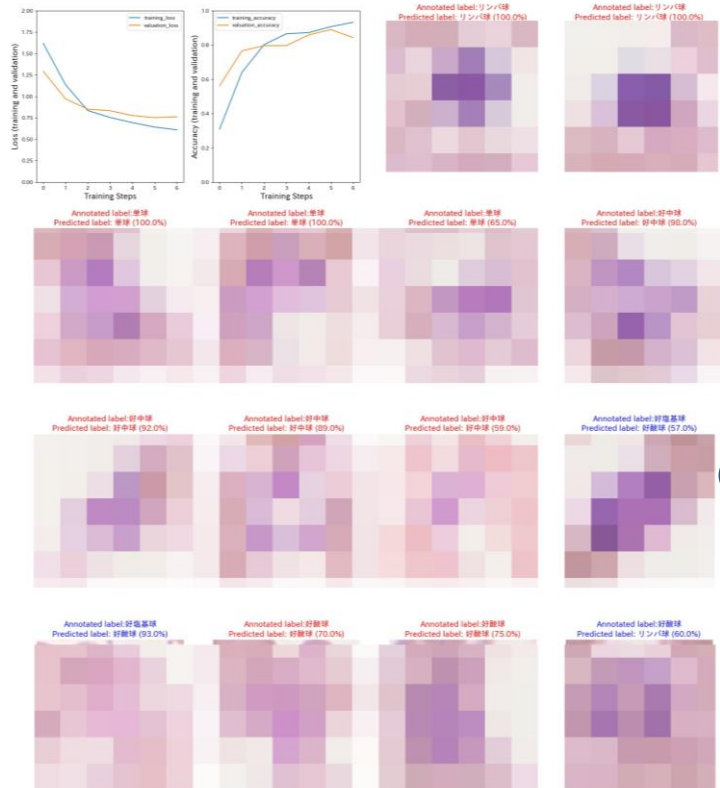
 keras_metadata.pb

 saved_model.pb

1.6 出力結果を見る(2)

推論用の画像が分類済み（1.3.1）の場合(1)

班番号 : 1
正答率 = 11/14 (78.0%)
画像数 (トータル) : 430
リンパ球: 100 単球: 104 好中球: 101 好塩基球: 28 好酸球: 97



正答率 : 教員が用意した画像をAIに認識させた時の
(教員がつけたラベルに対する) 正答率

画像数 : 学習に用いた画像の数
最下段 : それぞれのグループの画像数

Annotated label: 好酸球
Predicted label: リンパ球 (60.0%)

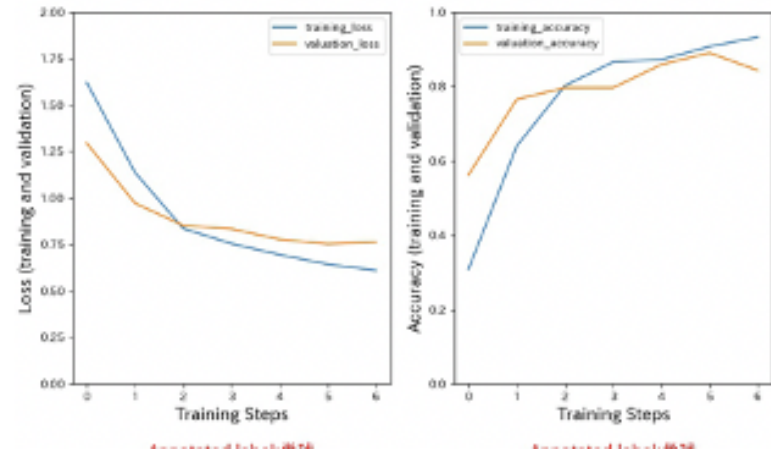
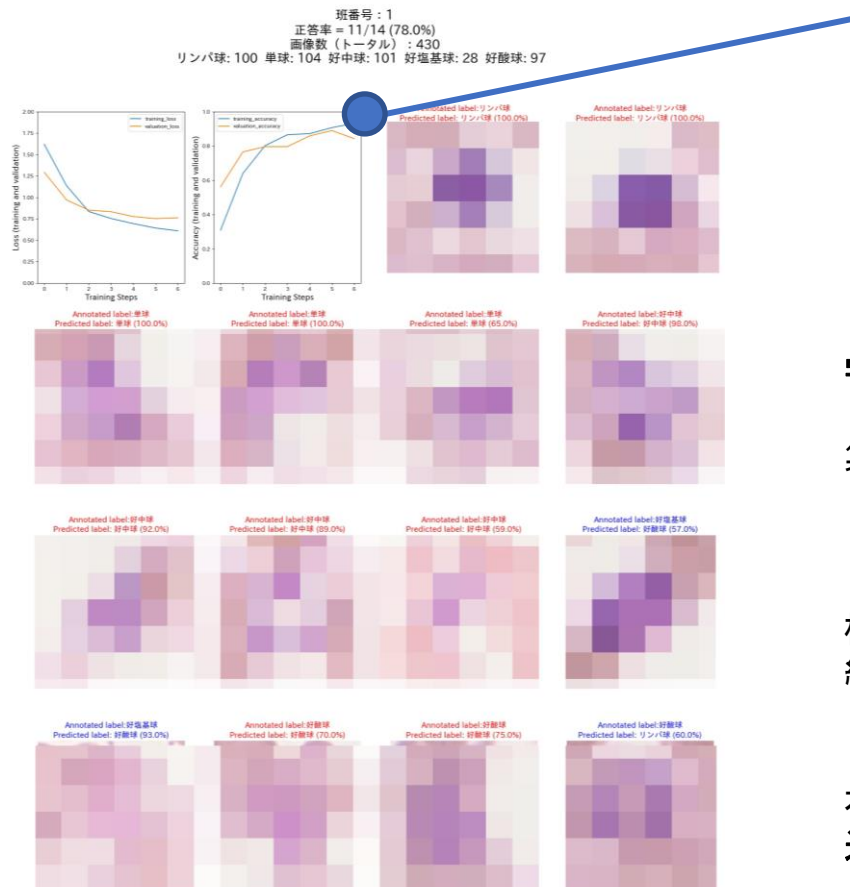
教員のラベル
AIの推測 (確率)

赤字は正解
青字は不正解

教員の用意した画像

1.6 出力結果を見る(3)

推論用の画像が分類済み（1.3.1）の場合(2)



学習のプロット

集めた画像は学習用の画像と、評価用の画像に分けられる

- ・ 青線：学習の結果
- ・ 黄色線：評価用画像で、そのモデルを使用した時の結果

横軸：学習のステップ数

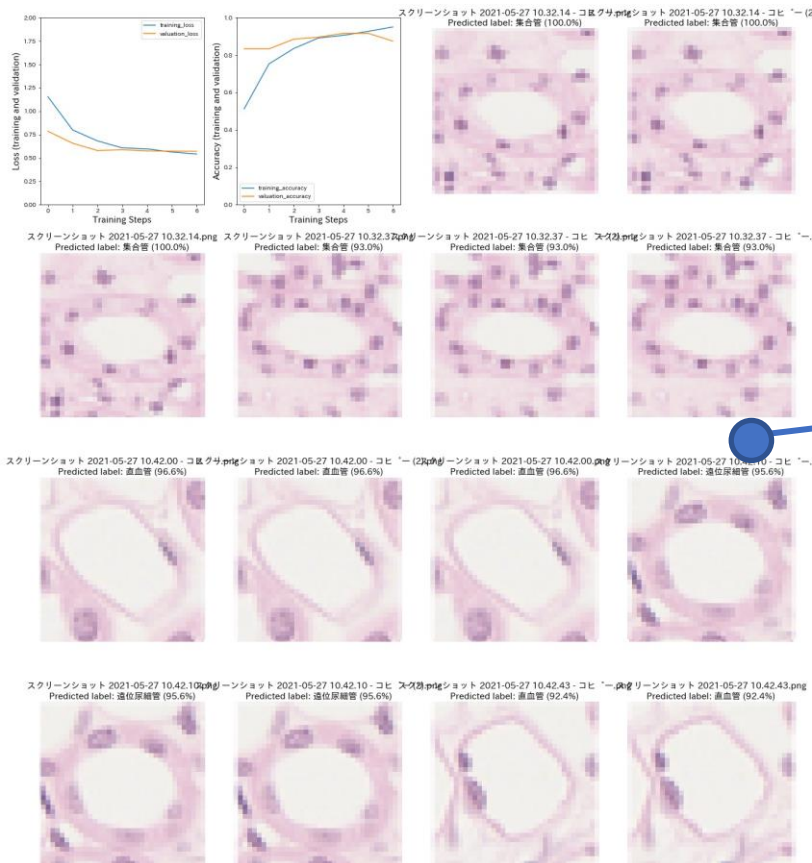
縦軸：（左）損失
（右）正確性

右図で青線のほうが黄色線よりも正確性が高くなるのは
過学習（学習用のサンプルに最適化されすぎることを表す

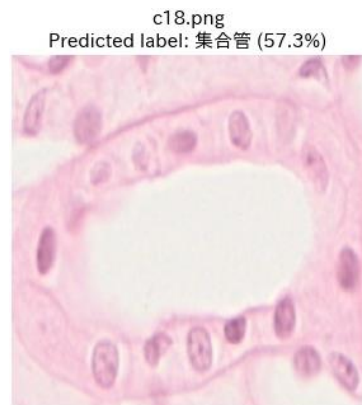
1.6 出力結果を見る(4)

推論用の画像を分類していない(1.3.2) 場合

訓練用画像数(トータル): 568
遠位尿管: 153 集合管: 148 直血管: 139 中間尿管: 128



訓練画像数: 学習に用いた画像の数
下段: それぞれのグループの画像数



画像ファイル名
予測分類 (%)

1.7 保存済みのモデルを使って推論する

model_test.ipynbをgoogle colaboratoryで実行する

P13を参照し、model_test.ipynb内の下記の部分を書き換えて同様に実行する。

```
28 #####
29 #汎用的に使えるようにするためにこの部分に課題ごとに変更が必要になるパラメータを集積する
30
31 model_saved_dir = '/content/drive/MyDrive/result' #モデルを保存したフォルダ
32 test_data_dir = '/content/drive/MyDrive/test' #推論したい画像のフォルダ
33 save_dir = '/content/drive/MyDrive/result' #jpegファイルを保存するフォルダ
34 save_file_name = "prediction5.jpg" #出力のjpegの名前。保存する際に上書きされないように、名前を変更できるようにここに設置
35
36 Test_images_are_classified = True #画像を分類済みの場合は"True"。分類していない画像の推測のみの場合は"False"
37 #####
```

結果は、1.6と同様に出力される。

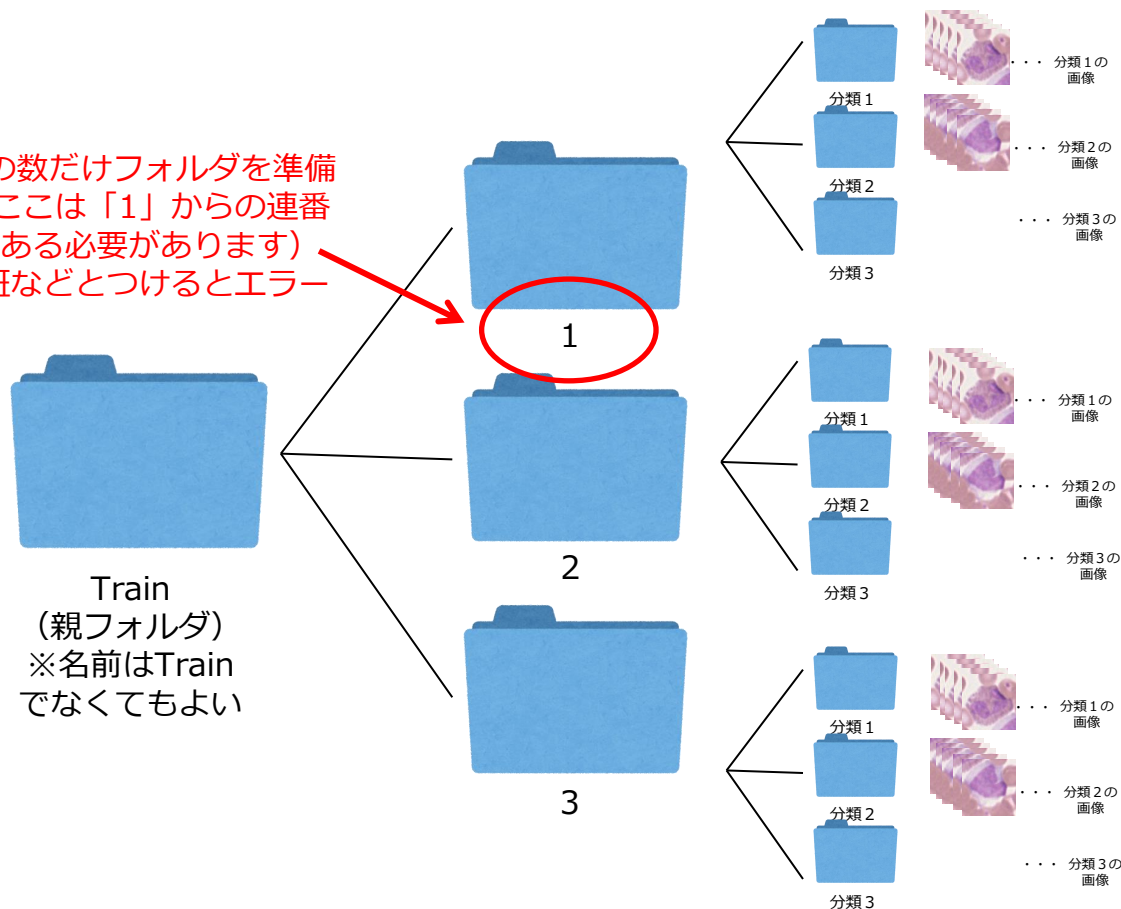
1.8 (補) 画像収集のグループワーク

1. 画像収集フォルダを作製する

—親フォルダの1つ下に班ごとのフォルダを作製

—推論用画像フォルダはこれまでと同じ

班の数だけフォルダを準備
(ここは「1」からの連番
である必要があります)
1班などにつけるとエラー



1.8 (補) 画像収集のグループワーク

2. Colaboratoryで実行する

—group.ipynbを使う。下記の部分のみ必要に応じて編集

```
32 #####
33 #汎用的に使えるようにするためにこの部分に課題ごとに変更が必要になるパラメータを集積する
34 number_of_groups = 4 ##班の数
35
36 train_data_dir = '/content/drive/MyDrive/group_train' #訓練用画像のフォルダdir
37
38 test_data_dir = '/content/drive/MyDrive/group_test' #作成したモデルによる推論を行いたい画像フォルダdir
39 Test_images_are_clasified = True #すでに人間が分類したテスト画像に対して推論する場合はTrue、画像に対して推論する場合はFalse
40
41 save_dir = '/content/drive/MyDrive/result' #最後に出力するjpegファイルを保存するdir
42
43 model_save = True #モデルを保存する場合はTrue、保存しない場合はFalseにする
44 only_make_model = False #モデルを作成するだけでの場合はTrue、モデルによる推論まで行う場合はFalse
45
46 ##学習にかかわるパラメータ
47 epoch_num = 7 #学習のエポック数
48 #####
```

number_of_group が新たに追加。用意したフォルダの数を入力

(保存名は選択できない)

→実行する。

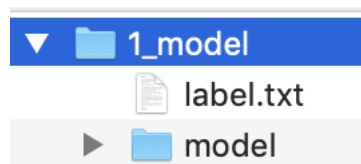
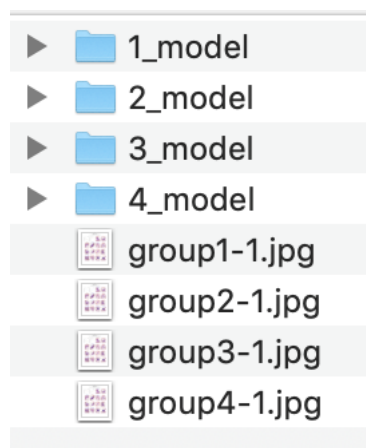
1.8 （補）画像収集のグループワーク

3. 結果

Colaboratoryのコンソールには最後に以下のような出力
正確性が高い順番でソートし、班番号、正確性、画像の数が表示される

```
Group No.: 3 Accuracy: 85.7% 564 images  
Group No.: 1 Accuracy: 78.6% 430 images  
Group No.: 4 Accuracy: 64.3% 189 images  
Group No.: 2 Accuracy: 57.1% 134 images  
elapsed_time:780.9778685569763[sec]
```

保存フォルダ内には、それぞれの班のモデルと、結果のjpgが保存される。



各班のモデルフォルダの中には
モデルとlabel.txtが保存される。

結果ファイルの見方についてはこれまでと同様。

2. ローカルPCでの実行

Pythonの統合開発環境(IDE)のインストールが必要
ここではSpyderを例として紹介する

メリット：画像のアップロードにかかる時間が短縮できる

デメリット：自身のPCの環境に依存するため、
ある程度知識が必要。

2.1 Spyderを使えるようにする

Windows, Macによって異なるが、下記をインストールすると、中にSpydeが含まれる。

(Windows)

Winpythonをダウンロードし、インストールする

<https://www.kkaneko.jp/tools/win/winpython.html>

<https://www.python.jp/install/anaconda/macos/install.html>

(Mac)

Anacondaをダウンロードし、インストールする

<https://www.python.jp/install/anaconda/macos/install.html>

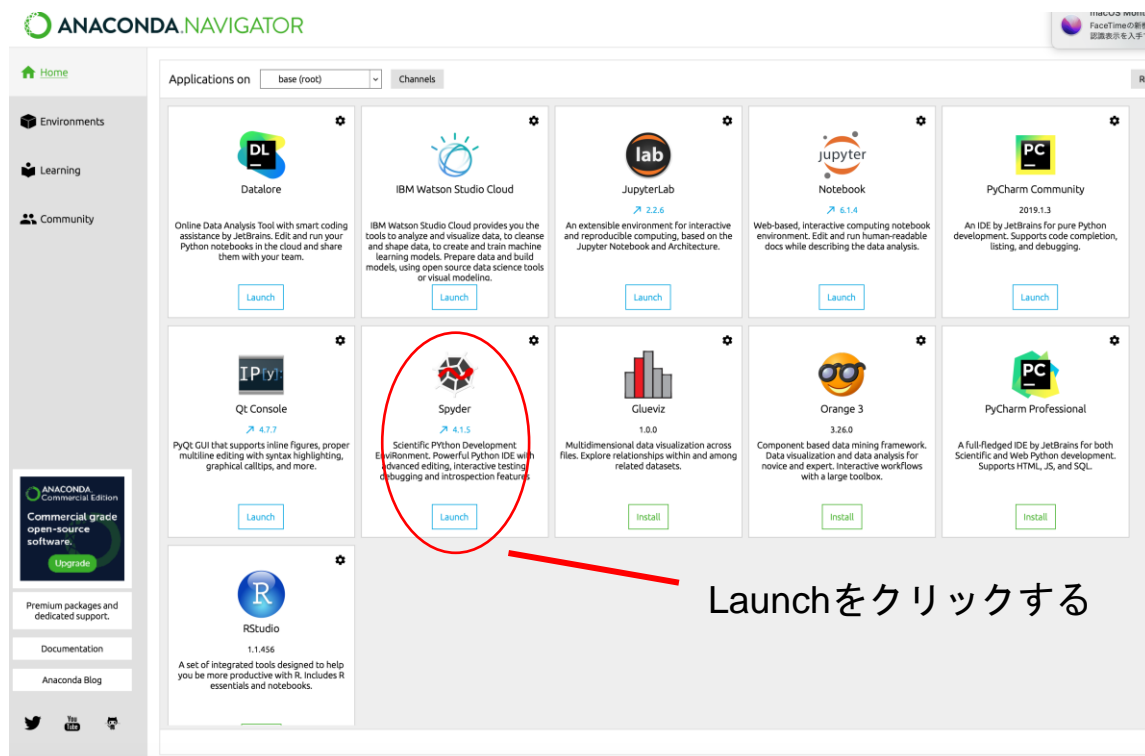
2.2 Spyderを起動する

(Windows)

Winpythonのフォルダ中にSpyder.exeが含まれる

(Mac)

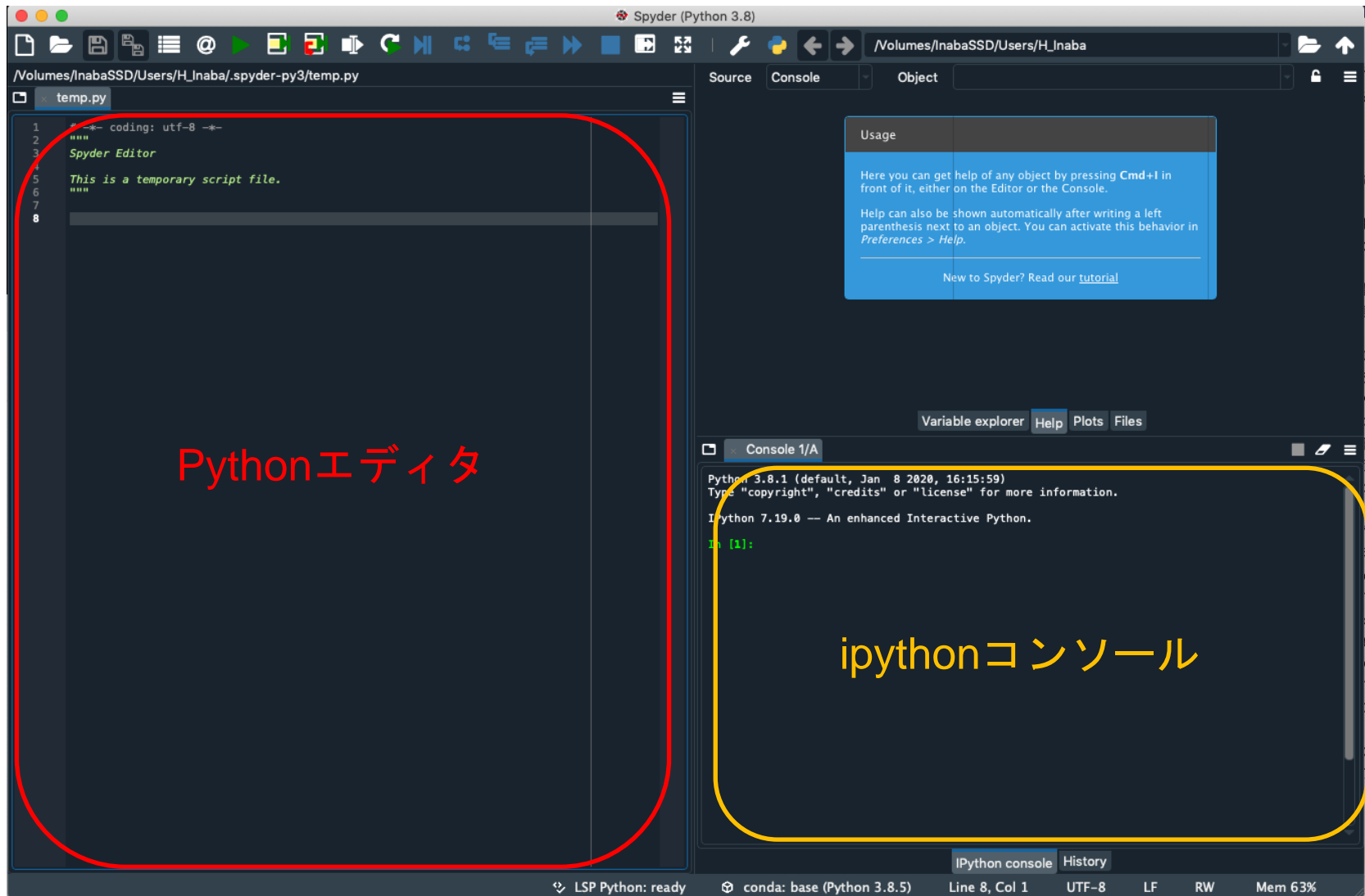
Anaconda-Navigatorを起動する



Launchをクリックする

2.3 Spyderを起動する

起動画面



2.4 必要なパッケージのインストール

パッケージ：ある（再利用可能な）機能のpythonコードをまとめたもの。

ここでは、tensorflow, tensorflow_hub, japanize_matplotlibをインストールする必要がある。

Spyderのipythonコンソールでpip install（パッケージ名）といれてEnterを押すと、（少し時間がかかるが）インストールされる。

```
Python 3.8.1 (default, Jan  8 2020, 16:15:59)
Type "copyright", "credits" or "license" for more information.

IPython 7.19.0 -- An enhanced Interactive Python.

In [1]: pip install tensorflow
```

同様に

```
pip install tensorflow_hub
pip install japanize_matplotlib
```

を実行する。

2.4 実行前の準備

画像ファイルや、画像フォルダの構成については、Google colaboratoryと同じ。ただし、Google driveではなくて、自分のマイドキュメントなどでOK



Open Fileからpythonファイルを開く→pythonエディタにプログラムが表示される。

必要に応じてパスなどを書き換える（Google colaboratoryの頁を参照）

```
#####  
#汎用的に使えるようにするためにこの部分に課題ごとに変更が必要になるパラメータを集積する  
  
train_data_dir = 'C:/Users/HIRONORI INBA/Google ドライブ/train' #訓練用画像のフォルダdir  
  
test_data_dir = 'C:/Users/HIRONORI INBA/Google ドライブ/test' #作成したモデルによる推論を行い  
Test_images_are_clasified = True #すでに人間が分類したテスト画像に対して推論する場合はTrue、画像は  
  
save_dir = 'C:/Users/HIRONORI INBA/Google ドライブ/result' #最後に出力するjpegファイルを保存する  
save_file_name = "prediction.jpg" #出力するjpegファイルの名前 #画像枚数が増えた場合 ***-2.jpg  
  
model_save = False #モデルを保存する場合はTrue、保存しない場合はFalseにする  
only_make_model = False #モデルを作成するだけの場合はTrue、モデルによる推論まで行う場合はFalse  
  
##学習にかかわるパラメータ  
epoch_num = 7 #学習のエポック数  
#####
```

2.5 実行・結果

Run fileボタンを押す



結果については、Google colaboratoryと同じ出力

Ipython consoleに実行の様子や、エラー、結果などが表示される。