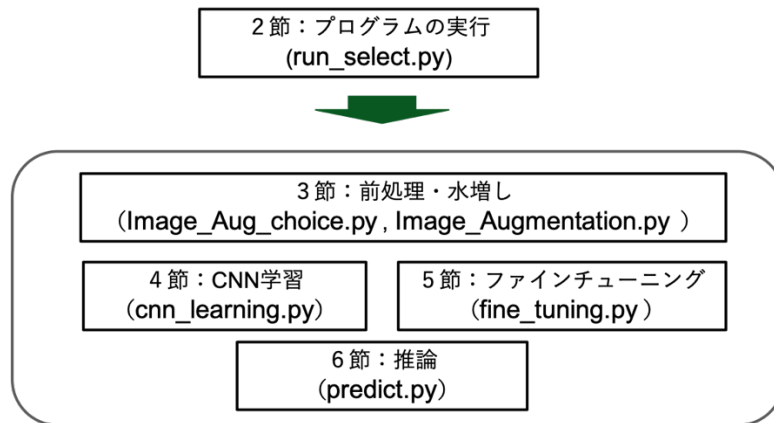


# 機械学習汎用システム仕様書

## 1. システム概要

本システムの構成は以下の通りである。



### (1) プログラムの実行

2 節において動作環境，および，プログラムを開始する方法について説明する。

### (2) 前処理・水増し

入力画像に対して，前処理・水増しを行う。

詳細については，3 節に記載する。

### (3) CNN ( Convolutional Neural Network) の学習

前処理・水増し後の画像を CNN に入力し，学習を行う。CNN の各層の層数やパラメータを変更することが可能となっている。詳細については，4 節に記載する。

### (4) ファインチューニング

追加で学習を行いたい画像を入力し，学習済みの CNN モデルの重みを更新する際に利用する。詳細については，5 節に記載する。

### (5) 推論

推論したい画像を入力すると，その画像のクラスを予測する。詳細については，6 節に記載する。

## 2. 動作環境・プログラムの実行方法

### 2.1 動作環境

ライブラリ等のバージョン

- Python Ver.3.6
- tensorflow Ver.2.3.0
- keras Ver.2.4.3
- scikit-learn Ver.0.24.2
- numpy Ver.1.17.0
- opencv Ver.4.0.1
- matplotlib Ver.3.3.4
- pillow Ver.8.3.1

※Python Ver.3.8 , numpy Ver.1.20.3 でも動作確認済み

### 2.2プログラムの実行方法

- ① 以下の六つのファイルを同じ作業ディレクトリに入れてください.

run\_select.py, Image\_Aug\_choice.py, Image\_Augmentation.py, cnn\_learning.py  
fine\_tuning.py, predict.py

- ② run\_select.py を実行する.

- ③ 実行すると、以下のような選択画面が表示される.

機械学習汎用システムの実行項目を選択(1~4)(終了: exit):

- 1を入力した場合  
前処理・水増しが実行される。(詳細は、3 節に記載)
- 2を入力した場合  
CNN が実行される。(詳細は、4 節に記載)
- 3を入力した場合  
ファインチューニングが実行される。(詳細は、5 節に記載)
- 4を入力した場合  
推論が実行される。(詳細は、6 節に記載)
- exitを入力した場合  
機械学習汎用システムを終了する.

※ 例) 1→前処理・水増し実行→2→CNN 実行→exit→終了

上記のようにすると、前処理・水増しで出力したデータセットをそのまま CNN へ入力することが可能である.

### 3. 前処理・水増しプログラムについて

`run_select.py` の実行後, 1 を選択すると前処理・水増しを実行できる

#### 3.1 前処理・水増しプログラムの動作

① 入力画像のパスを入力してください(終了: `exit`):

入力画像が入っているフォルダのパスを入力してください. (`exit` で入力を終了する)

※分類したいクラスごとにパスを入力

入力例) `C:\Users\wnl\jupyter\machine_learning\sample_project\input\deer`  
`C:\Users\wnl\jupyter\machine_learning\sample_project\input\boar`  
`C:\Users\wnl\jupyter\machine_learning\sample_project\input\crow`  
`exit`

上記の例のように入力を行うと, 3 クラス (`deer`, `boar`, `crow`) に分類できる.

② 画像サイズ変更方法選択(1~6):

画像サイズを変更する際の方法を選択する.

選択できる項目は以下の 6 個である.

1 : パディング (`padding`)

2 : リサイズ (`resize`)

3 : Center Crop (`center_crop`)

4 : Random Crop (`random_crop`)

5 : Scale Augmentation (`scale_augmentation`)

6 : Aspect Ratio Augmentation (`aspect_ratio_augmentation`)

[入力例]

入力例 a) 3

`Center Crop` が実行され, 下記パラメータを入力する. (1~3 選択の場合)

変更サイズ:

入力例) 128

入力例 b) 4

`Random Crop` が実行され, 下記パラメータを入力する. (4 選択の場合)

変更サイズ:

入力例) 224

大きめにリサイズする際の短辺の長さ:

入力例) 256

何枚ずつ生成しますか:

入力例) 5

入力例 c) 5

**Scale Augmentation** が実行され、下記パラメータを入力する。(5~6 選択の場合)

変更サイズ:

入力例) 224

大きめにリサイズする際の短辺の長さの上限:

入力例) 280

大きめにリサイズする際の短辺の長さの下限:

入力例) 230

何枚ずつ生成しますか:

入力例) 5

③ 学習データの割合(%):

学習データとテストデータの割合を決める.

入力例) 70

上記のように入力すると入力画像が学習データ 70%テストデータ 30%に分けられる.

④ 前処理・水増し方法の選択(7~31)(終了: exit):

前処理・水増し方法を選択し、それぞれのパラメータを入力する.

選択できる項目は下記の通りである.

7: はみ出して回転 (rotation\_1)

8: はみ出さずに回転 (rotation\_2)

9: 上下反転 (vertical\_flip)

10: 左右反転 (horizontal\_flip)

11: 拡大のみ (zoom\_in\_only)

12: 拡大縮小 (zoom\_in\_out)

13: 垂直移動 (vertical\_shift)

14: 水平移動 (horizontal\_shift)

15: 二値化 (ノーマル) (threshold\_normal)

16: 大津の二値化 (threshold\_otsu)

- 17 : Adaptive threshold(mean) (adaptive\_threshold\_mean)
- 18 : Adaptive threshold(gaussian) (adaptive\_threshold\_gaussian)
- 19 : 平均値フィルタ (blur)
- 20 : ガウシアンフィルタ (gaussian\_blur)
- 21 : 中央値フィルタ (median\_blur)
- 22 : バイラテラルフィルタ (bilateral\_filter)
- 23 : ガウシアンノイズ (gaussian\_noise)
- 24 : インパルスノイズ (impulse\_noise)
- 25 : 積和演算による輝度・コントラスト調整 (brightness\_contrast)
- 26 : ガンマ補正による輝度調整 (gamma\_correction)
- 27 : PCA Color Augmentation (pca\_color\_augmentation)
- 28 : 画像の色相をシフト (channel\_shift)
- 29 : アフィン変換 (affine)
- 30 : 射影変換 (syaei)
- 31 : シアー変換 (shear)

入力例) 7

```

    回転角度: 10          ←10° ずつ回転する
    10                    ←左右反転は、パラメータなし
    19
    ぼかしの強度(フィルタサイズ)(奇数):  ←入力なしデフォルト値 5 で動作
    exit
  
```

上記のように入力すると、はみ出して回転と左右反転と平均値フィルタが実行される。

※ 前処理・水増しの各処理のパラメータは、デフォルト値が設定されており、  
入力なしで **Enter** を押すとデフォルト値で動作する。

⑤ 別の前処理・水増しの組み合わせでも画像を生成しますか(y/n):

入力例 a) y

別の前処理・水増しの組み合わせにより追加で画像を生成する。(④に戻る)

入力例 b) n

前処理・水増しを終了する。(⑥に進む)

⑥ 水増し後の画像が何枚以上の場合間引くか:

水増し後の画像が増えすぎた場合に、何枚以上で間引くかを入力する.

入力例) 20000

水増し後の画像が 20000 枚以上の場合間引く.

※入力をせずに Enter を押すとデフォルト値 10000 で動作する.

(水増し後の画像が 10000 枚以上の場合間引く.)

⑦ png でデータセットを出力するパス:

水増し後の出力画像を png (可視化できる状態) で保存する.

入力例) C:\Users\wnl\jupyter\machine\_learning\sample\_project\output1

上記のように入力すると, output1 フォルダの中に train (学習画像) と test (テスト画像) フォルダが自動生成され, train と test それぞれクラスごとにフォルダで分類し保存する.

下記に出力例を示す.

出力例)

C:\Users\wnl\jupyter\machine\_learning\sample\_project\output1\train\class0\_deer\0.png  
C:\Users\wnl\jupyter\machine\_learning\sample\_project\output1\train\class1\_boar\0.png  
C:\Users\wnl\jupyter\machine\_learning\sample\_project\output1\train\class2\_crow\0.png  
C:\Users\wnl\jupyter\machine\_learning\sample\_project\output1\test\class0\_deer\0.png  
C:\Users\wnl\jupyter\machine\_learning\sample\_project\output1\test\class1\_boar\0.png  
C:\Users\wnl\jupyter\machine\_learning\sample\_project\output1\test\class2\_crow\0.png

⑧ 全体にかける前処理選択(32~38)(終了: exit):

全体にかける前処理を選択する.

選択できる項目は下記の通りである.

32 : PCA 白色化 (PCA\_Whitening)

33 : ZCA 白色化 (ZCA\_Whitening)

34 : 標準化 (standardization)

35 : 正規化 最大値最小値(normalization\_max\_min)

36 : 正規化 255 で割る(normalization\_255)

37 : Mean Subtraction (mean\_subtraction)

38 : Per-pixel Mean Subtraction (per\_pixel\_mean\_subtraction)

入力例 a) 33

ZCA 白色化が実行される.

入力例 b) exit

exit を入力すると全体にかかる前処理は行わない.

⑨ 学習画像のデータセットを出力するパス(.npz まで入力してください):

学習画像のデータセットを npz ファイルで保存する.

入力例) C:\Users\wnl\jupyter\machine\_learning\sample\_project\output1\train.npz

⑩ テスト画像のデータセットを出力するパス(.npz まで入力してください):

テスト画像のデータセットを npz ファイルで保存する.

入力例) C:\Users\wnl\jupyter\machine\_learning\sample\_project\output1\test.npz

⑪ クラス名を保存するパス(.txt まで入力してください):

入力した画像のクラス名をテキスト(txt)で保存する.

入力例) C:\Users\wnl\jupyter\machine\_learning\sample\_project\output1\class.txt

※ クラス名のテキストファイルは, 推論時に必要となる.

## 3.2 前処理・水増しの関数一覧

番号	関数名	引数(デフォルト値)・・・引数の説明
1	padding	input_path・・・入力画像のパス,resize_size・・・変更サイズ
2	resize	input_path・・・入力画像のパス,resize_size・・・変更サイズ
3	center_crop	input_path・・・入力画像のパス,resize_size・・・変更サイズ
4	random_crop	input_path・・・入力画像のパス,crop_size・・・変更サイズ, resize_size・・・大きめにリサイズする際の短辺の長さ,count・・・生成枚数
5	scale_augmentation	input_path・・・入力画像のパス,crop_size・・・変更サイズ, resize_size_upperlimit・・・大きめにリサイズする際の短辺の長さの上限, resize_size_lowerlimit・・・大きめにリサイズする際の短辺の長さの下限 count・・・生成枚数
6	aspect_ratio_augmentation	input_path・・・入力画像のパス,crop_size・・・変更サイズ, resize_size_upperlimit・・・大きめにリサイズする際の短辺の長さの上限, resize_size_lowerlimit・・・大きめにリサイズする際の短辺の長さの下限, count・・・生成枚数
7	rotation_1	input_x,input_y,angle(60)・・・回転角度

8	rotation_2	input_x,input_y,angle(60)・・・回転角度
9	vertical_flip	input_x,input_y
10	horizontal_flip	input_x,input_y
11	zoom_in_only	input_x,input_y,zoom_range_lowerlimit(100)・・・拡大倍率下限 , zoom_range_upperlimit(150)・・・拡大倍率上限 ,count(3)・・・生成枚数
12	zoom_in_out	input_x,input_y,zoom_range_lowerlimit(50)・・・拡大縮小倍率下限 , zoom_range_upperlimit(150)・・・拡大縮小倍率上限 ,count(5)・・・生成枚数
13	vertical_shift	input_x,input_y,zure(20)・・・何ピクセルずつずらすか , up(40)・・・上方向どこまでずらすか ,down(40)・・・下方向どこまでずらすか
14	horizontal_shift	input_x,input_y,zure(20)・・・何ピクセルずつずらすか , left(40)・・・左方向どこまでずらすか ,right(40)・・・右方向どこまでずらすか
15	threshold_normal	input_x,input_y,threshold_range(127)・・・二値化の閾値
16	threshold_otsu	input_x,input_y
17	adaptive_threshold_mean	input_x,input_y,block_size(9)・・・近傍ピクセルサイズ(奇数) , c_value(0)・・・平均または加重平均から引く値
18	adaptive_threshold_gaussian	input_x,input_y,block_size(9)・・・近傍ピクセルサイズ(奇数) , c_value(0)・・・平均または加重平均から引く値
19	blur	input_x,input_y,blur_strength(5)・・・ぼかしの強度(フィルタサイズ)(奇数)
20	gaussian_blur	input_x,input_y,blur_strength(5)・・・ぼかしの強度(フィルタサイズ)(奇数)
21	median_blur	input_x,input_y,blur_strength(5)・・・ぼかしの強度(フィルタサイズ)(奇数)
22	bilateral_filter	input_x,input_y,diameter(9)・・・周辺の何ピクセルを見るか(9 程度) , sigma_color(75)・・・色の分散値(75~100 程度) , sigma_space(75)・・・距離の分散値(75~100 程度)
23	gaussian_noise	input_x,input_y,sigma(15)・・・ノイズの量(15 程度)
24	impulse_noise	input_x,input_y,amount(0.5)・・・ノイズの量(%)
25	brightness_contrast	input_x,input_y,alpha_lowerlimit(0.6)・・・コントラストの調整下限(0~3 程度) , alpha_upperlimit(1.8)・・・コントラストの調整上限(0~3 程度) , beta_lowerlimit(50)・・・明るさの調整下限(-100~100 程度) , beta_upperlimit(50)・・・明るさの調整上限(-100~100 程度) , count(5)・・・生成枚数
26	gamma_correction	input_x,input_y,gamma_lowerlimit(0.4)・・・ガンマ下限 , gamma_upperlimit(2.4)・・・ガンマ上限 ,count(3)・・・生成枚数
27	pca_color_augmentation	input_x,input_y,count(5)・・・生成枚数
28	channel_shift	input_x,input_y,shift_range(360)・・・色相をランダムでシフトする上限, count(5)・・・生成枚数



29	affine	input_x,input_y,x_range(0.25)・・・元の x 座標からどれだけ動かすか, y_range(0.25)・・・元の y 座標からどれだけ動かすか
30	syaei	input_x,input_y,syaei_range_left(0.2)・・・台形の上底左側, syaei_range_right(0.8)・・・台形の上底右側
31	shear	input_x,input_y,sh_range(40)・・・せん断する際の角度の範囲, count(3)・・・生成枚数
32	PCA_Whitening	input_x,input_y
33	ZCA_Whitening	input_x,input_y
34	standardization	input_x,input_y
35	normalization_max_min	input_x,input_y
36	normalization_255	input_x,input_y
37	mean_subtraction	input_x,input_y
38	per_pixel_mean_subtraction	input_x,input_y

※ input\_x は前処理・水増しの各処理に入力される画像, input\_y は, input\_x の中に入っている各画像に対しての分類クラスである. input\_x と input\_y はいずれもユーザ側の入力が必要ではない.

## 4. CNN プログラムについて

run\_select.py で 2 を選択すると CNN が実行される. 前処理・水増しプログラムでの出力画像を CNN に入力し, 学習する. この際, CNN の各層の層数とパラメータを変更することができる.

### 4.1 CNN プログラムの動作

※ run\_select.py で前処理・水増しと CNN を連続して行った場合, ①, ②は問われず, ③の入力から始まる.

#### ① 学習画像のデータセットを入力(npz ファイル):

学習画像のデータセット (npz ファイル) のパスを入力してください.

入力例) C:\Users\¥wnl¥.jupyter¥machine\_learning¥sample\_project¥output1¥train.npz

基本的には, 前処理・水増しプログラムで出力したデータセット

#### ② テスト画像のデータセットを入力(npz ファイル):

テスト画像のデータセット (npz ファイル) のパスを入力してください.

入力例) C:\Users\¥wnl¥.jupyter¥machine\_learning¥sample\_project¥output1¥test.npz

基本的には、前処理・水増しプログラムで出力したデータセットを指定する.

③ Conv 層と Pooling 層と Dropout 層を入力(CNN 前半)

畳み込み層とプーリング層とドロップアウト層の層数を入力する.

入力方法は、1) 畳み込み層の層数入力, 2) 半角スペース空ける,  
3) プーリング層の層数入力, 4) 半角スペース空ける, 5) ドロップアウト層の層数入力.  
この手順を繰り返す. ※ 入力の末尾は、半角スペースを空けないでください.

入力例 a) 2 1 0 2 1 0 3 1 0 3 1 0 3 1 0

VGG16 を作成する場合の入力例である. 0 になっているのは、ドロップアウト層の部分で使用する場合は、1 を入力する.

入力例 b) 2 1 0 1 1 1

Conv 層 2 つ→Pooling 層 1 つ→Dropout 層なし→Conv 層 1 つ→Pooling 層 1 つ  
→Dropout 層 1 つの CNN を作成する場合の入力例である.

④ 畳み込みのカーネルサイズを入力:

[入力例]

入力例) 3

※入力せずに Enter を押すとデフォルト値 3 で動作する.

⑤ 畳み込みのストライドを入力:

[入力例]

入力例) 1

※入力無しで Enter を押すとデフォルト値 1 で動作する.

⑥ 畳み込みの特徴マップ数の決め方(1:畳み込みとプーリングが繰り返されるごとに 2 をかける, 2:畳み込みとプーリングが繰り返されるごとに特徴マップ数を入力):

入力例 a) 2

2 を選択した場合、下記のように畳み込みとプーリングが繰り返されるごとに特徴マップ数を入力する.

入力例) 1 ユニット目の畳み込みの特徴マップ数を入力: 64

- 2 ユニット目の畳み込みの特徴マップ数を入力: 128
- 3 ユニット目の畳み込みの特徴マップ数を入力: 256
- 4 ユニット目の畳み込みの特徴マップ数を入力: 512
- 5 ユニット目の畳み込みの特徴マップ数を入力: 512

入力例 b) 1

1 を選択した場合、畳み込みとプーリングが繰り返されるごとに 2 をかける。

入力例) 最初の畳み込みの特徴マップ数を入力: 64

上記のように入力すると、1 ユニット目:64, 2 ユニット目:128, 3 ユニット目:256

4 ユニット目:512, 5 ユニット目:1024 と設定される。

※ 入力無しで Enter を押すとデフォルト値 32 で動作し、  
最初の畳み込みの特徴マップ数が 32 になる。

⑦ 活性化関数を選択(1~6):

活性化関数を選択する。

選択できる項目は下記の通りである。

1 : ReLU

2 : Sigmoid

3 : tanh

4 : LeakyReLU

5 : elu

6 : selu

入力例 a) 1

1 を選択すると活性化関数は、ReLU になる。(VGG16 は、ReLU を用いる)

入力例 b)

入力無しで Enter を押すとデフォルト値 1 で動作し、活性化関数は ReLU になる。

⑧ プーリング手法を選択(1,2):

入力例 a) 1

1 を選択すると最大値プーリングになる。

※入力せずに、Enter を押すとデフォルト値 1 で動作し、最大値プーリングになる。

入力例 b) 2

2 を選択すると平均値プーリングになる。

⑨ プーリングサイズを入力:

入力例) 2

※入力無しで **Enter** を押すとデフォルト値 2 で動作する.

⑩ ドロップアウト率を入力(ex : 0.25):

※ドロップアウト層を 1 層も挿入しなかった場合, ドロップアウト率は, 問われず,  
⑪の入力へ進む.

入力例) 0.2

※入力せずに, **Enter** を押すとデフォルト値 0.25 で動作する.

⑪ 全結合層とドロップアウト層を入力(CNN 後半)

全結合層とドロップアウト層の層数を入力する. 入力方法としては, 1) 全結合層の層数入力, 2) 半角スペース空ける, 3) ドロップアウト層の層数を入力する. この手順を繰り返し入力する. ※ 入力の末尾には, 半角スペースを空けない.

入力例 a) 2 0

全結合層は 2, ドロップアウト層は 1 の場合である.

入力例 b) 1 1 1 1

全結合層 1 つ→ドロップアウト層 1 つ→全結合層 1 つ→ドロップアウト層 1 つの場合の入力である.

⑫ 全結合層のノード数の決め方(1:全結合の層数が増えるごとに 2 をかける, 2:全ての全結合層に対してノード数を入力):

入力例 a) 2

2 を選択した場合, 下記のように全ての全結合層に対してノード数を入力する.

入力例) 1 層目の全結合層のノード数を入力: 4096

2 層目の全結合層のノード数を入力: 4096

入力例 b) 1

1 を選択した場合, 全結合の層数が増えるごとに 2 をかける.

入力例) 最初の全結合層のノード数を入力(ex : 512): 512

上記のように入力すると, 1 層目:512, 2 層目:1024 と設定される.

※ 入力無しで **Enter** を押すとデフォルト値 512 で動作し,

最初の全結合層のノード数が 512 になる.

⑬ ドロップアウト率を入力(ex : 0.5):

※ドロップアウト層を 1 層も挿入しなかった場合, ドロップアウト率は, 問われず,  
⑭の入力へ進む.

入力例) 0.4

※入力無しで Enter を押すとデフォルト値 0.5 で動作する.

⑭ 学習回数を入力:

入力例) 100

※入力無しで Enter を押すとデフォルト値 50 で動作する.

⑮ バッチサイズを入力:

入力例) 16

※入力無しで Enter を押すとデフォルト値 32 で動作する.

⑯ オプティマイザを選択(1~6):

オプティマイザを選択する.

選択できる項目は下記の通りである.

1 : Adam

2 : Nadam

3 : RMSProp

4 : Adamax

5 : Adagrad

6 : SGD

入力例) 1

※入力無しで Enter を押すとデフォルト値 1 で動作し, Adam になる.

⑰ Early Stopping を使用しますか(y/n):

入力例) y

過学習等で loss が悪化しないように学習回数を満たしていなくても途中で学習を終了させる. (y を選択時)

- ⑬ 学習後のモデルを保存するパスを入力(.h5 まで入力してください):  
入力例) C:\Users\wnl\jupyter\machine\_learning\sample\_project\output1\model.h5

## 5. ファインチューニングのプログラムについて

run\_select.py で3を選択するとファインチューニングが実行される.

※ run\_select.py で追加したい画像の前処理・水増しとファインチューニングを連続して行った場合, ①,②は問われず③の入力から始まる.

- ① 追加で学習させる学習画像のデータセットを入力(npz ファイル):

入力例) C:\Users\wnl\jupyter\machine\_learning\sample\_project\output2\train\_fine.npz

※ 追加で学習させる画像は, 学習済みモデルを作成した時と同様の前処理・水増しを行わなければならない部分があるため, 以下のように前処理・水増しプログラムを用いて作成してください.

- 3.1-①入力画像のフォルダのパスをクラスの順番が同じになるよう入力.
- 3.1-②画像サイズ変更方法と変更サイズを同じにする.
- 3.1-⑧全体にかける前処理を同じにする.

各出力パスが以前のものと被らないように注意してください.

- ② 追加で学習させるテスト画像のデータセットを入力(npz ファイル):

入力例) C:\Users\wnl\jupyter\machine\_learning\sample\_project\output2\test\_fine.npz

- ③ 学習済みモデル(.h5)のパスを入力:

追加で学習させたいモデルを入力する.

入力例) C:\Users\wnl\jupyter\machine\_learning\sample\_project\output1\model.h5

- ④ 学習回数を入力:

入力例) 100

※入力無しでEnterを押すとデフォルト値 50 で動作する.

- ⑤ バッチサイズを入力:

入力例) 16

※入力無しでEnterを押すとデフォルト値 64 で動作する.

- ⑥ オプティマイザを選択(1~6):

オプティマイザを選択する.

選択できる項目は以下の通りである.

- 1 : Adam
- 2 : Nadam
- 3 : RMSProp
- 4 : Adamax
- 5 : Adagrad
- 6 : SGD

入力例) 1

※入力無しで Enter を押すとデフォルト値 1 で動作し, Adam になる.

⑦ Early Stopping を使用しますか(y/n):

入力例) y

過学習等で loss が悪化しないように学習回数を満たしていなくとも, 途中で学習を終了させる. (y を選択時)

⑧ 学習後のモデルを保存するパスを入力(.h5 まで入力してください):

入力例) C:\Users\wnl\jupyter\machine\_learning\sample\_project\output2\model\_fine.h5

## 6. 推論のプログラムについて

run\_select.py で 4 を選択すると推論が実行される. run\_select.py で CNN と推論する画像の前処理・水増しと推論を連続して行った場合, ①,③は問われずに推論を実行可能である. また, 上記に加え, CNN で学習する前に学習画像の前処理・水増しも行っている場合, ②は問われずに推論を実行可能である.

① 学習済みモデル(.h5)のパスを入力:

推論を行うために使用する学習済みモデルを入力してください.

入力例) C:\Users\wnl\jupyter\machine\_learning\sample\_project\output1\model.h5

② クラス名を保存したパスを入力(.txt まで入力してください):

①の学習済みモデルを作成した際のクラス名が保存されているテキストファイルのパスを入力してください.

入力例) C:\Users\wnl\jupyter\machine\_learning\sample\_project\output1\class.txt

③ 推論する画像のパスを入力(.npz まで入力してください):

入力例) C:\Users\wnl\jupyter\machine\_learning\sample\_project\output3\predict.npz

※ 推論する画像は、学習済みモデルを作成した時と同様の前処理・水増しを行わなければならない部分があるため、以下のように前処理・水増しプログラムを用いて作成してください。

- 3.1-②画像サイズ変更方法と変更サイズを同じにする.
- 3.1-③学習データの割合を 0 %にする.  
(推論する画像は、水増ししてはいけないためテスト画像で出力)
- 3.1-⑧全体にかける前処理を同じにする.
- 3.1-⑩テスト画像のデータセット出力パスの最後を `test` ではなく `predict` にする.  
各出力パスが以前のものと被らないように注意してください.