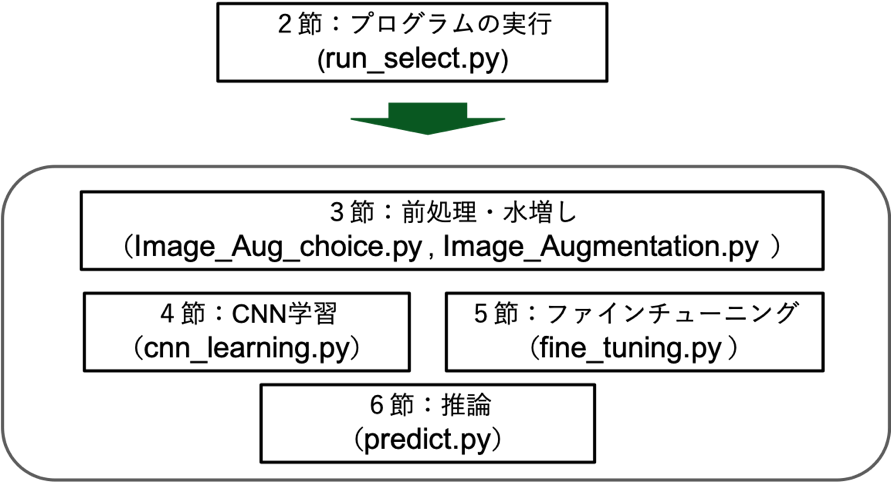
機械学習汎用システム仕様書

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．

1. 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．

1. 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．

1. 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．

1. 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．

1. 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

* 1. 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

1. run\_select.pyを実行する．
2. 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．

1. 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．

1. 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

1. 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．

1. 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.

1. 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）

1. 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．）

1. 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

1. 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ける前処理は行わない．

1. 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

1. 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

1. 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は必要ではない．

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

1. 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．

1. 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．

1. 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．

1. 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．

1. 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．

1. 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．

1. 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．

1. 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．

1. 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．

1. 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．

1. 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．

1. 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．

1. 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．

1. 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．

1. 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．

1. 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）

1. 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-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-画像サイズ変更方法と変更サイズを同じにする．
* 3.1-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．

1. 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

1. 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

1. 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．

1. 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．

1. 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．

1. 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）

1. 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

1. 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

1. 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-画像サイズ変更方法と変更サイズを同じにする．
* 3.1-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-全体にかける前処理を同じにする．
* 3.1-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．