

画像データのデジタル化

広島大学 AI・データイノベーション教育研究センター
村上 祐子

目標

画像を特定のルールで分類するための前処理ができる

この授業で紹介すること

- 画像データのデジタル化
- アノテーション

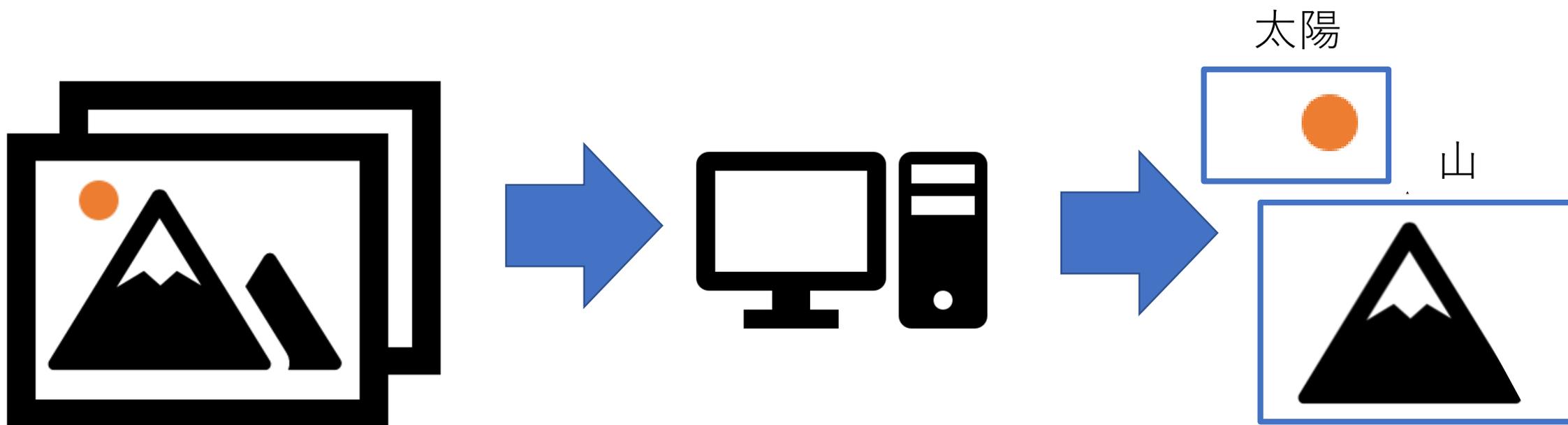
キーワード

画像データ、デジタル化、アノテーション

この講義ではGoogle Colab(Colaboratory)を使ったプログラミングを実施します。

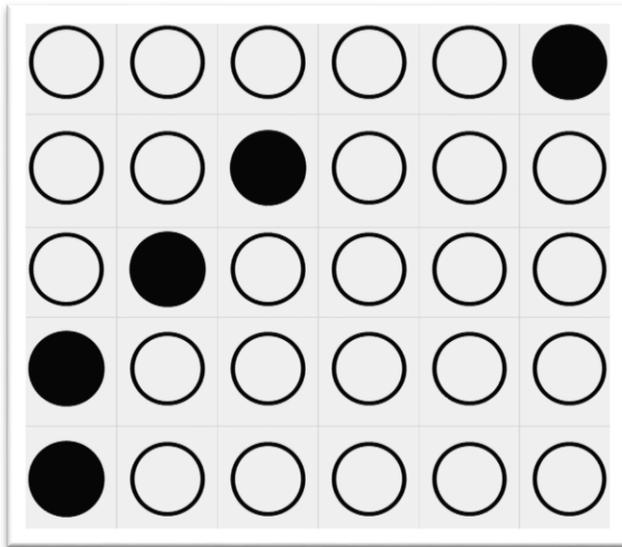
こんなことはありませんか？

写真やカメラ映像から個人を認識したり、生物の種類を識別できるシステムがあります。画像はどのようにコンピュータで認識されているのでしょうか。

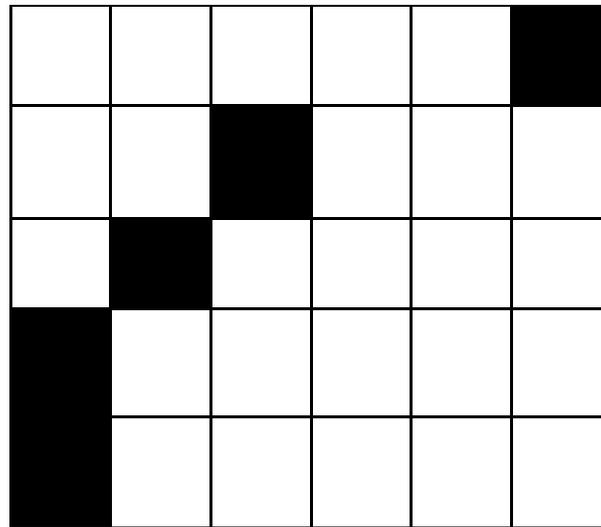


画像データのデジタル化

元データ (写真)



標本化



量子化

A 5x6 grid of numerical values representing quantized image data. The values are either 1 or -1, corresponding to the original image data. The pattern is as follows:

1	1	1	1	1	-1
1	1	-1	1	1	1
1	-1	1	1	1	1
-1	1	1	1	1	1
-1	1	1	1	1	1

例題

6×6のリバーシの盤面を作ってみましょう。

「白が勝った」盤面のプログラム例を次のページで掲載します。

これを元に黒が勝った盤面を作りましょう。

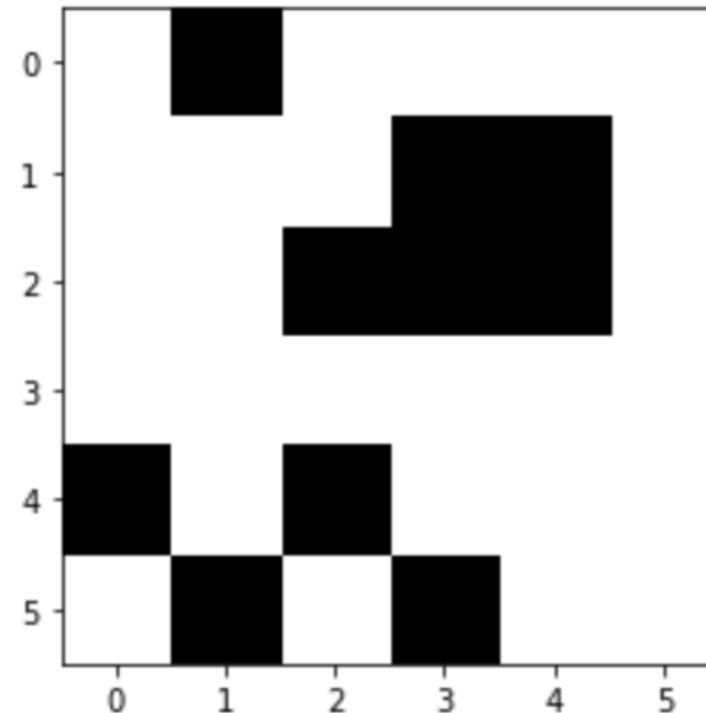
例題

```
import numpy as np
import random
N = 36
kuro = 10
ransu = random.sample(range(1,N+1), k=N)
banmen = np.ones((6,6))

for i in range(36):
    n1 = int(ransu[i]/6)
    if n1==6:
        n1=0
    n2 = ransu[i] % 6
    if i < kuro:
        banmen[n1][n2] = -1
```

```
import matplotlib.pyplot as plt
plt.imshow(banmen, cmap='gray')
plt.show()
```

これを実行すると
図示できる

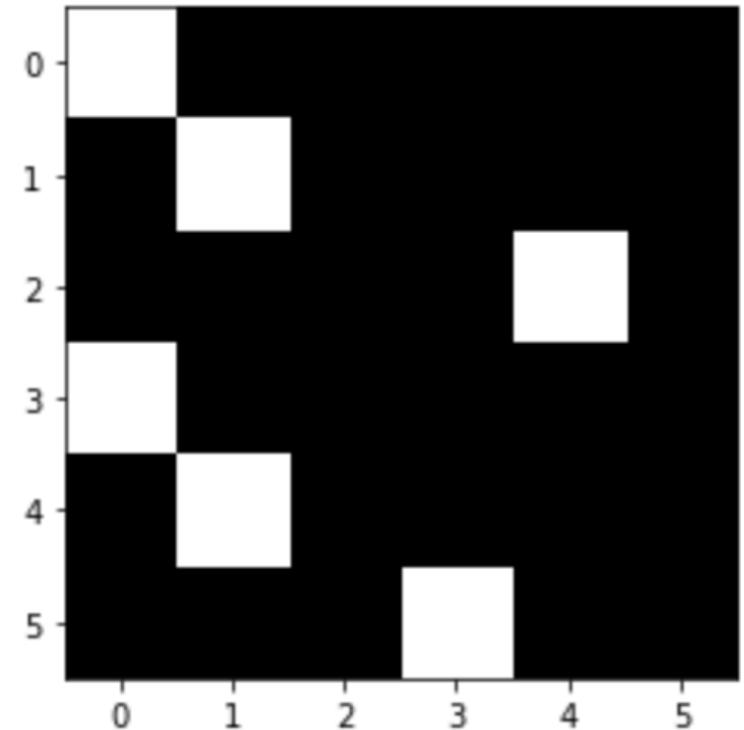


解説

黒が勝つには、36個のコマのうち半分の18より多くが黒でないといけない。

```
import numpy as np
import random
N = 36
kuro = 30
ransu = random.sample(range(1,N+1), k=N)
banmen = np.ones((6,6))
```

18より大きい数値を
入れる



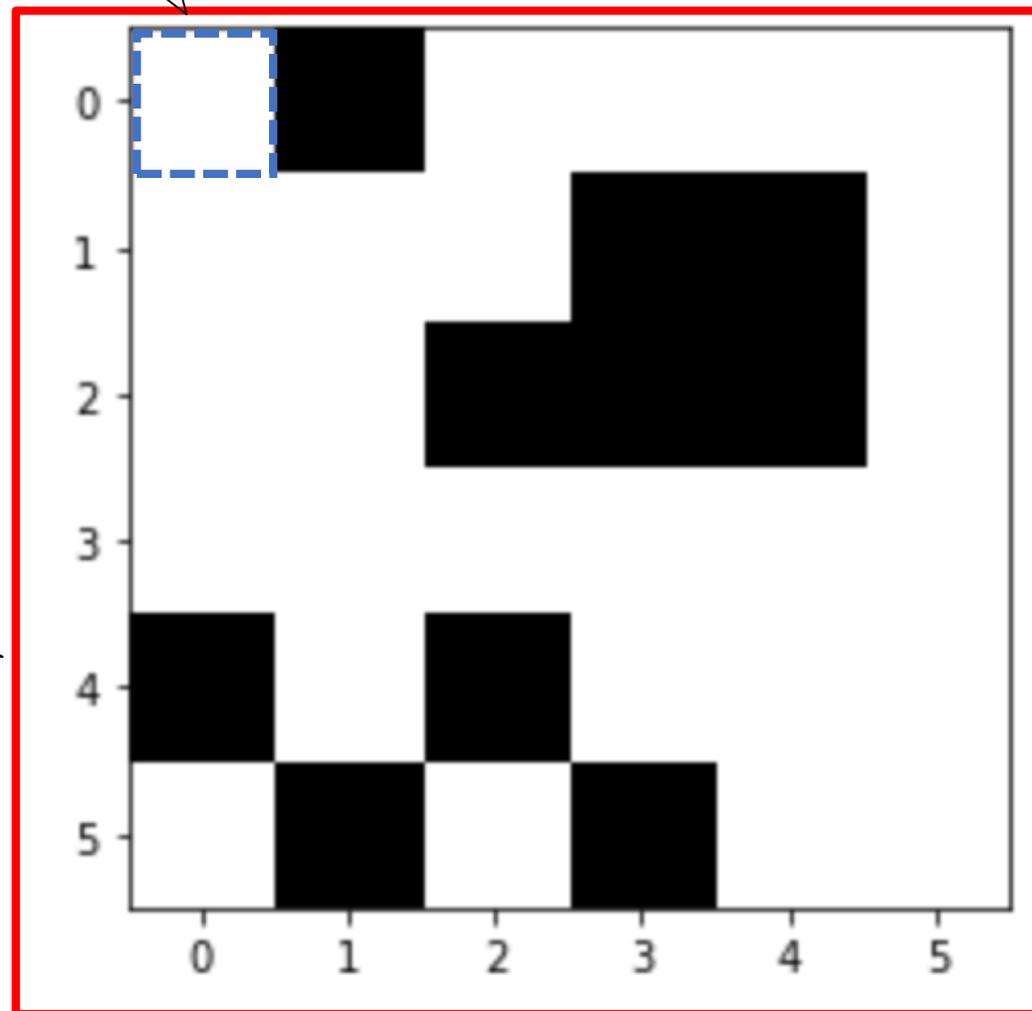
アノテーション

- 例題で作ったデータは、ある位置が白(1)か黒(-1)かの情報のみ
- 画像からリバーシの勝ち負けを分類させるには勝ち負けの情報をデータに付与しなければならない

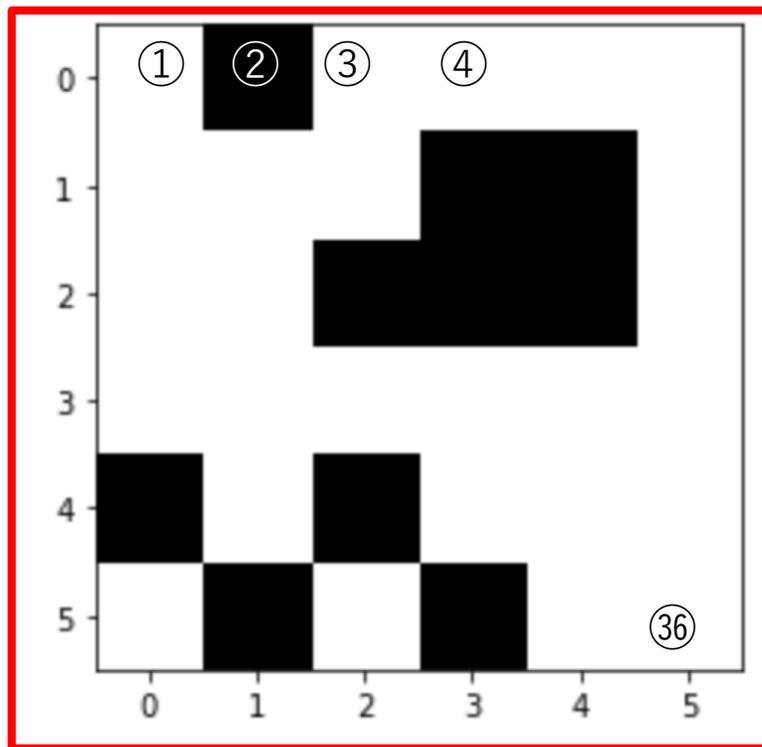
どちらが勝ったのか？

この作業をアノテーションという

白



アノテーション



	0	1	2	3	4	5
0	1	-1	1	1	1	1
1	1	1	1	-1	-1	1
2	1	1	-1	-1	-1	1
3	1	1	1	1	1	1
4	-1	1	-1	1	1	1
5	1	-1	1	-1	1	1

- 図と同様に2次元（縦横）で盤面を表現している
- このままでは複数回の試合結果を表現できない



Index	1番目のマス	2番目のマス	...	36番目のマス	勝敗
0	白	黒	...	白	白
1	白	白	...	黒	黒
2	黒	黒	...	白	白

盤面の状況を1次元(横1列)で表現するように変更

問題

6×6のリバーシで

- (1) 黒が勝つ
- (2) 白が勝つ

場合の盤面の画像をそれぞれ3個ずつデータを保存するプログラムを次のページで紹介いたします。これを元に、2つの場合の盤面の画像をそれぞれ30個ずつ、すなわち、全部で60個分のデータを作成してください。

問題

```
import numpy as np
import random
import pandas as pd
N = 36
```

次のページに続きます。

```
def make_table_reversi(kuro):
    ransu = random.sample(range(1,N+1), k=N)
    banmen = np.ones((6,6))
    for i in range(36):
        n1 = int(ransu[i]/6)
        if n1==6:
            n1=0
        n2 = ransu[i] % 6
        if i < kuro:
            banmen[n1][n2] = -1
    tmp = []
    for j in range (6):
        for i in range (6):
            tmp.append(banmen[i][j])
    df = pd.DataFrame([tmp])
    return df
```

問題

```
df_siro = pd.DataFrame(index=[], columns=[])
df_kuro = pd.DataFrame(index=[], columns=[])

for i in range(3):
    df_siro = pd.concat([df_siro,make_table_reversi(10)],axis=0)
    df_kuro = pd.concat([df_kuro,make_table_reversi(26)],axis=0)

df_siro = df_siro.assign(winner=1)
df_kuro = df_kuro.assign(winner=-1)

df_all = pd.concat([df_kuro,df_siro], axis=0)
```