



# 画像音声認識-1,2

- 画像処理とは、画像入出力、色変換

4/10,13/2023

上條浩一



- 画像処理、音声処理に関して、理論を理解していただきつつ、実装やアプリを通じて、体感していただきます
- 画像処理では、主にOpenCV (python上で)を使います
- 並行して行われている“人工知能システム開発II”も意識しながら進めます

- **出席80%未満は単位を落とします**
- 以下の提出物があります (( )内の数字は最終評価での割合)
  - おさらい小テスト (25): 前の週のおさらい(授業開始10~15分)
  - 演習(授業内) (25): 授業中LMSに提出する演習
  - 宿題 (25): 提出期限(4,5日程度)までLMSに提出する宿題
  - 最終レポート (25): 最終課題(発表+資料提出)
  - **宿題提出80%未満は単位を落とします**
  - **最終レポートは、未提出だとそれだけで単位を落とします**
- 期末テストはありません
  - 小テスト、演習、宿題は、最終的には、上記点数に配分されるように点数を normalization(標準化)します

# 学生要覧との対応

- p22
- 11. 単位認定

## 2. 単位の認定

科目担当教員が期末に評価を行い、以下の条件をすべて満たした場合、その授業科目について定められた単位数が与えられます。

- ① 成績評価で60点（評価C）以上の評価を得たとき
- ② 出席率 80%以上
- ③ 課題認定率 80%以上

※成績評価は「12. 成績評価」を、課題認定は「14. 課題認定のルール」を確認してください。

- p27
- 14. 課題認定

## 3. 課題の種類

授業中に指定される課題は、下記の2つに分かれます。

**小課題**：期限までに未提出または未認定の場合、課題認定率に影響します。  
 ≪例≫授業ごとの宿題、授業内課題

**必須課題**：試験に代わる課題であり、期限までに未提出または未認定の場合、単位認定に影響します。  
 ≪例≫中間課題、期末課題、期末レポート

それぞれ指定の提出先へ期限までに提出してください。IPUT LMSやメールでの提出の場合は、提出後に課題が確実に送信されているかを確認してください。

小課題＝宿題に対応  
 授業内の演習も、評価の点数に反映されます

必修課題＝  
 最終レポートに対応



# スケジュール

(途中で変更の可能性あり)

赤：変更の可能性あり  
(上條が海外学会参加)  
→8/1 or 2に補講の可能性

Aクラス	Bクラス	回数	内容	おさらい小テスト (授業開始時)	演習(授業内 LMS提出)	宿題(基本授業後 4,5日後迄に提出)	人工システム開発II
4月13日	4月10日	1,2	画像 - 画像処理とは、画像入出力、色変換			○	(三宅先生課題)
4月20日	4月17日	3,4	画像 - フィルタ処理	○	○	○	(三宅先生課題)
4月27日	4月24日	5,6	画像 - 幾何学変換	○	○	○	(三宅先生課題)
5月11日	5月08日	7,8	画像 - 2値化画像処理	○	○	○	自然言語処理関連
5月18日	5月15日	9,10	画像 - 特徴量抽出	○	○	○	自然言語処理関連
5月25日	5月22日	11,12	画像 - 動画画像処理、カメラモデル	○	○	○	自然言語処理関連
6月01日	5月29日	13,14	画像 - サポートベクターマシン	○	○	○	(三宅先生課題)
6月08日	6月05日	15,16	画像 - CNN	○	○	○	(三宅先生課題)
6月15日	6月12日	17,18	画像 - JPEG	○	○	○	(三宅先生課題)
6月22日	6月19日	19,20	音声 - 音、音声、音声認識とは	○	○	○	画像処理関連
6月29日	6月26日	21,22	音声 - 音声認識の特徴抽出	○	○	○	画像処理関連
7月06日	7月03日	23,24	音声 - 音声識別	○	○	○	画像処理関連
7月13日	7月10日	25,26	音声 - 最新言語処理	○	○	○	(最終課題)
7月20日	7月24日	27,28	今までの総復習	○	○		(最終課題)
7月27日	7月31日	29,30	最終課題発表会				(最終課題)

演習・宿題の実施は予定が変わる場合があります

# 出欠

- 授業開始から5分以内にLMSに記入 → 出席
- 授業開始から5分以上15分以内にLMSに記入 → 遅刻
- 授業開始から15分以上 → 欠席
- 遅刻3回で、欠席1回扱い



# 注意-1！！

- **出席80%以上、宿題80%以上のぎりぎりを狙うのはやめてください!!**
- それを狙っているかどうかは、出席や宿題の提出状況からわかり、その場合は、印象は非常に悪くなります
- また、その場合、もし何かあって出席できなくなった場合、単位を落とすことになります
- 授業内の演習には、LMSに提出するものとしのないものがありますが、**提出しない演習も、皆さんがちゃんとやっているかどうか確認しているの**  
**で+そもそもやらないと理解が進まない**ので、未提出の演習もしっかり**実施**してください!!

# 注意-2！！

- 本講義の資料、本講義で扱うコード、画像等は**絶対に**外部SNS (twitter, discord等) にuploadしないでください
- 宿題、最終レポート等においては、外部サイトのコピペは厳禁です
  - 数行や図・表を少し参考にするのは良いですが、その場合は必ず**出典元を記載**してください
- 授業内の演習や小テストの答えを別のクラス (BからA) に教えるのは絶対にやめてください
  - もし教えた場合、教えられた方も、教えた方も、全てのテストを0点にします
  - A, Bで問題は異なるので、別のクラスの問題の答えをコピーすると発覚します



# slackに関するお願い

1. 以下のslackに自分が登録されていることを確認し、されていない場合は、至急教えてください
  - 1. Aクラス：2\_23\_3\_画像-音声認識-a
  - 2. Bクラス：2\_23\_3\_画像-音声認識-b
2. この授業に関する情報は、上記slackに流しますので、1日最低1回はcheckするようにしてください

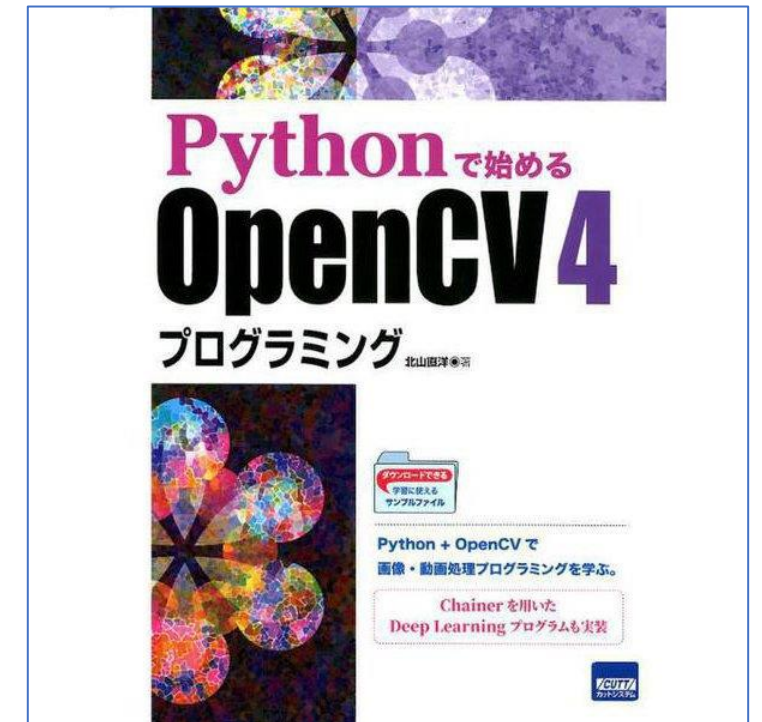
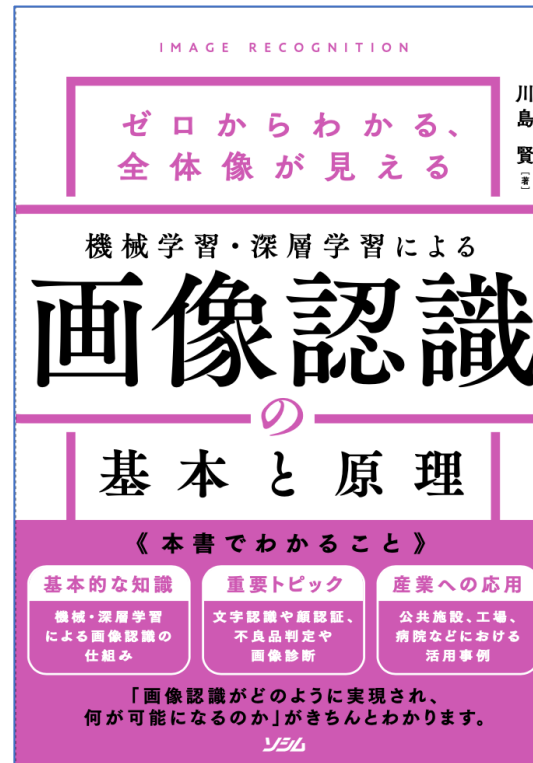
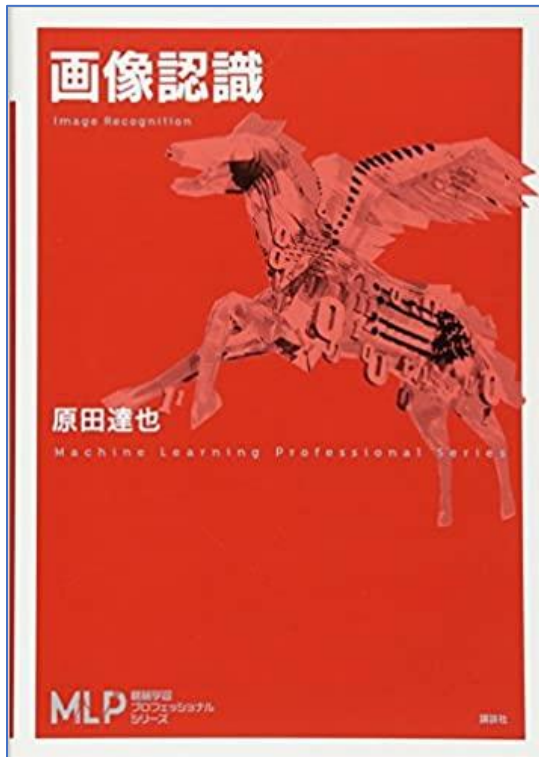
# 教科書(画像)

- 画像処理 (未来へつなぐ デジタル  
シリーズ 28) (共立出版)  
大町 真一郎, 陳 謙, 大町 方子, 宮田 高  
道, 他著



# 参考書(画像)

- 画像認識 (講談社)  
原田 達也著
- 原理が細かく記載されているが、数式が多く少し難解
- 画像認識の基本と原理  
(ソシム) 川島 賢著
- 画像認識の原理が機械学習ベースにわかりやすく書かれている
- Pythonで始めるOpenCV4  
プログラミング(カットシステム)北山直洋著
- 画像処理の具体例がpython codeでわかりやすく書かれている



# 教科書(音声)

- イラストで学ぶ音声認識 (講談社)  
荒木雅弘 著



# 参考書(音声)

- 音声認識 (講談社)  
篠田浩一 著
- クラシックな音声認識から深層学習まで  
幅広くカバー





# 参考書(自然言語処理)

- 音声認識と自然言語処理は密接な関連があり、本授業の最後(7月)に自然言語処理の授業を予定しています

## BERTによる自然言語処理入門

オーム社、2,970円

BERTの原理から、実際のコードまで細かく解説



## BERT/GPT-3/DALL-E 自然言語処理・画像処理 音声処理 人工知能プログラミング実践入門

株式会社ボーンデジタル、3,960円

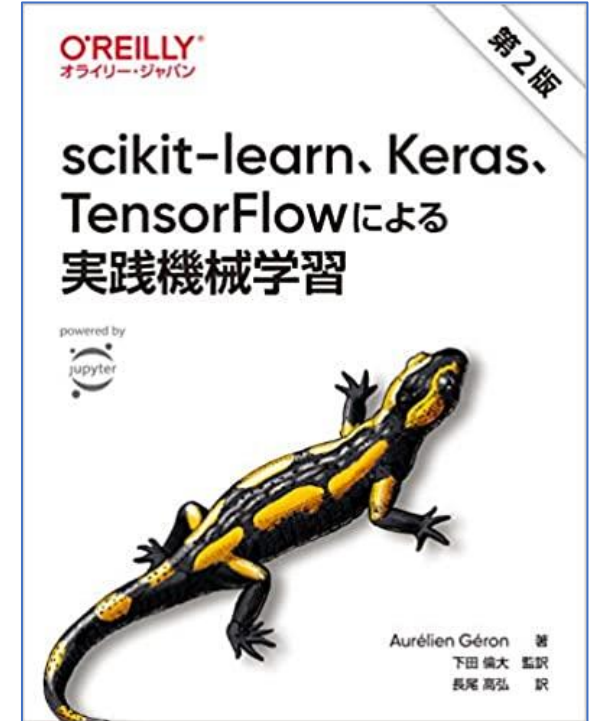
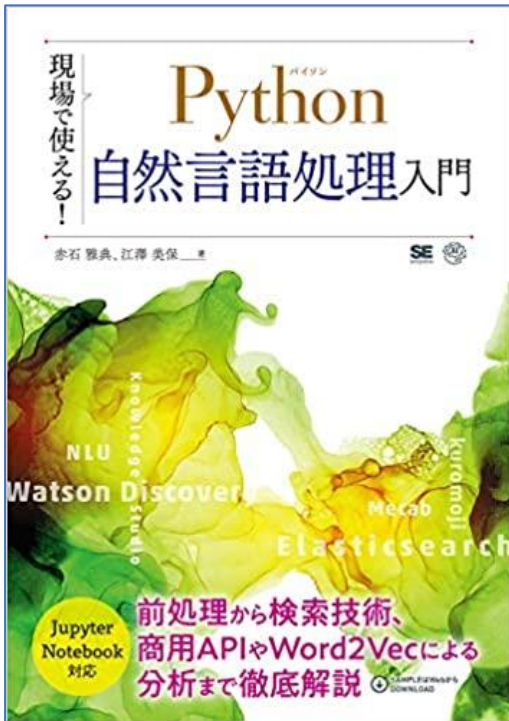
最新のAIの理論を、コードも含めて解説





# 参考書(自然言語処理 続き)

- 現場で使える!  
Python自然言語処理入門 (AI& TECHNOLOGY)  
赤石 雅典、江澤 美保 著
- 自然言語処理の基礎から実装まで解り易く記述されている
- つくりながら学ぶ!  
PyTorchによる発展ディープラーニング(マイナビ)  
小川雄太郎著
- 7,8章に、Transformer, BERTの最新紹介がある
- scikit-learn、Keras、TensorFlowによる実践機械学習(O'Reilly)Aurelien Geron (著)
- 15,16章に、RNN,LSTMの技術と実装の詳細がある



# 進め方（画像処理）

- 講義を行い、基本、OpenCV (python)で実装して、画像処理を体感していただきます
  - OpenCVは、基本、Google Colabで行います
  - ただし、5月に動画を扱う際は、PC localで処理を行います
    - ✓Google Colabだと、local PCのカメラが使いにくいいため
- lena.bmp等、授業で使う画像等は、授業前に予め自分のPCにdownloadしておいてください（コクーンの通信速度が遅いことが多いので）
- また、授業にはできる限り通信ケーブルを持ってきてください

- 去年の鈴木先生と上條の“メディア処理実習”の講義等で少々使ったかと思いますが、Google Colab上でOpenCVを頻繁に使うので、Google Colabは使えるようにしておいてください。→今日の宿題の1つ
- Google Colab、OpenCVに関しては、2年後期の“メディア実習”の2日目の資料の”メディア情報処理実習第2回\_aft.pdf”のp18~30を必ず復習しておいてください

- Editorを1つ用意してください
- 既に使っているeditorがあれば何でもOKです
- 例：
  - Visual Studio
  - Sublime Text
  - ATOM



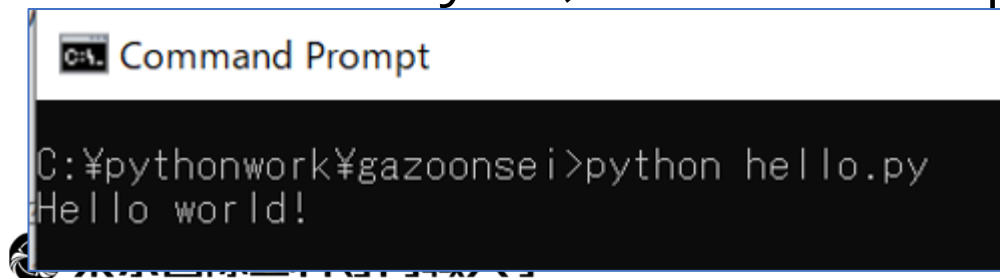
# 確認3 – Local PC上でのpython

- もし、openCVが自分のPCにinstallされていない場合（メディア処理実習でinstallしているはずですが）、必ずinstallして、下記が正しく動くことを確認しておいて下さい。Install方法は、“メディア実習”の2日目の資料“メディア情報処理実習第2回\_aft.pdf”のp67,68を参照ください
- 正しくinstallされていることの確認方法➡今日の宿題の1つ

1. 適当なdirectoryに、以下の内容のhello.pyというファイルを作成

```
1 import cv2
2 print("Hello world!")
```

2. そのdirectoryで、command promptで、python hello.pyを実行



# Local PCで、module not foundとなった場合

- 基本、pip install (module名) でinstallできます

例えば、以下の場合

```
C:\pythonwork\gazoonsei>python jpegenc.py
Traceback (most recent call last):
  File "C:\pythonwork\gazoonsei\jpegenc.py", line 4, in <module>
    import matplotlib.pyplot as plt
ModuleNotFoundError: No module named 'matplotlib'
```

このようにする

```
C:\pythonwork\gazoonsei>pip install matplotlib
Collecting matplotlib
  Downloading matplotlib-3.5.1-cp310-cp310-win_amd64.whl (7.2 MB)
----- 7.2/7.2 MB 16.4 MB/s eta 0:00:00
```

(以下略)

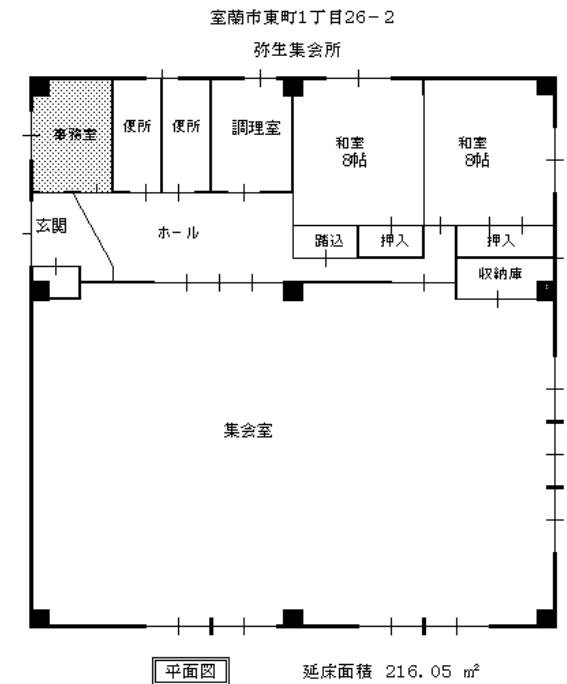
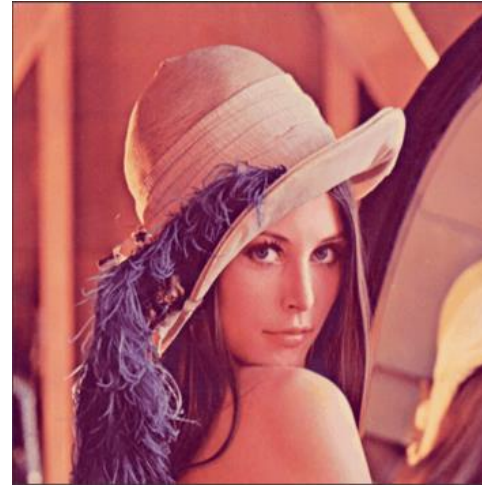


# 本日のゴール

- 画像とは
  - 写真、絵、図面等を電子化したもの
- 解像度と量子化
  - 解像度：画像の細かさ
  - 量子化：色、輝度の階調
- 画像処理と画像認識
  - 画像処理：画像を扱う信号処理全般
  - 画像認識：画像に写る内容を理解すること
- 画像の入出力
  - 入力装置
  - 出力装置
- 光とは？
  - 電磁波の1種
  - 量子化：色、輝度の階調
- 加法混色と減法混色
- HSV色空間

# 画像とは

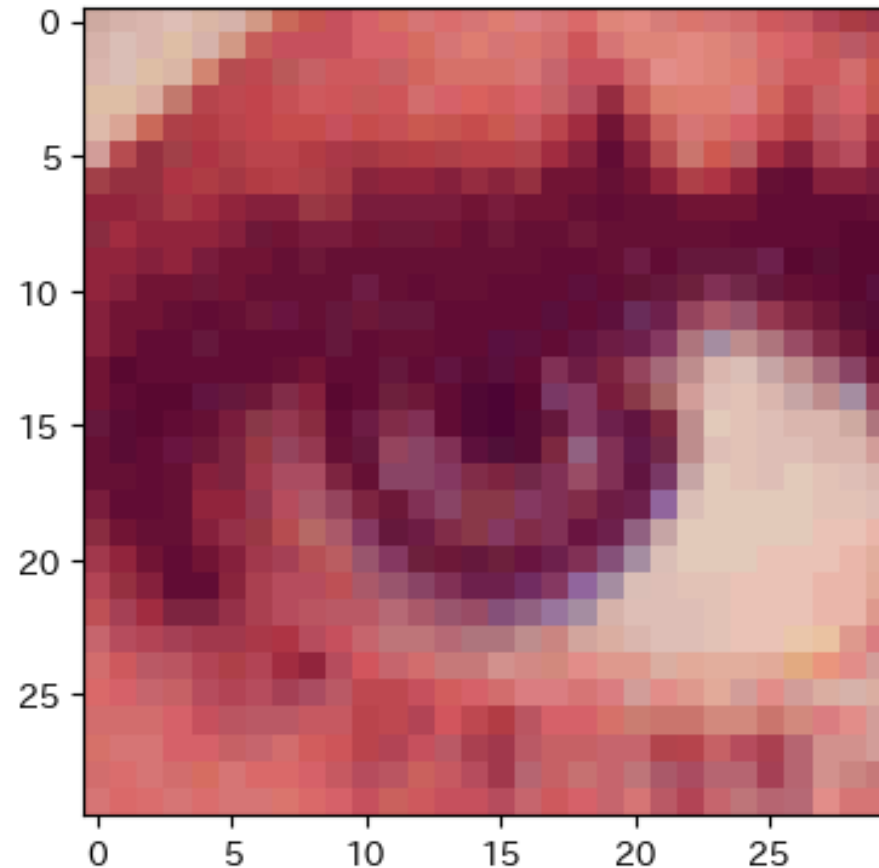
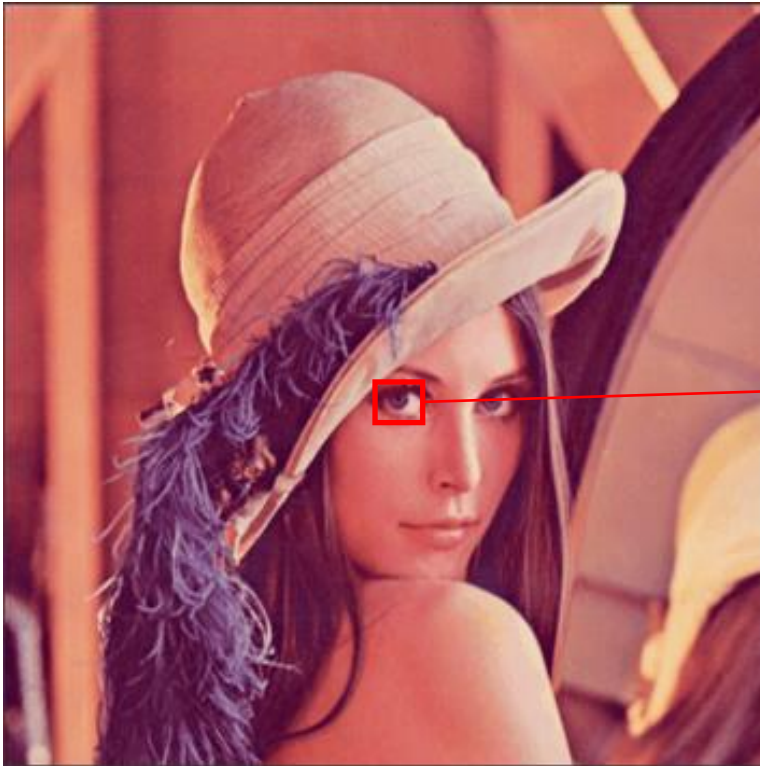
- 写真、絵、図面等を電子化したもの
  - 元々は、画いた絵、程度の意味だった
  - 現在は、“画像”はデジタル画像の総称
  - テキスト（文字）ではない



# 画像を拡大すると。。。。

- 画像は点の集合

- 1つ1つの点を“画素”または“ピクセル(pixel)”と呼ぶ
- (昔の銀塩カメラで撮った) 写真には、画素という概念はない

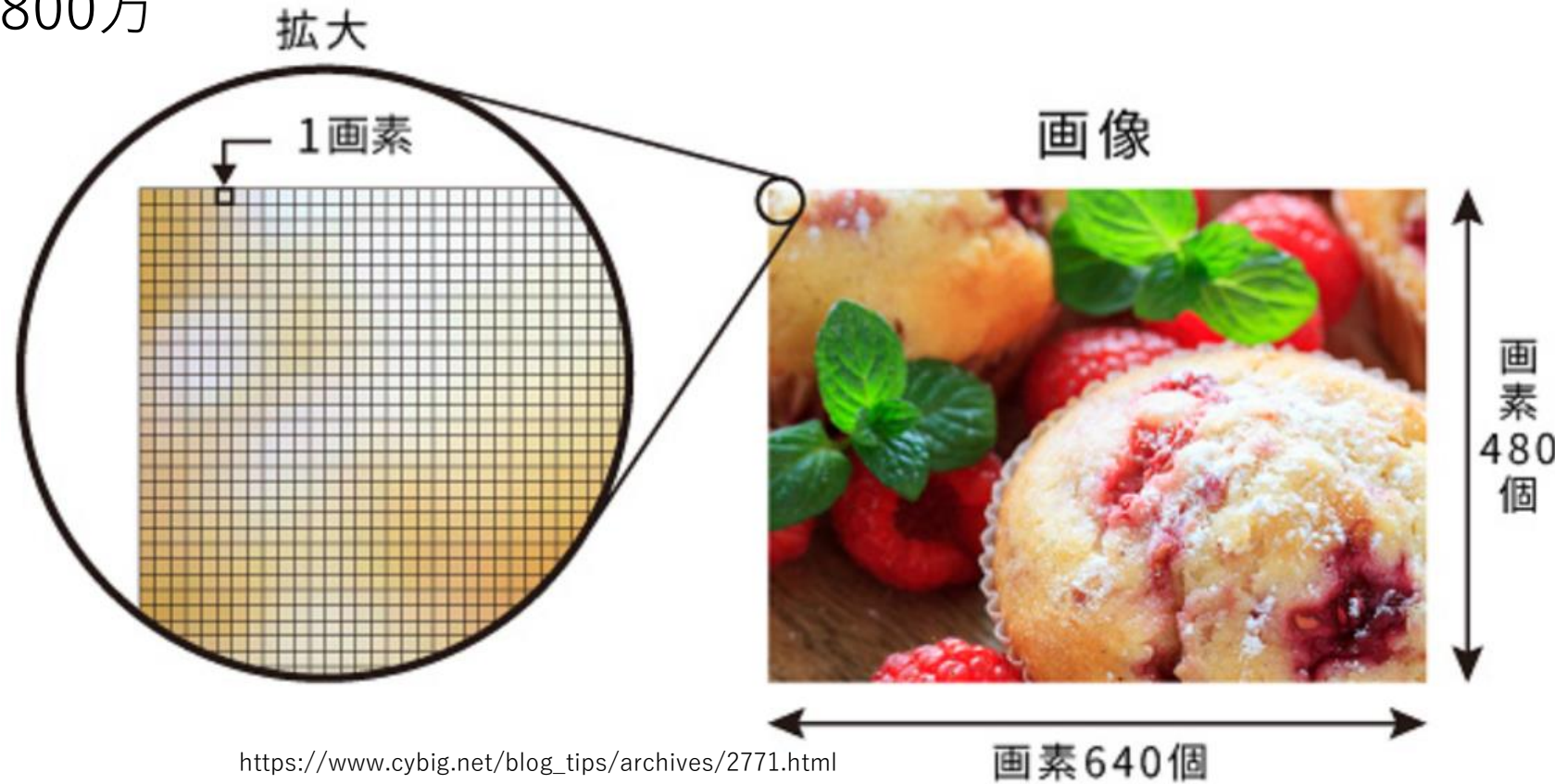


<http://lumilia.blog73.fc2.com/blog-entry-565.html>

[https://www.city.muroran.lg.jp/main/org7500/heimen\\_shuukaijo.html](https://www.city.muroran.lg.jp/main/org7500/heimen_shuukaijo.html)

# 画像サイズ

- 画素（ピクセル）が横、縦にいくつ並んでいるか
  - 右下の画像は、 $640 \times 480 = 307,200$  画素
  - 4K TV =  $3840 \times 2160 = 8,294,400$
  - iphone カメラ = 200万~4800万





# カラー画像と白黒画像



## カラー：

各画素は、Red, Green, Blueの値を持つ

8bit colorの場合、各色0~255( $2^8$ )値

R,G,Bで $8 \times 3 = 24$ bit : 24bitカラー

各画素が $256 \times 256 \times 256 = 16,777,216$ の色を表現

## 白黒：

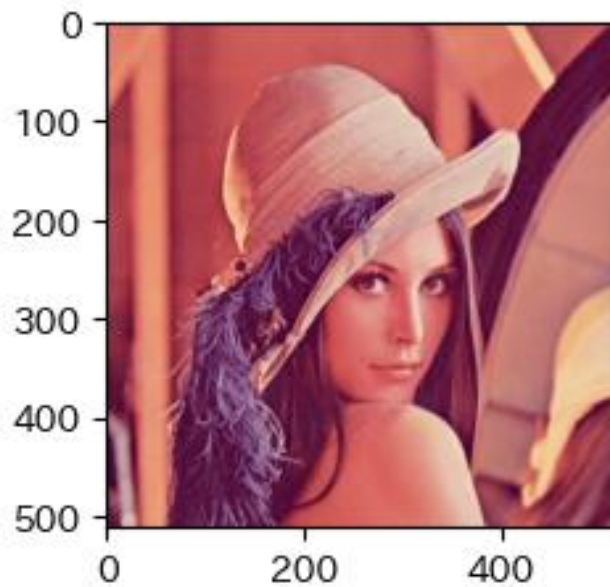
各画素は、輝度のみの値を持つ  
8bitの場合、0~255の値

- 解像度：
  - 画像の細かさ
  - 単位面積当たりの画素数
- 量子化：
  - 色や輝度の階調
  - 1画素を表すビット数(ビットの深さ)

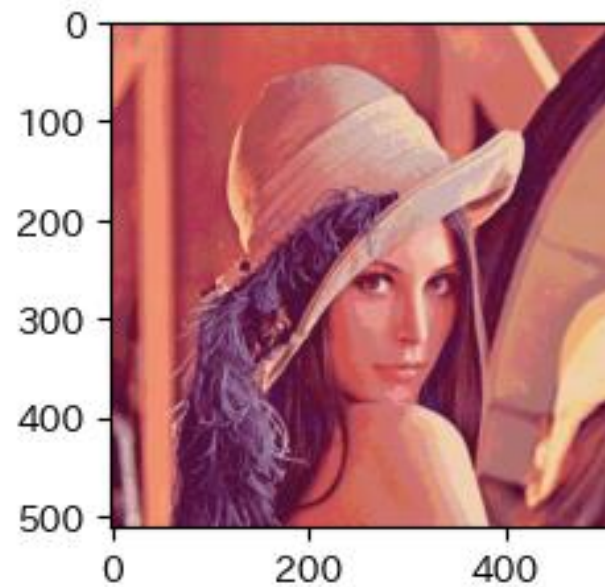


# 量子化

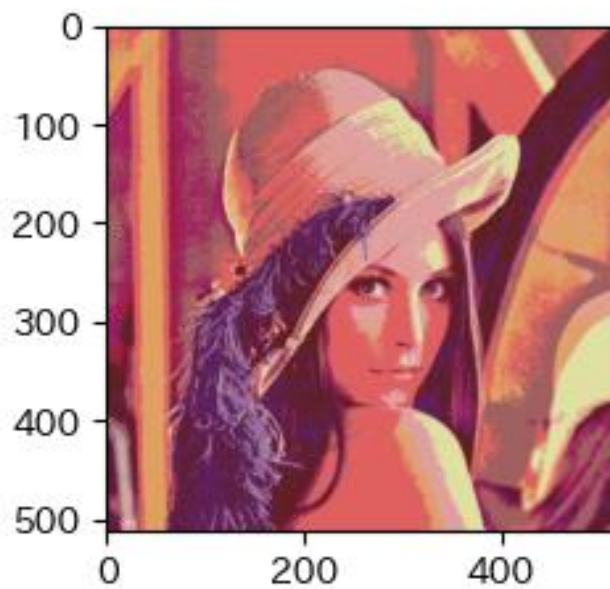
256値(8 bit)



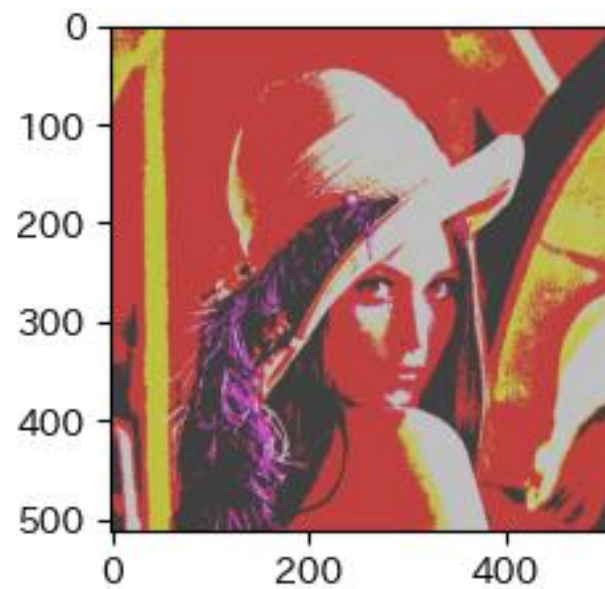
8値(3bit)



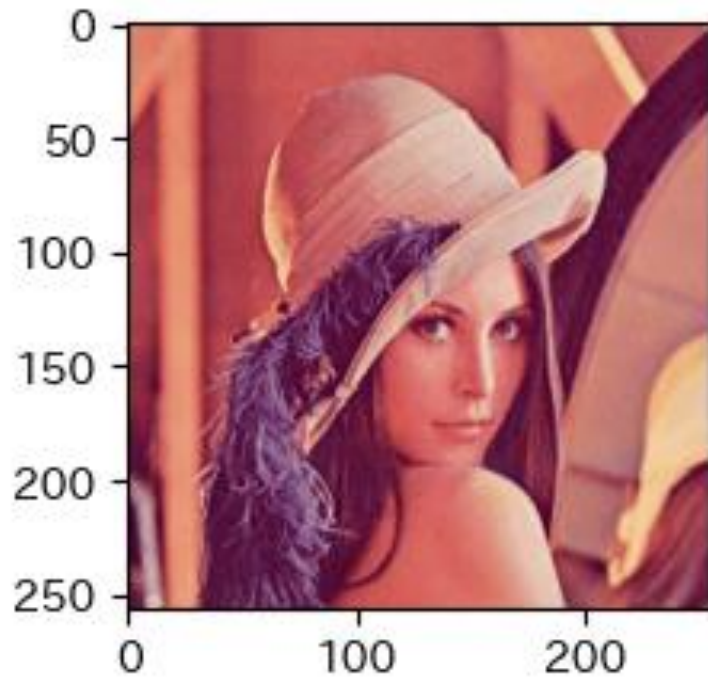
4値(2bit)



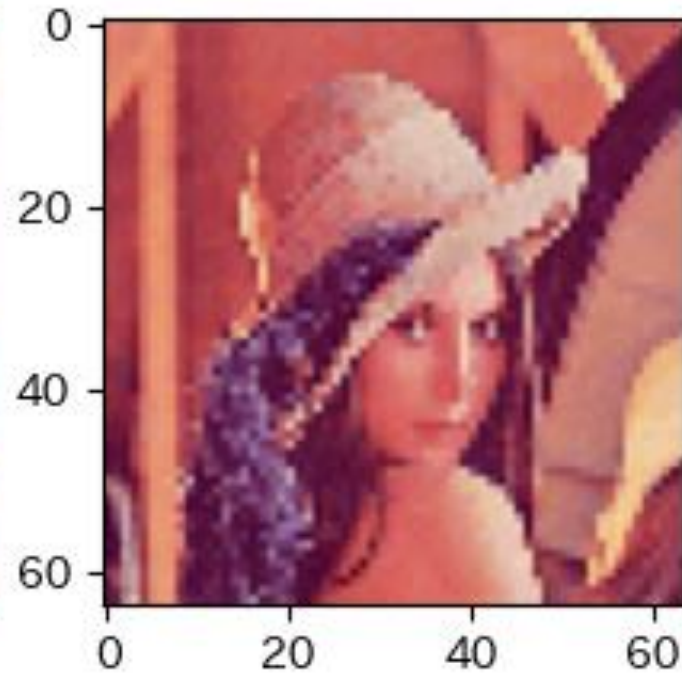
2値(1bit)



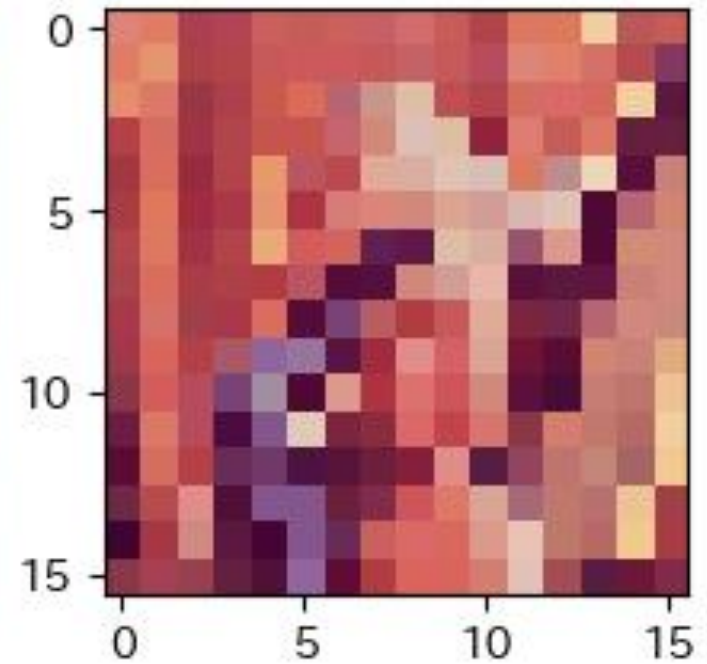
# 解像度



256x256



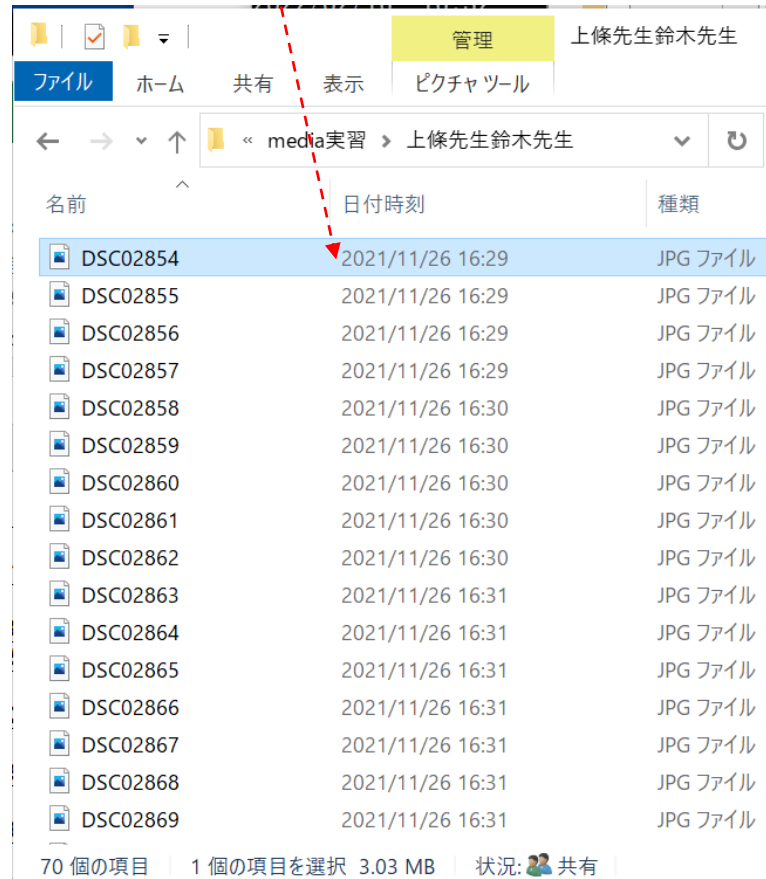
64x64



16x16

# 画素数、量子化等の見分け方 - windows

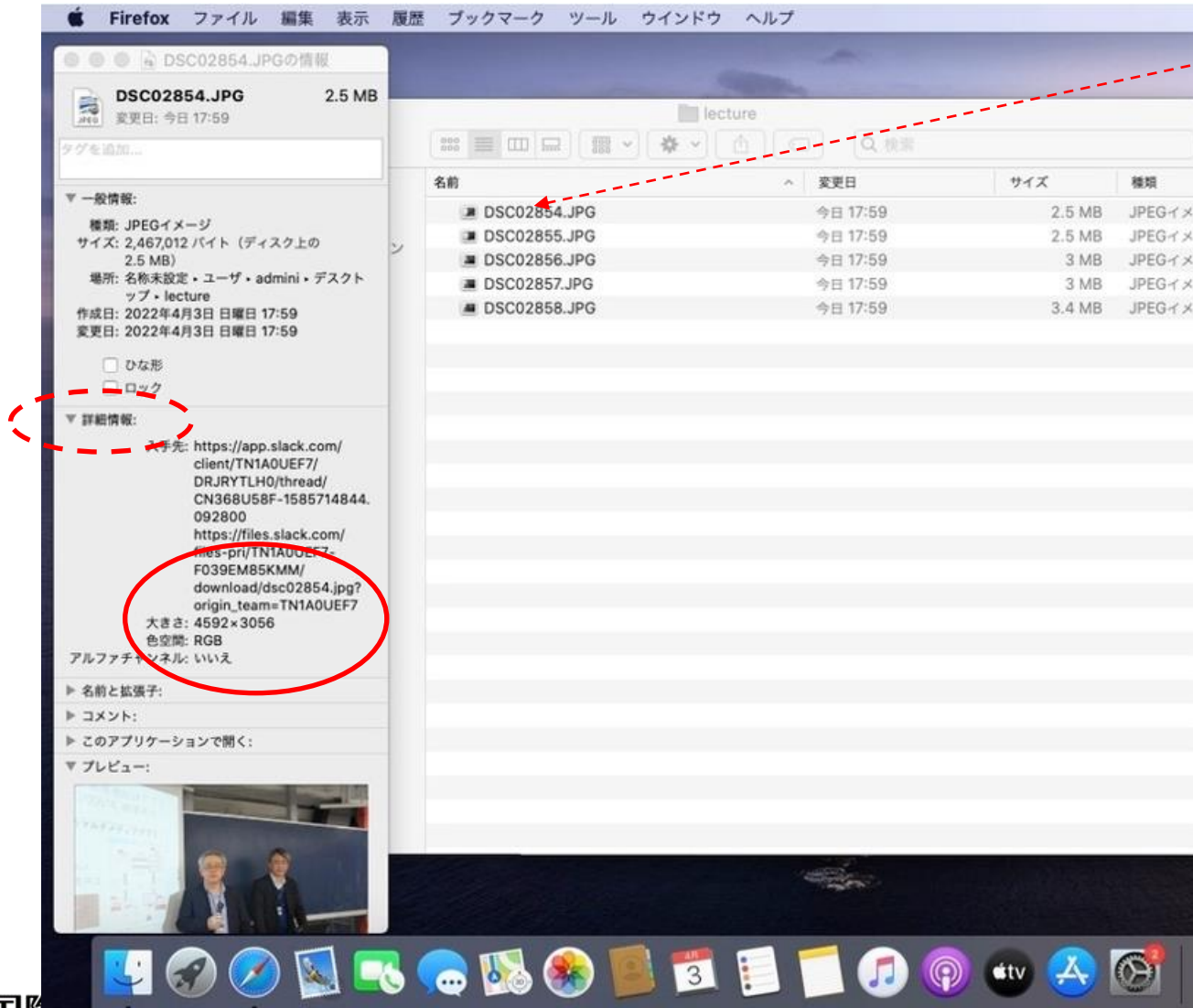
ファイルを右クリック



• 各自、自分のPCで確認してください

• PC上に適当な画像がない場合、LMSにupしてある画像をdownloadして確認してください

# 画素数、量子化の見分け方 - mac



ファイルを右クリック

・各自、自分のPCで確認してください



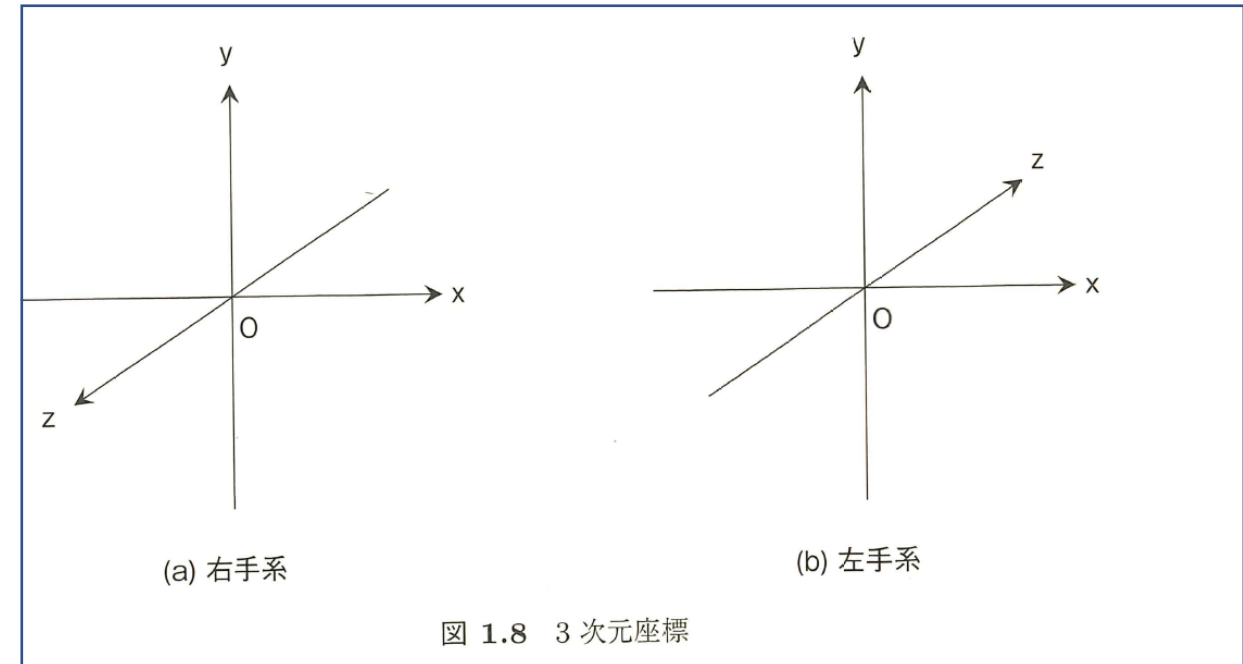
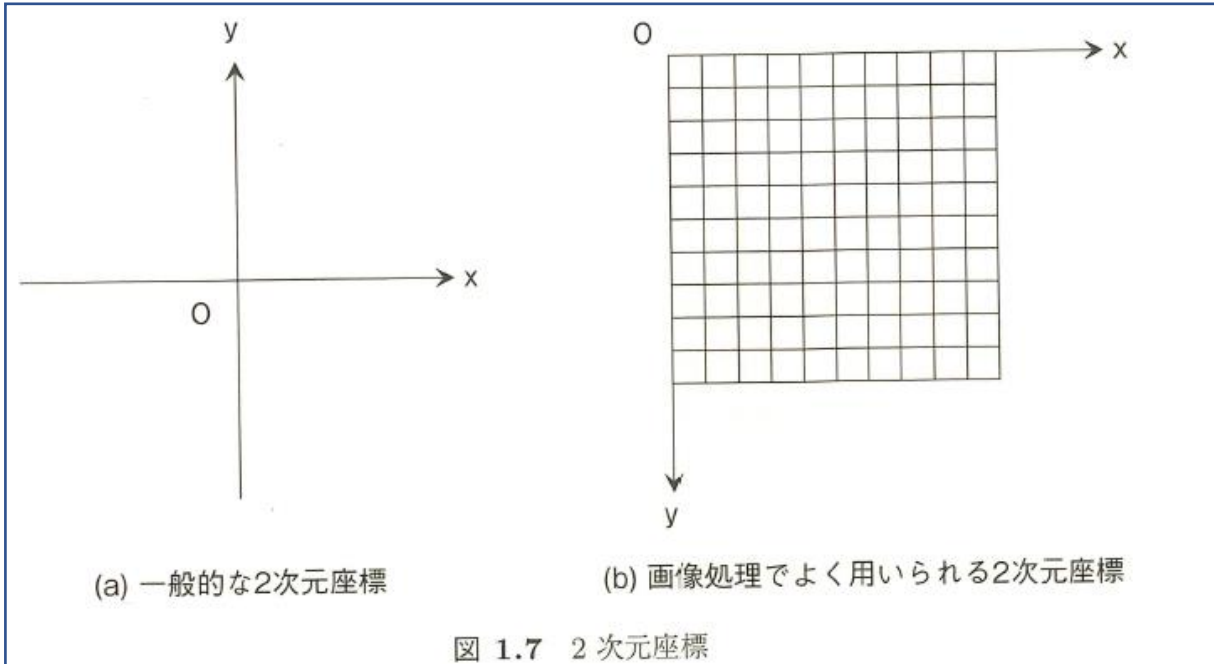
# 画像タイプ

	JPEG（ジェイペグ）	GIF（ジフ）	PNG（ピング）	BMP（ビーエムピー）
拡張子	.jpg（.jpeg）	.gif	.png	.bmp
特徴	<ul style="list-style-type: none"> <li>フルカラー1670万色を表現できる</li> <li>非可逆圧縮の画像形式</li> <li>背景の透過は行えない</li> <li>デジタルカメラの画像形式として知られている</li> </ul>	<ul style="list-style-type: none"> <li>256色で構成される</li> <li>可逆圧縮の画像形式</li> <li>背景透過も可能</li> <li>アニメも可能</li> </ul>	<ul style="list-style-type: none"> <li>フルカラー1670万色を表現できる</li> <li>可逆圧縮の画像形式</li> <li>背景透過も可能</li> <li>ウェブ用として有用</li> </ul>	<ul style="list-style-type: none"> <li>Microsoft Windowsのために作られた画像形式</li> <li>圧縮処理がされていない大きい画像</li> </ul>
用途	写真など、色数の多い画像に適している	ロゴや図版、イラストなど、色数の少ない画像に適している	写真と背景透過をどちらも実現したい場合やアニメ・漫画等に	解像度の高い画像が必要な場合に
容量	用途によって異なる ※圧縮率を上げれば上げるほど容量は小さく、劣化度合いは大きくなる ※ウェブでは、解像度72dpiが一般的	比較的小さい	比較的小さい ※ 同じ写真を同解像度で保存する場合、JPEGよりは少し容量が大きくなる	大きい
ウェブ対応	○	○	○	×

# 演習1-1

- LMSに添付してある、lena.bmpを皆さんのPCにdownloadして、大きさ、解像度、ビットの深さ、等を確認してください
- それ以外の画像に対しても、大きさ、解像度、ビットの深さ、等を確認してください

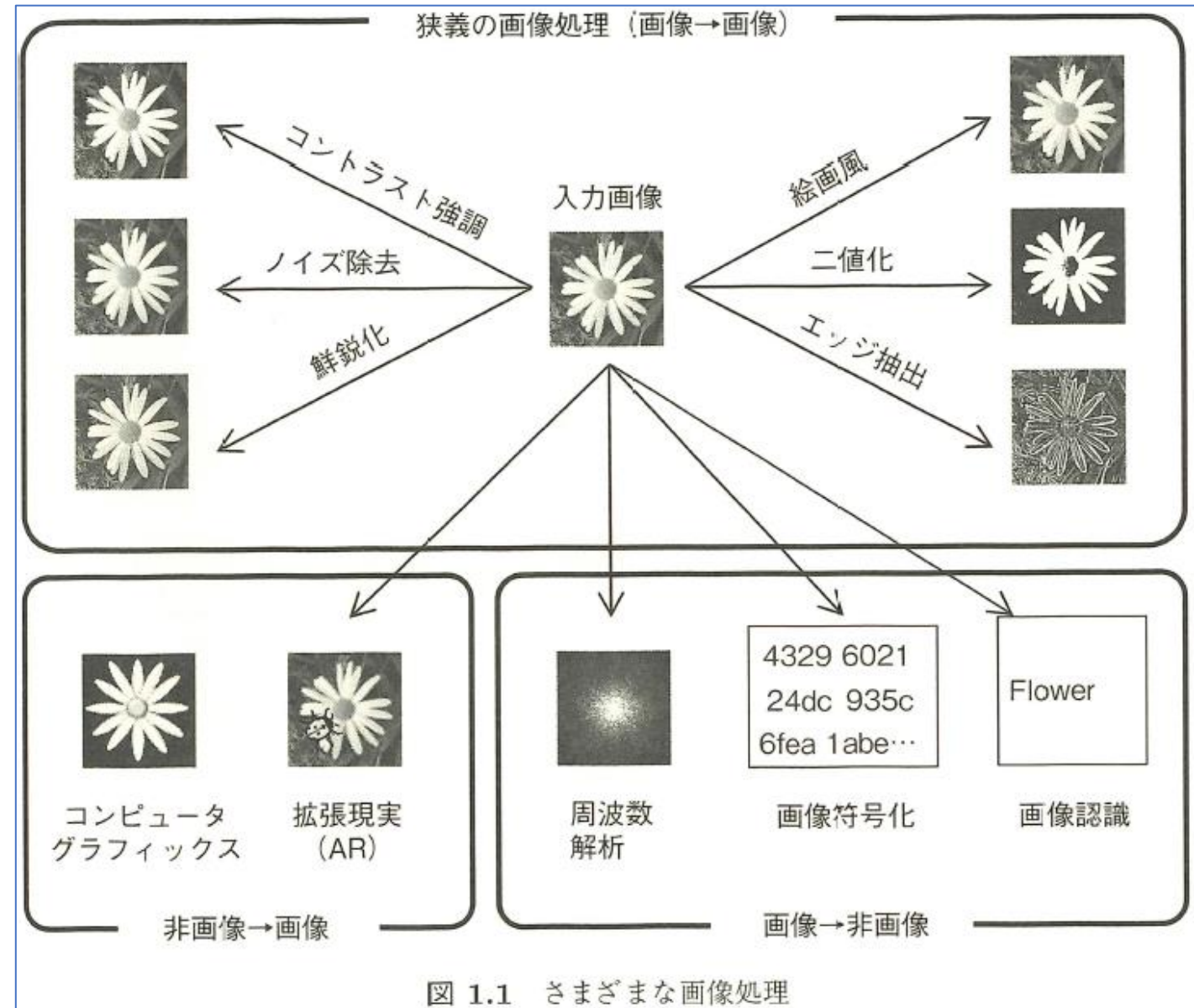




右手系が多いが、左手系も使われることはある

# 画像処理とは

- 画像を扱う信号処理全般



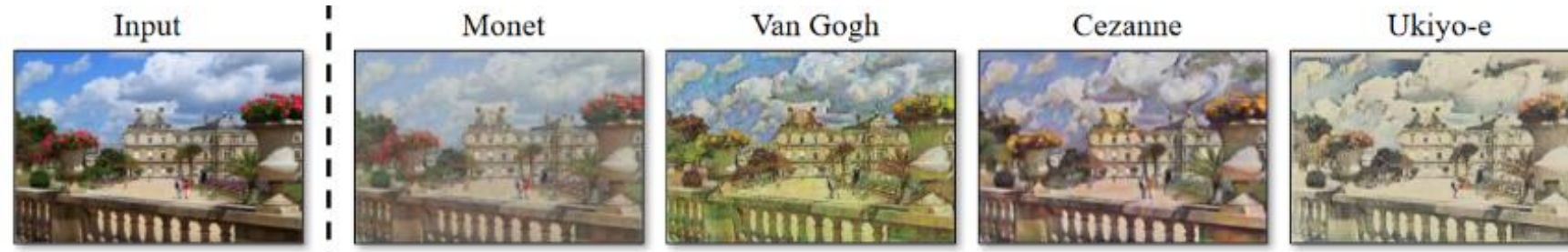
# 画像処理の目的

- 画像の見やすさを向上
  - コントラスト強調、ノイズ除去、鮮鋭化
- 画像中に存在する物体を計測
  - サイズ、物体の数
- 画像中に存在する物体を識別・認識
  - 画像認識(image recognition)
- 欲しい画像を検索:
  - 画像検索(image retrieval)

金塊に囲まれた黒熊



- 欲しい画像を生成
  - 画像生成、GAN, Stable Diffusion





# 画像認識とは

- 画像に写る内容を理解すること



# シーン認識と物体認識

- 物体認識

- 入力画像に写る物体を理解し、適切なラベルを付与する

- シーン認識

- 実世界の環境。複数の物体が存在



図 1.4 シーン認識.

原田達也 “画像認識” 講談社 p6



## インスタント認識とクラス認識

- インスタント認識
  - “橋”ではなく、“ゴールデンブリッジ”
  - 英語の **it**
- クラス認識
  - “ゴールデンブリッジ”ではなく、“橋”
  - 英語の **one**



- 画像の入力装置、出力装置には、どんなものがあるでしょうか？
- 入力
  - カメラ：デジカメ
  - 入力
- 出力
  - Monitor PC, Projector
  - 液晶Display, 液晶色々、量子何多羅寛太ら

# カメラのレンズによる実像の形成

レンズの倍率

$$\frac{BB'}{AA'} = \frac{B'O}{A'O} = \frac{b}{a} \quad \dots\dots ①$$

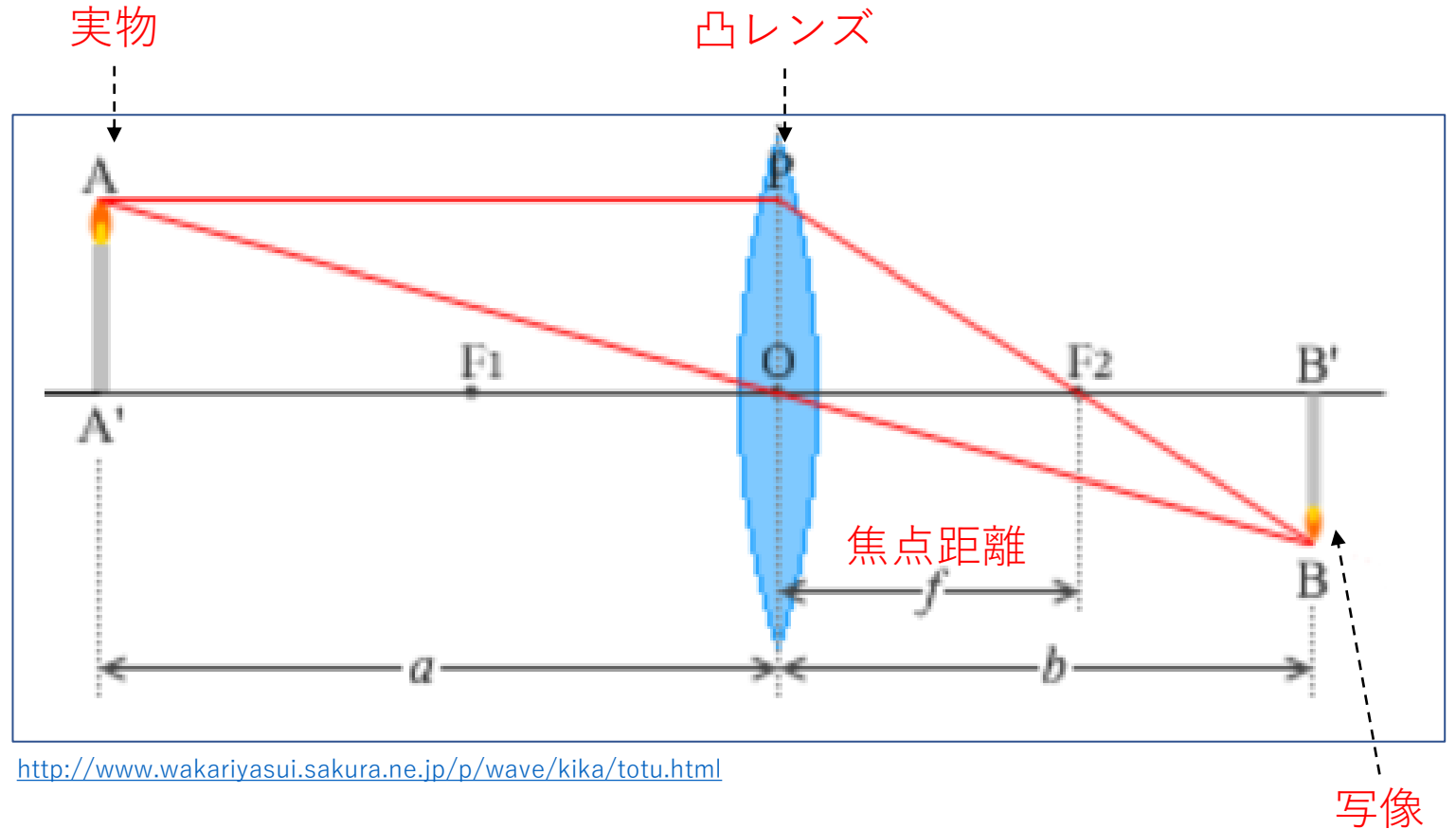
$$\frac{BB'}{AA'} = \frac{BB'}{PO} = \frac{B'F_2}{OF_2} = \frac{b-f}{f}$$

$$\frac{b}{a} = \frac{b-f}{f}$$

$$= \frac{b}{f} - 1$$

$$\frac{1}{a} = \frac{1}{f} - \frac{1}{b}$$

$$\frac{1}{a} + \frac{1}{b} = \frac{1}{f} \quad \dots\dots ② \quad \text{レンズの公式 or 写像公式}$$



# スキャナによる画像入力

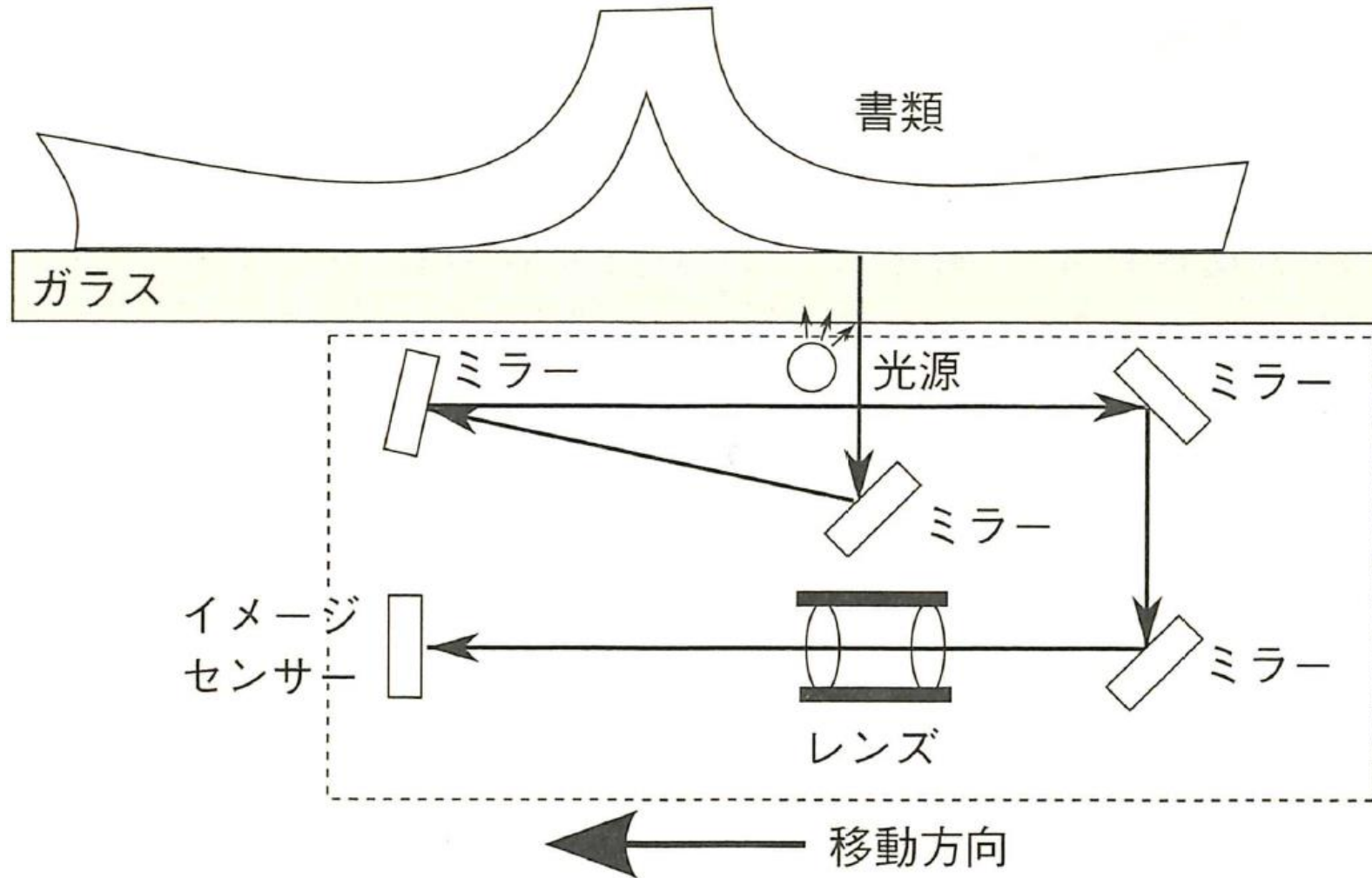
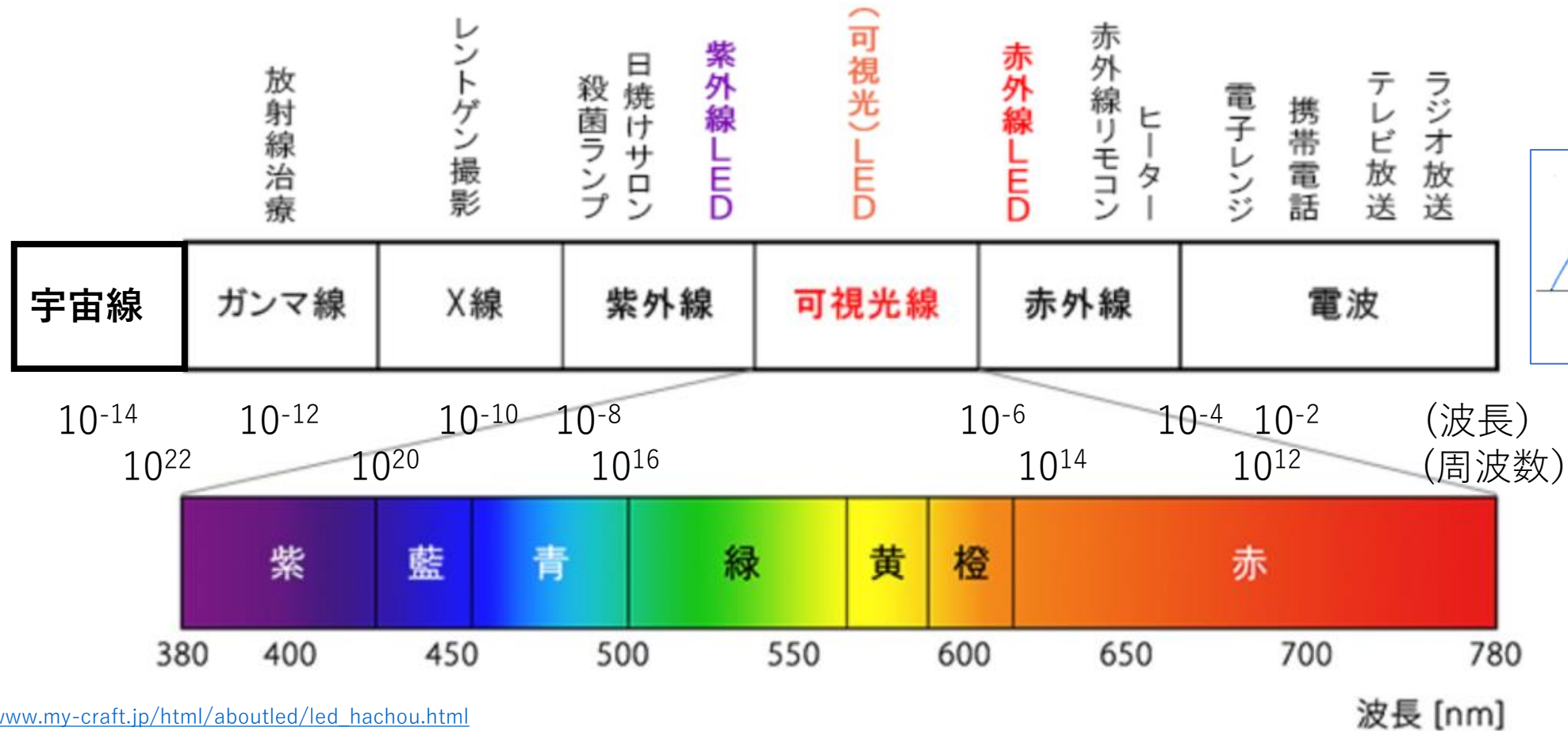


図 2.13 フラットベッドスキャナーの内部構造

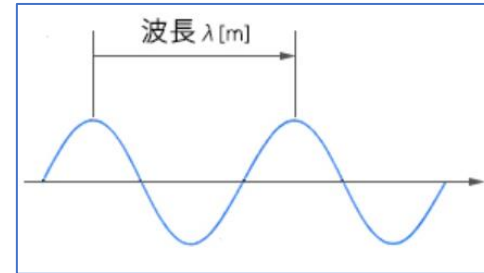
教科書 p19

# 光とは？

- 電磁波と呼ばれる空間を伝わる波の1種



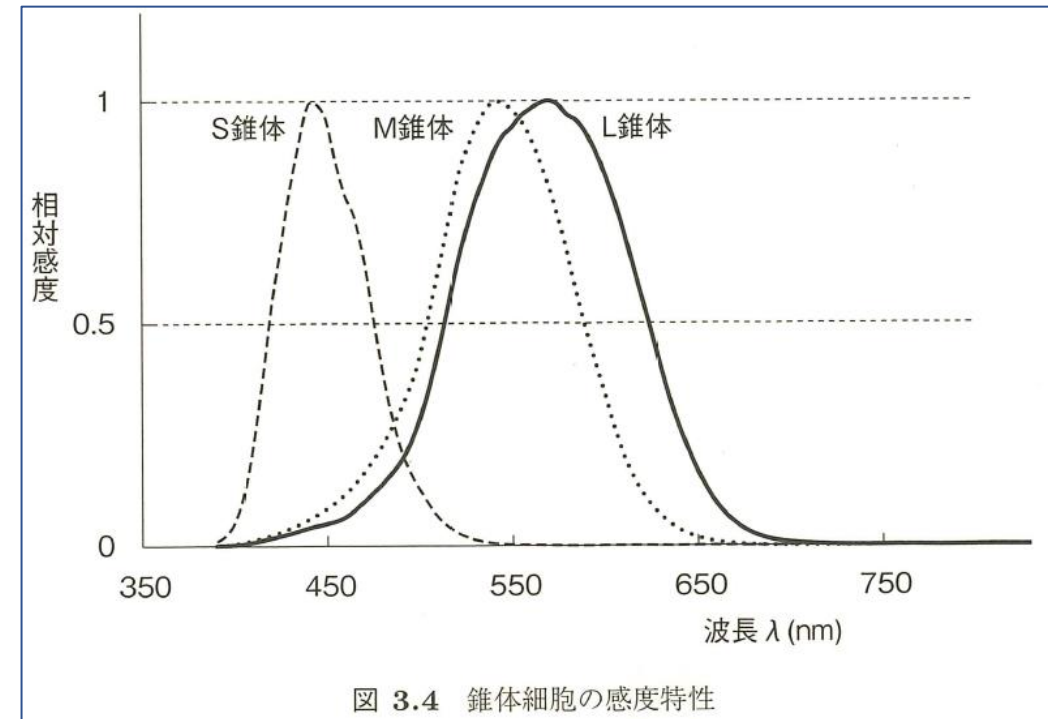
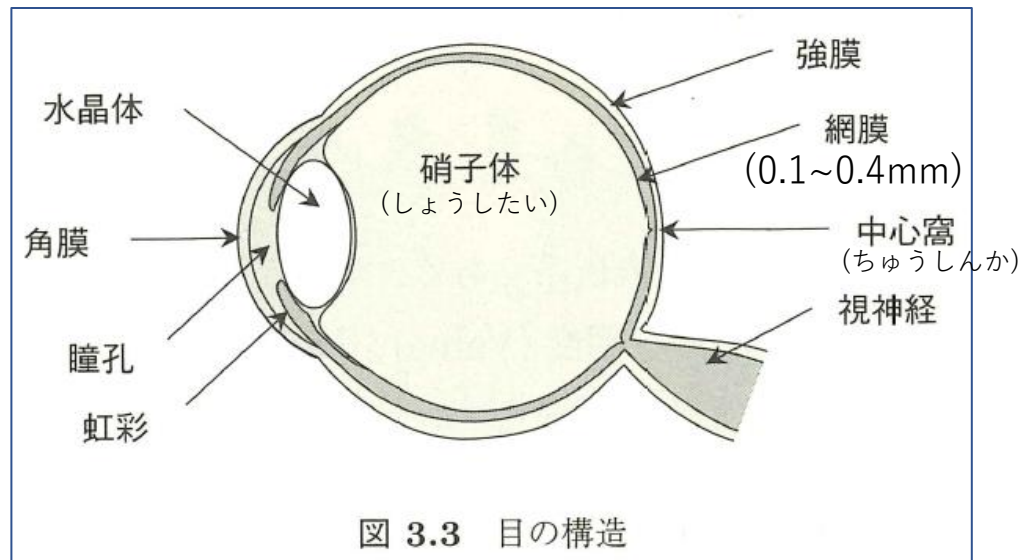
波長×周波数=速度  
(光=約3×10<sup>8</sup>m/sec)





# 人間の目と色知覚

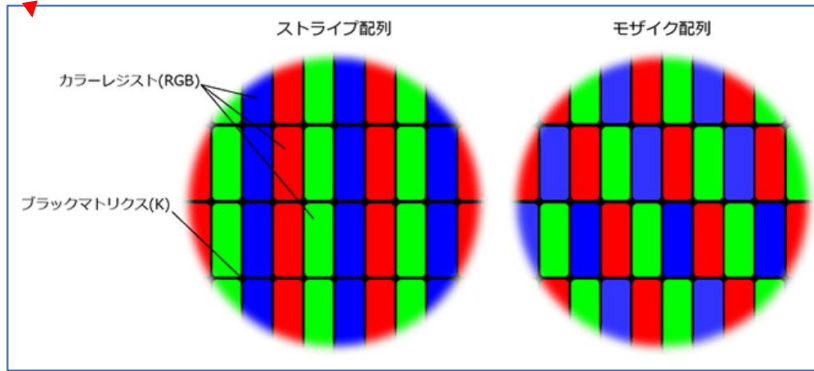
- 瞳孔から取り込まれた光は角膜と水晶体で屈折し、網膜上に像を結ぶ
  - 遠くの物を見るときは水晶体を薄くし、近くの物を見るときは厚くする
- 虹彩は光の量に応じて瞳孔の大きさを調整する
- 網膜の視細胞→光が電気信号に変換→視神経→大脳
  - 桿(かん)体細胞→感度は高い（明るさに反応）が色には反応せず
  - 錐体細胞→(S,M,L) 3種類で色を識別



教科書 p25

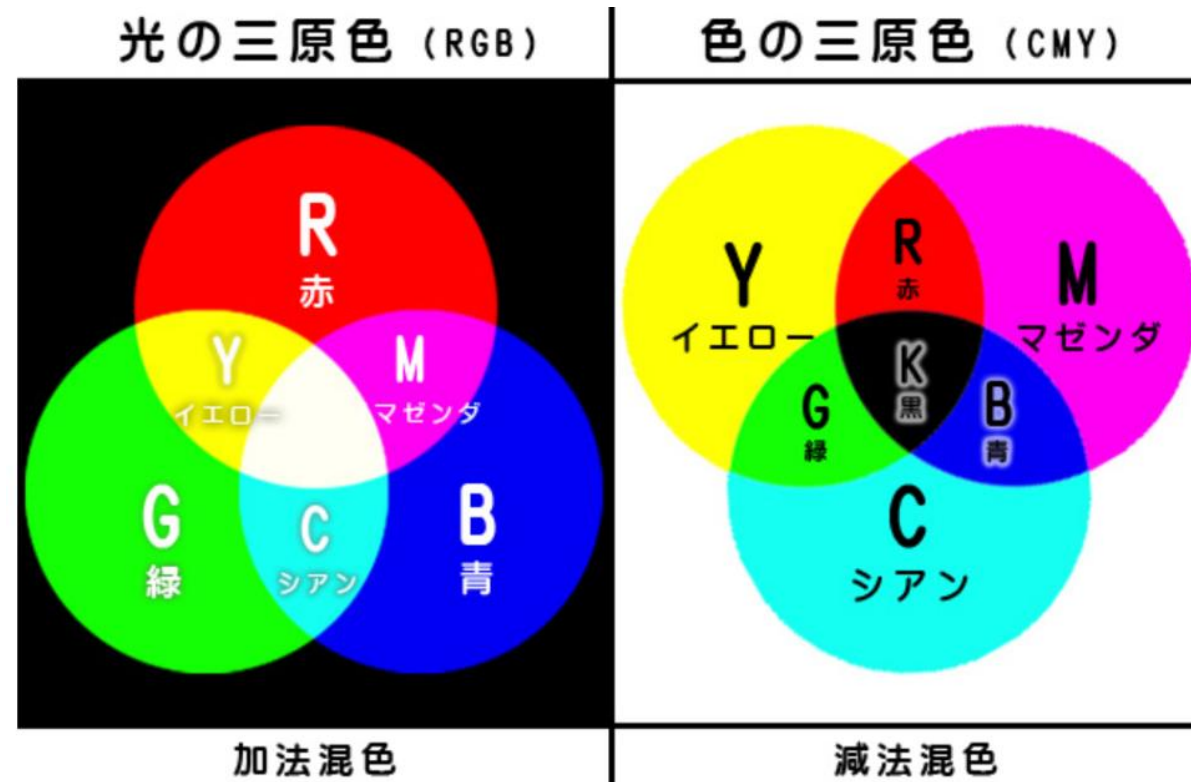
# 加法混色と減法混色(こんしょく)

- 加法混色：光、ディスプレイ
  - 同時加法混色：3種類の色を同時に1ヶ所に投影→ライトの重ね合わせ
  - 並置加法混色：複数の小さい点が近くに存在→液晶ディスプレイ
  - 継時加法混色：同じ場所でごく短期間ずつ異なる色を提示→コマ
- 減法混色：絵具



<https://www.youtube.com/watch?v=OA4tNWJ4u14>

<http://www.amy.tokyo/color/>  
<https://www.toyo-visual.com/ja/products/fpdcf/colorfilter.html>



# HSV色空間

(メディア処理実習6日目p61)

- 色を「色相(Hue)」 「彩度(Saturation)」 「明度(Value・Brightness)」 の3要素で表現する方法

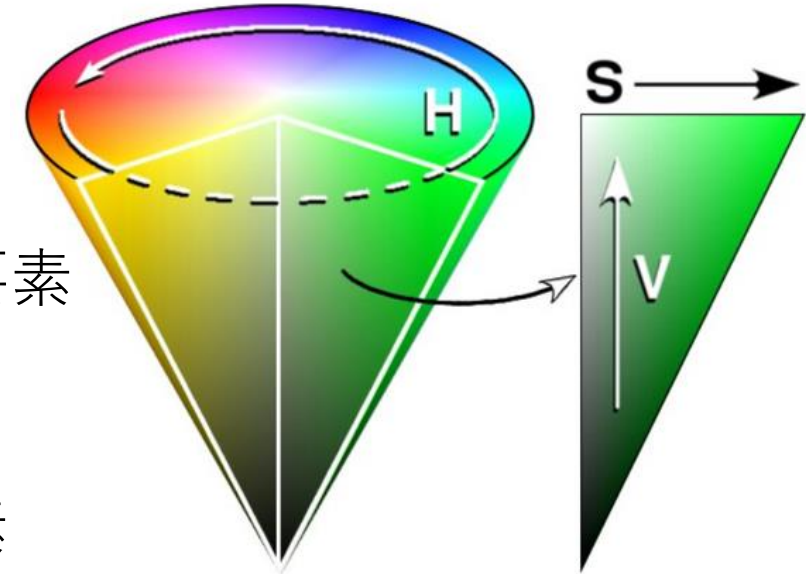
➤色相：具体的な色を定義する要素



➤彩度：色相で定義された色の鮮やかさ・濃さを表す要素



➤明度：色相で定義された色の明るさ・暗さを表す要素



- “色の範囲”の定義をしやすい

<https://www.peko-step.com/html/hsv.html>

[https://ja.wikipedia.org/wiki/HSV%E8%89%B2%E7%A9%BA%E9%96%93#/media/%E3%83%95%E3%82%A1%E3%82%A4%E3%83%AB:HSV\\_cone.jpg](https://ja.wikipedia.org/wiki/HSV%E8%89%B2%E7%A9%BA%E9%96%93#/media/%E3%83%95%E3%82%A1%E3%82%A4%E3%83%AB:HSV_cone.jpg)

- 現在実用化されている、画像処理、画像認識の応用で、皆さんが気に入っている、お世話になっている、役に立っているものにはどのようなものがあるでしょうか？また、将来、どのような画像処理技術があると思いますか？

# 本日のまとめ

- 画像とは
  - 写真、絵、図面等を電子化したもの
- 解像度と量子化
  - 解像度：画像の細かさ
  - 量子化：色、輝度の階調
- 画像処理と画像認識
  - 画像処理：画像を扱う信号処理全般
  - 画像認識：画像に写る内容を理解すること
- 画像の入出力
  - 入力：カメラ、センサ、スキャナ
  - 出力：テレビ、プリンタ
- 光とは？
  - 電磁波の1種
  - 量子化：色、輝度の階調
- 加法混色と減法混色
- HSV色空間



# 宿題1

1. Google colabを立ち上げ、`print("hello world")`の出力結果のsnapshotをLMSに提出して下さい



2. Local PC上で、LMSに添付してある“hello.py”に対し、“python hello.py”を実行し、その出力画面のsnapshotをLMSに提出して下さい。Local PCに、pythonとopenCVが正しくinstallされていれば、下記のように、openCVのversionが表示されます

- 提出期限 B=4/15(土), A=4/17(月),  
どちらも9:00

