

# 実世界情報実験1

## D1/D2クラス

## 画像処理テーマ

島田伸敬

13:00より開始

Google Colaboratory ( <https://colab.research.google.com/> ) を用いて実験を行います。Googleアカウントが必要です。

manaba+Rのコンテンツから lesson3.zip をダウンロードしておいてください。

# 課題提出ルール（現時点）

- 画像処理テーマ第 $n$ 回授業で課題 $n$ の解説を行う。
- 課題 $n$ の一次締切は画像処理テーマ第 $(n+1)$ 回授業日の12:00とする。但し課題6は一次締め切りなし。
- 一次締切までに提出されたものはTAがチェックを行い、結果を受講生本人に公開（フィードバック）する。但し文章で回答する小課題については内容のチェックは行わず、回答があれば10点とする。
  - 小課題それぞれについて10点満点の点数を付ける。これは小課題単体の点数で、その単純和で最終成績を決めるわけではない。例えば課題1-1と課題5-1では難易度が違うので同じ10点でも最終成績に与える影響は異なる可能性がある。
- 課題 $n$ の一次締切内提出物は第 $(n+1)$ 回授業中にTAがチェックする。全提出物について完了したら公開する(各受講生は自分の成績のみ閲覧可能)。
- 全課題の最終締切は画像処理テーマ第7回(最終回)授業の1週間後とする。
- 最終締切までは何度でも再提出できるが、一次締切以後の提出物はフィードバックされないかもしれない。
- TAの負荷、授業の進捗度合いによって上記ルールは変更される可能性がある。

# 計算機上での画像の表現



細かい区画(画素, pixel)ごとに色を記憶し、その集まりとして画像を表す

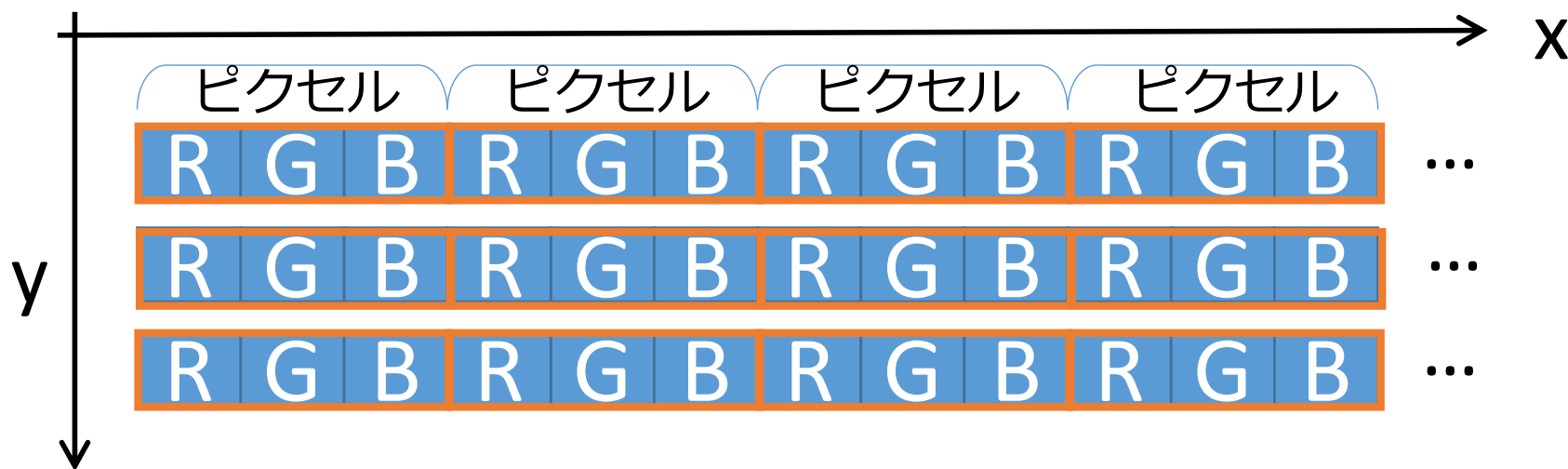
	x=0	x=1	x=2	
	画素			
y=0	B	G	R	...
y=1	B	G	R	...
y=2	B	G	R	...

各画素の位置は画像の左上を原点とした座標系で表現することが多い。各画素が濃度や色情報を持つ。

# 画素値

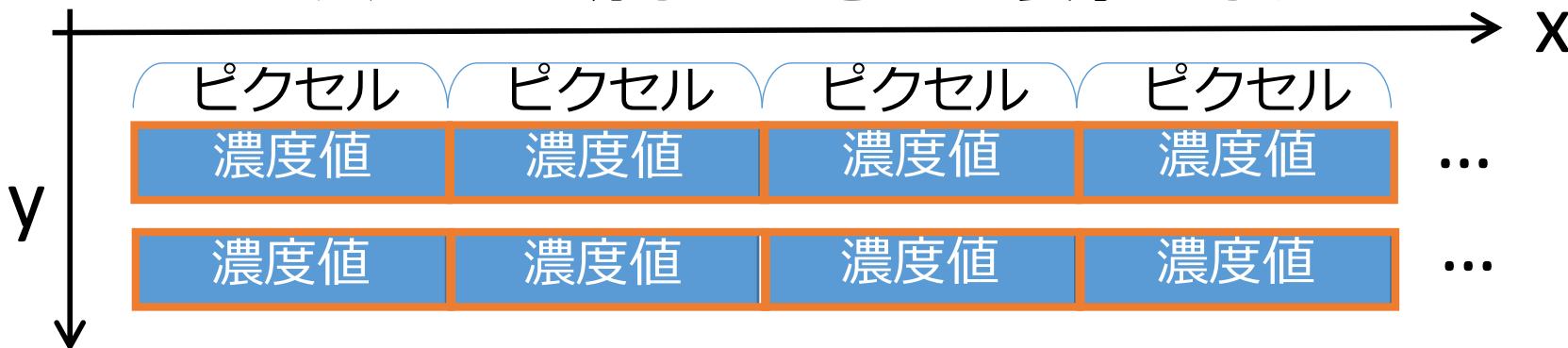
## カラー画像

各画素がRGB(Red, Green, Blue)の3つの成分を持つ。各成分の大きさ（階調値）は8bit(0 - 255)で表すことが多い。



## 濃淡画像

各画素が1つの成分（濃度値）を持つ。成分の大きさを明るさと思って表示する。



# 2値化

データを0/1の情報に置き換える操作

複雑な情報からあえて情報量を削減することで  
注目したい情報を浮かび上がらせる



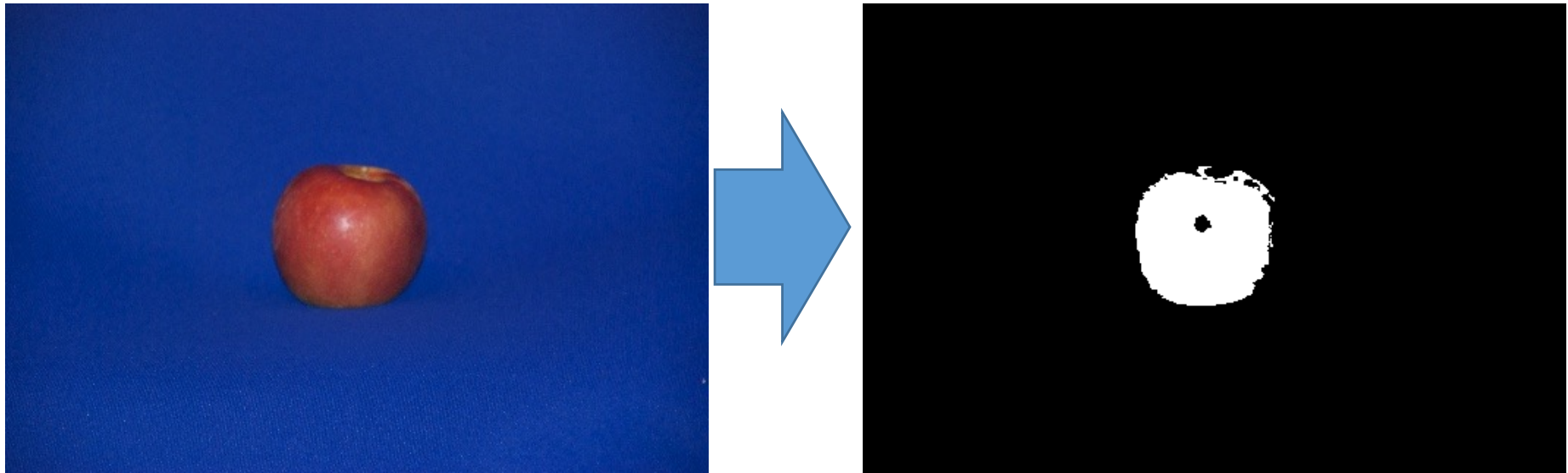
温度分布(15°C~28°C)

ある温度以上or以下の情報のみ

細かい温度の高低はわからなくなるが、

猫の写っている場所が明確になった

# 色による2値化

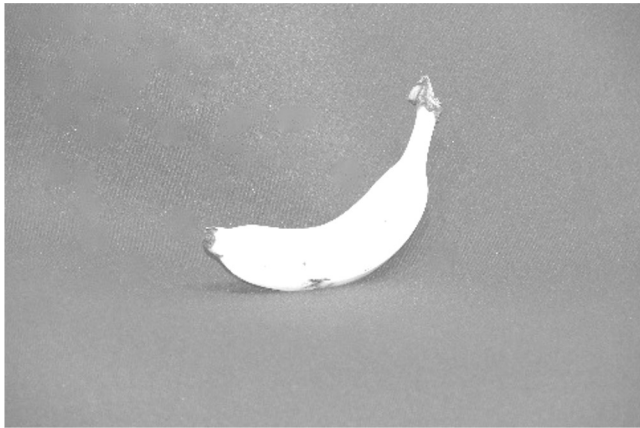


あらかじめ学習しておいたリンゴの色に似ている色の部分を白、それ以外を黒とした  
**2値化の条件を厳密に定義する必要がある**

この例では**似ている色**を定義しないといけない

# 輝度を用いた2値化

画像中の**明るい部分を白、それ以外を黒**としたい



全体に明るい



全体に暗い

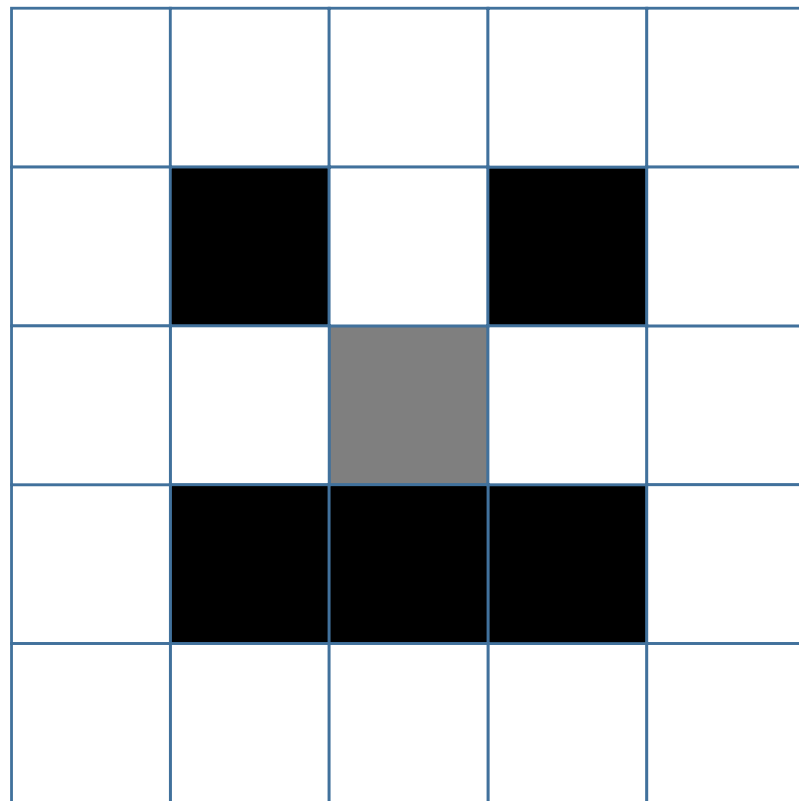
明るい画素の輝度、暗い画素の輝度が

**画像によって異なる**

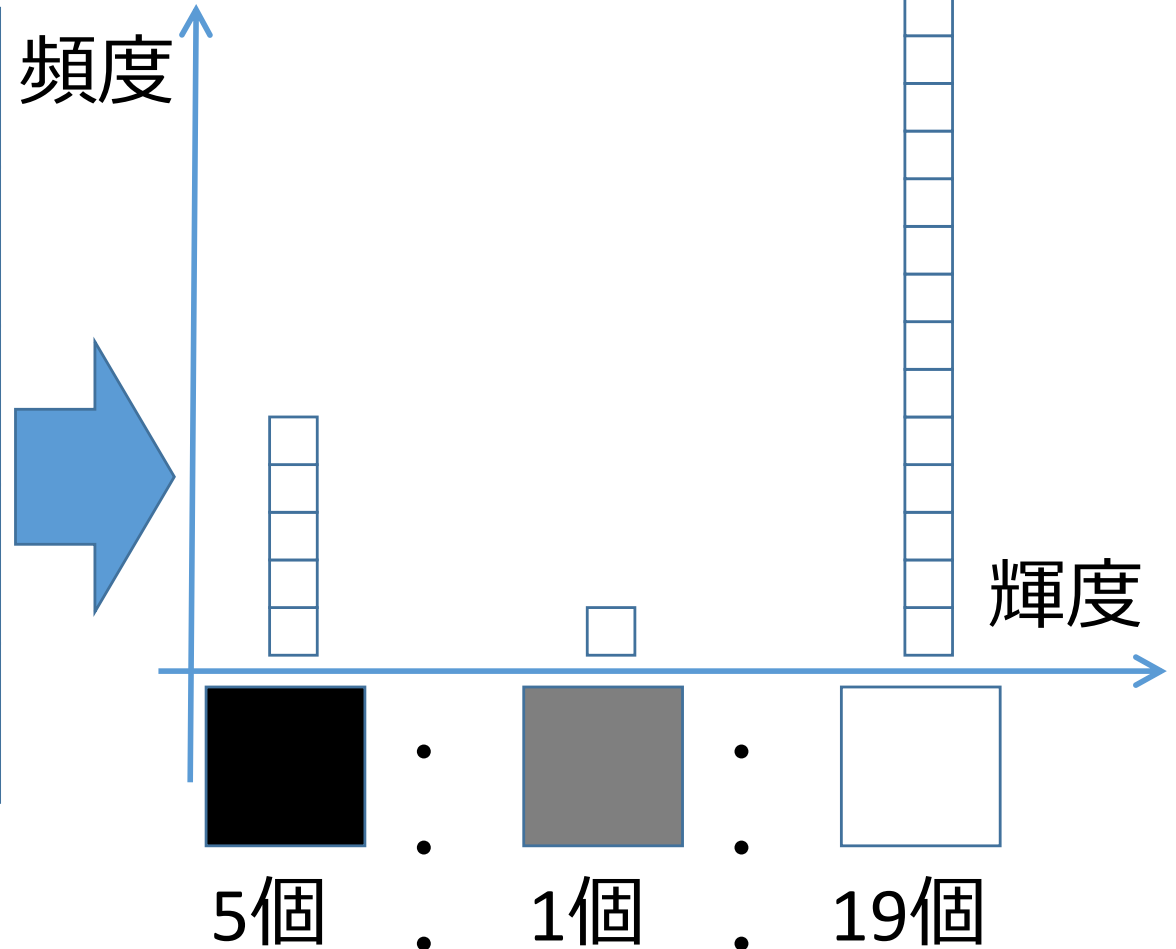


“明るい/暗い”の基準となる輝度(**閾値**)を  
画像にあわせて設定する必要がある

# 輝度値のヒストグラム



入力画像(5×5)



それぞれの輝度が、**画像中に何個現れたか**を表す  
**現れ得る輝度それぞれについて個数を数えればよい**



# 輝度値のヒストグラムの例

それぞれの輝度が、**画像中に何個現れたか**を表す

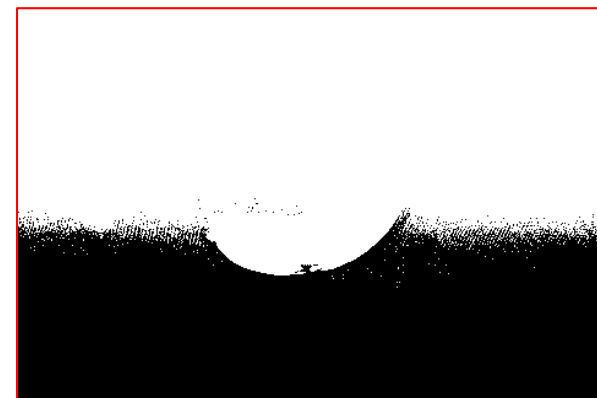
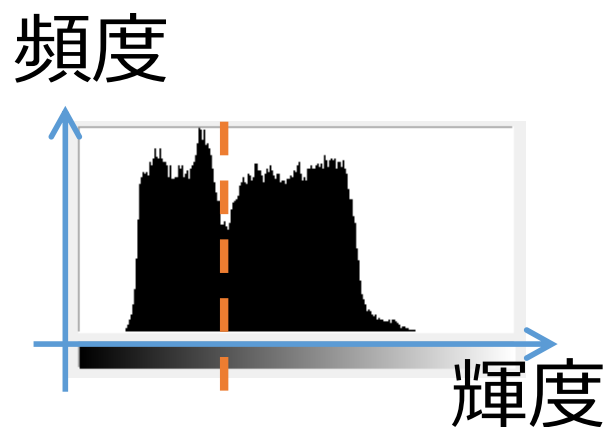
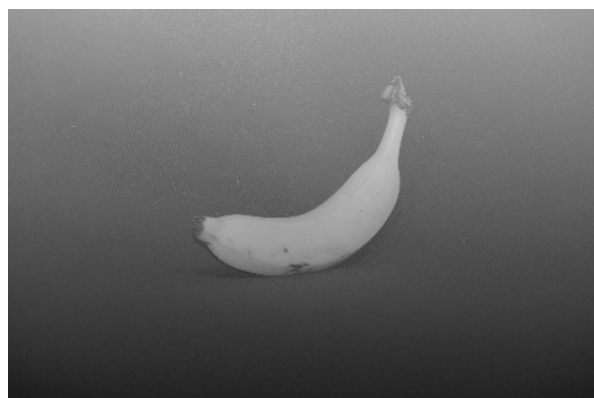
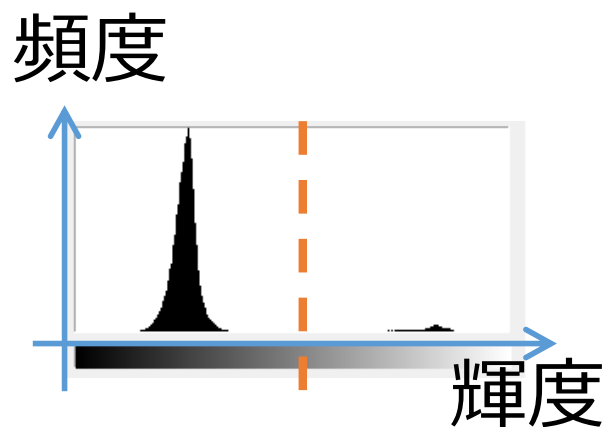


この色の画素は非常に多い！

この色の画素は少ない

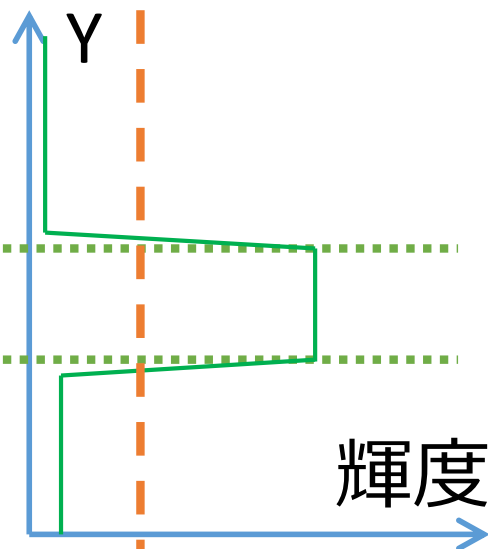
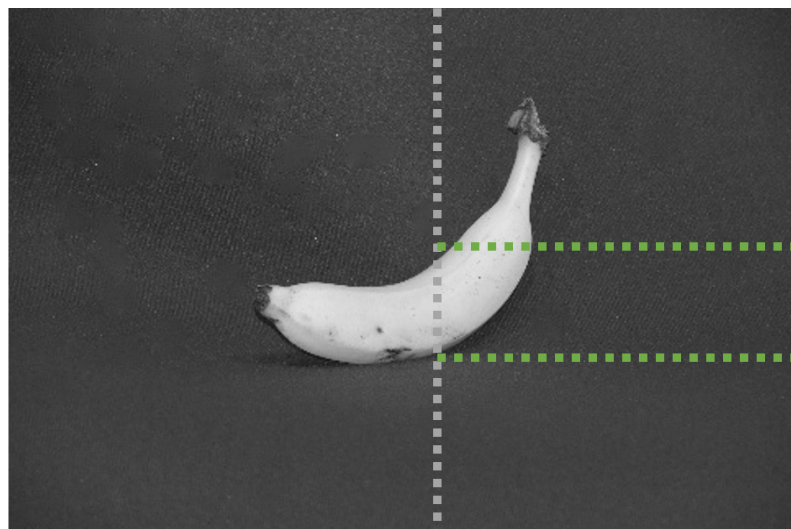
“明るい/暗い”の基準となる輝度(**閾値**)を  
**ヒストグラムを見て決めることもできる**

# 閾値を用いた2値化が難しい例

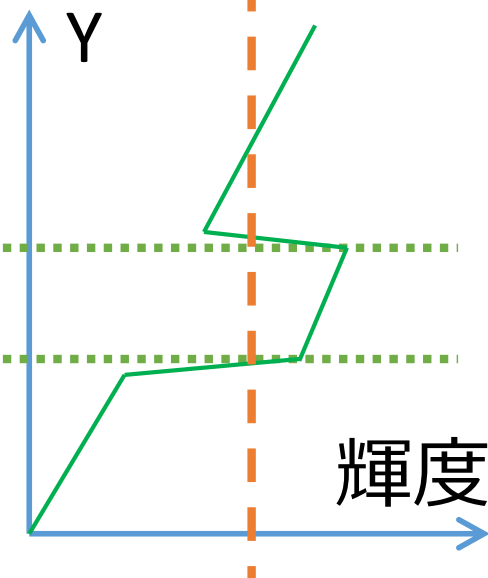
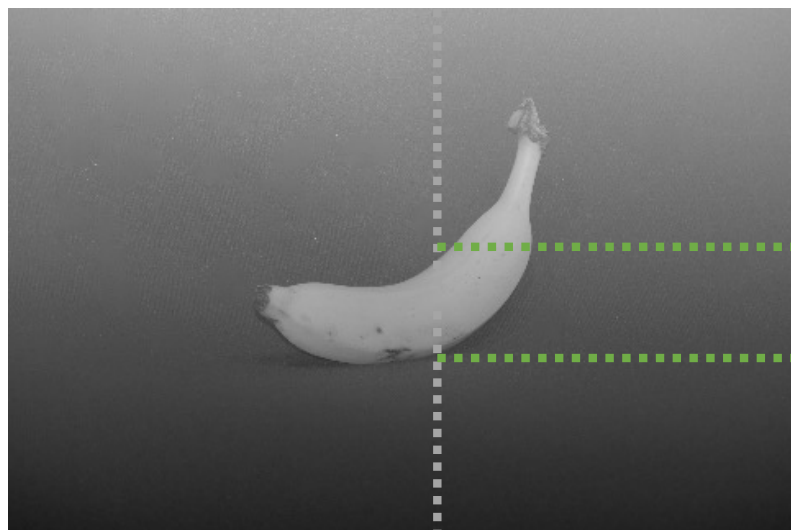


対象の輝度が一定でないと  
輝度の閾値を単純に使う2値化は難しい

# どういふ状況なのか？

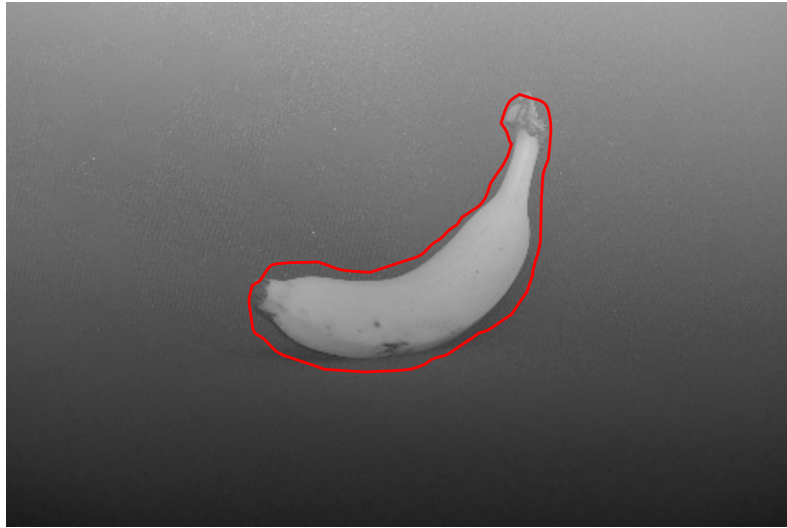


単純な閾値  
処理で背景  
と物体を分  
離しやすい



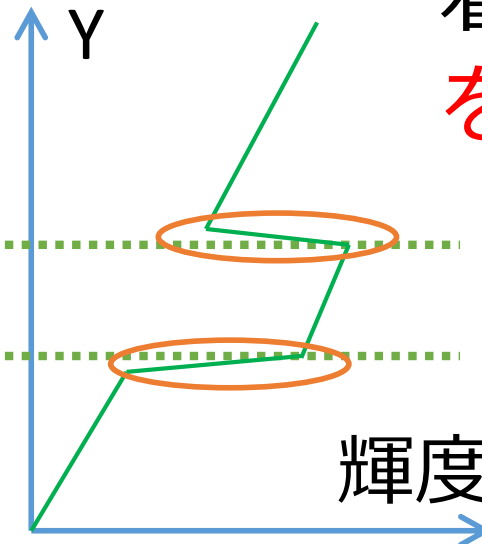
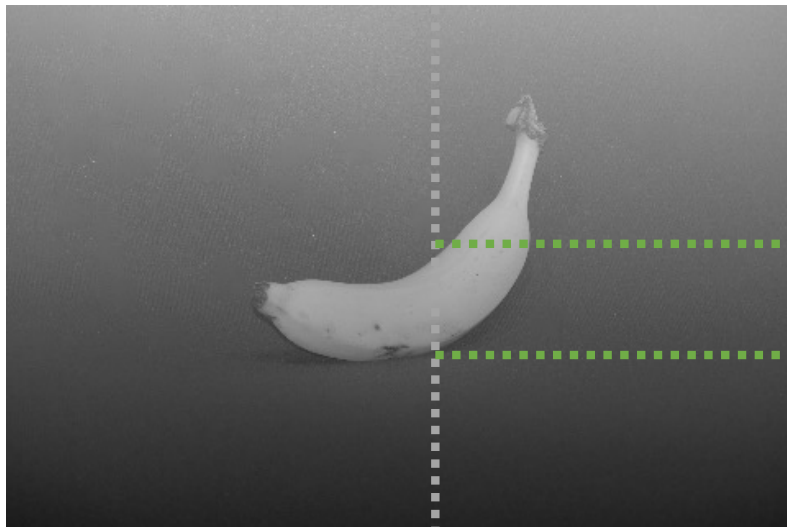
単純な閾値  
処理では分  
離が難しい

# どうすればよいか？

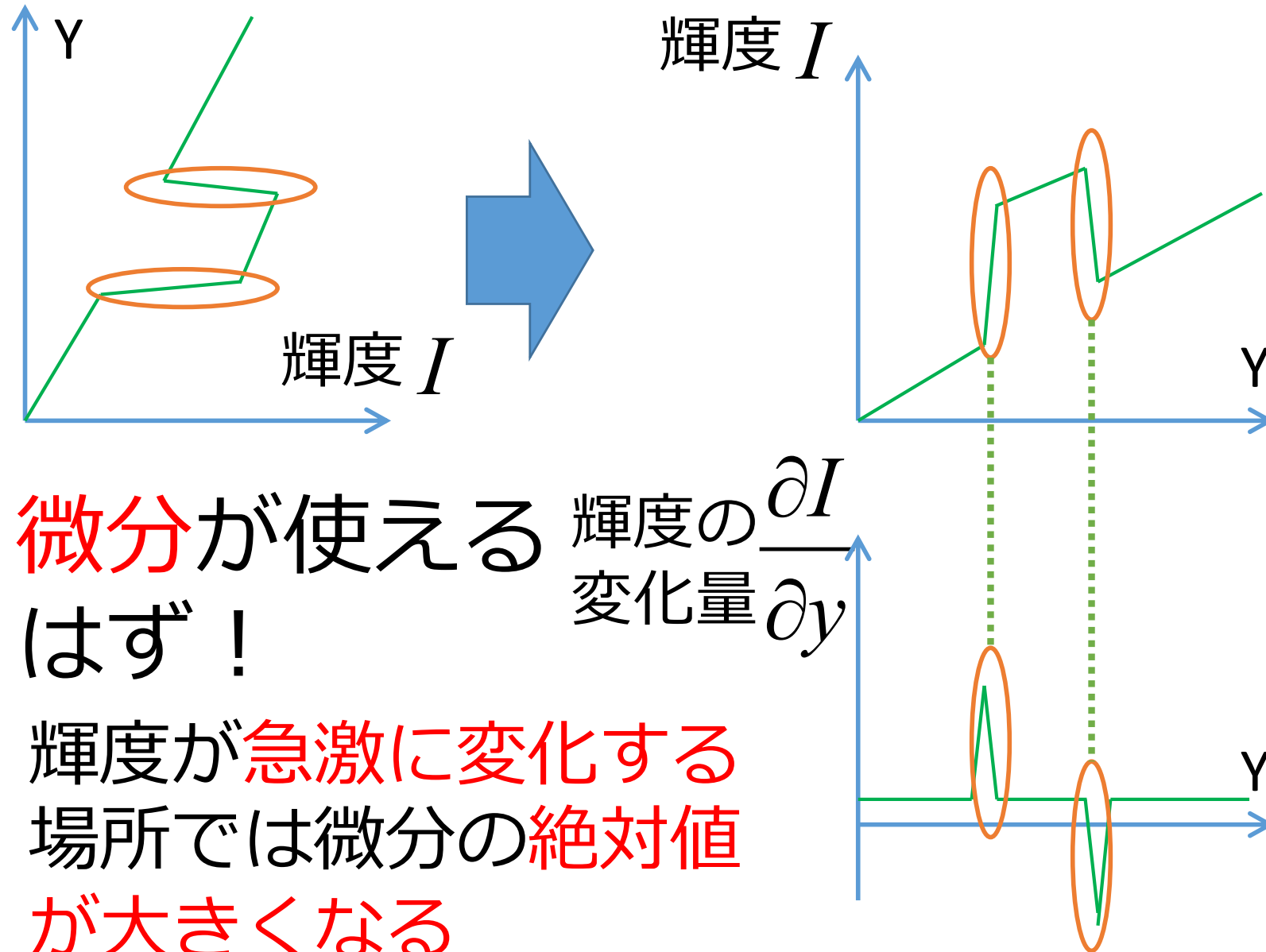


人間は輪郭を把握  
することで物体領  
域を切り出してい  
る

輝度の変化量に  
着目すれば輪郭  
を検出できる



# 輝度の変化量を測るには？



微分が使える  
はず！

輝度が急激に変化する  
場所では微分の絶対値  
が大きくなる

# 画像への微分をどう定義する

X=0 Y=0	X=1 Y=0	X=2 Y=0	X=3 Y=0
X=0 Y=1	X=1 Y=1	X=2 Y=1	X=3 Y=1
X=0 Y=2	X=1 Y=2	X=2 Y=2	X=3 Y=2

画像は離散データなので連続的な定義は使えない！

## 微分の定義

$$\frac{\partial I}{\partial y} = \lim_{h \rightarrow 0} \frac{I(x, y+h) - I(x, y)}{h}$$

$h \rightarrow 0$ の代わりに  
最小値 $h=1$ で止  
めてみる

$$\frac{\Delta I}{\Delta y} = \frac{I(x, y+1) - I(x, y)}{1}$$

# 画像上の微分の定義

連続的な定義は複数の書き方がある！

$$\begin{aligned}\frac{\partial I}{\partial y} &= \lim_{h \rightarrow 0} \frac{I(x, y+h) - I(x, y)}{h} \\ &= \lim_{h \rightarrow 0} \frac{I(x, y+h) - I(x, y-h)}{2h} \\ &= \frac{1}{4} \lim_{h \rightarrow 0} \frac{I(x+h, y+h) - I(x+h, y-h)}{2h} \\ &\quad + \frac{1}{2} \lim_{h \rightarrow 0} \frac{I(x, y+h) - I(x, y-h)}{2h} \\ &\quad + \frac{1}{4} \lim_{h \rightarrow 0} \frac{I(x-h, y+h) - I(x-h, y-h)}{2h}\end{aligned}$$



連続的には  
同じ定義

しかし  
離散化した  
結果はそれ  
ぞれ異なる

# 微分フィルタ

$$\frac{\Delta I}{\Delta y} = \frac{I(x, y+1) - I(x, y)}{1}$$

	$x$
$y$	-1
$y+1$	+1

狭い範囲の変化もとらえる

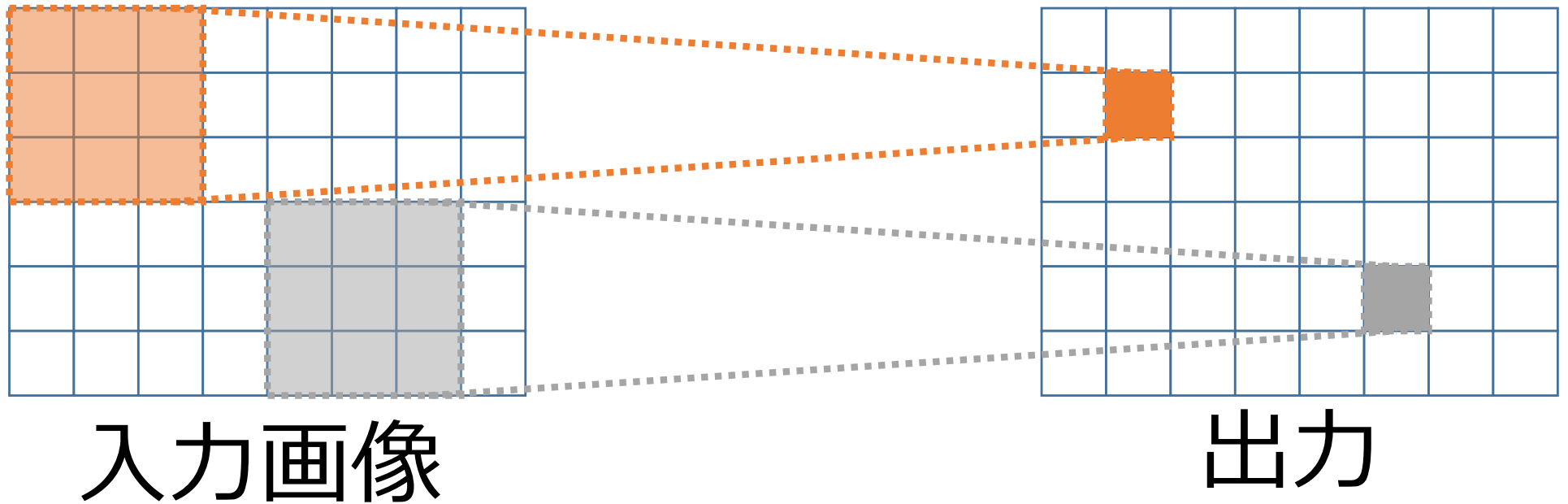
$$\frac{\Delta I}{\Delta y} = \frac{I(x, y+1) - I(x, y-1)}{2}$$

	$x$
$y-1$	-1
$y$	0
$y+1$	+1

上下対称



# フィルタ処理 (3×3の場合)



入力画像の複数画素から  
出力の値を決める

# Sobelフィルタ(Y方向)

$$\begin{aligned}\frac{\Delta I}{\Delta y} = & (-1) \times I(x-1, y-1) + (-2) \times I(x, y-1) + (-1) \times I(x+1, y-1) \\ & + 0 \times I(x-1, y) + 0 \times I(x, y) + 0 \times I(x+1, y) \\ & + (+1) \times I(x-1, y+1) + (+2) \times I(x, y+1) + (+1) \times I(x+1, y+1)\end{aligned}$$

	x-1	x	x+1
y-1	-1	-2	-1
y	0	0	0
y+1	+1	+2	+1

2画素だけ使う方法に比べてどういう利点があるか？

x方向のフィルタも同様に定義できる

# 勾配の大きさ

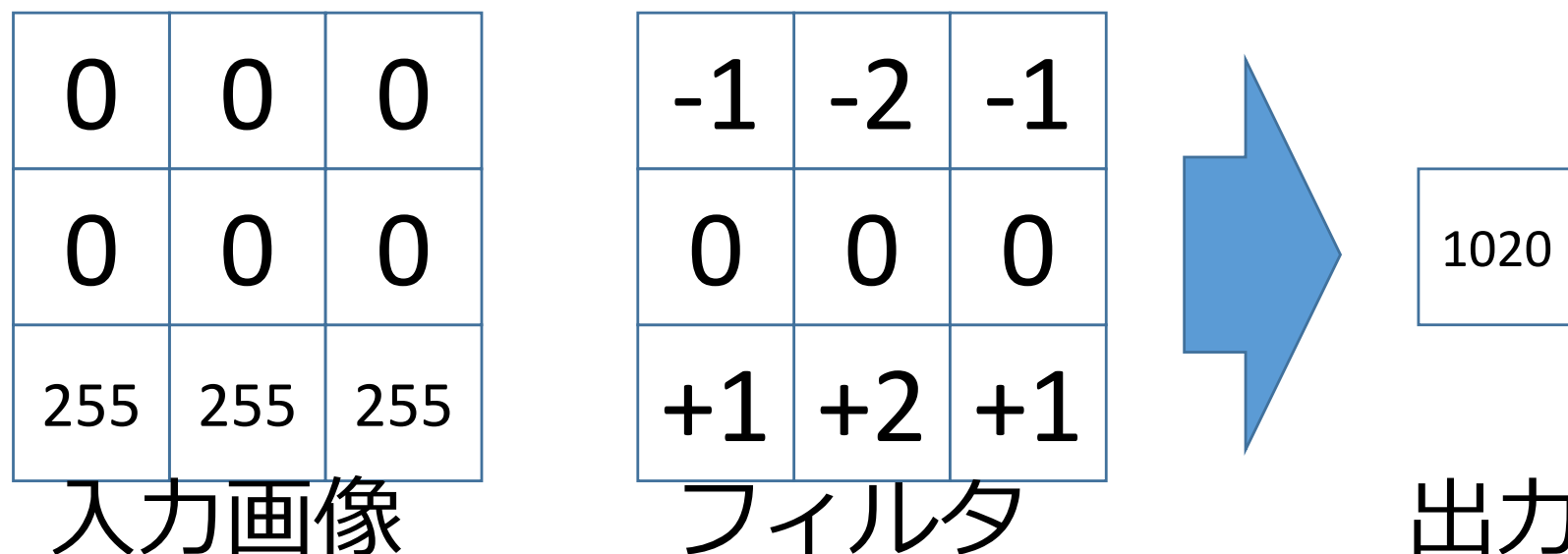
連続的な定義の類推を用いる

$$|\nabla I| = \sqrt{\left(\frac{\partial I}{\partial x}\right)^2 + \left(\frac{\partial I}{\partial y}\right)^2}$$



$$|\nabla I| = \sqrt{\left(\frac{\Delta I}{\Delta x}\right)^2 + \left(\frac{\Delta I}{\Delta y}\right)^2}$$

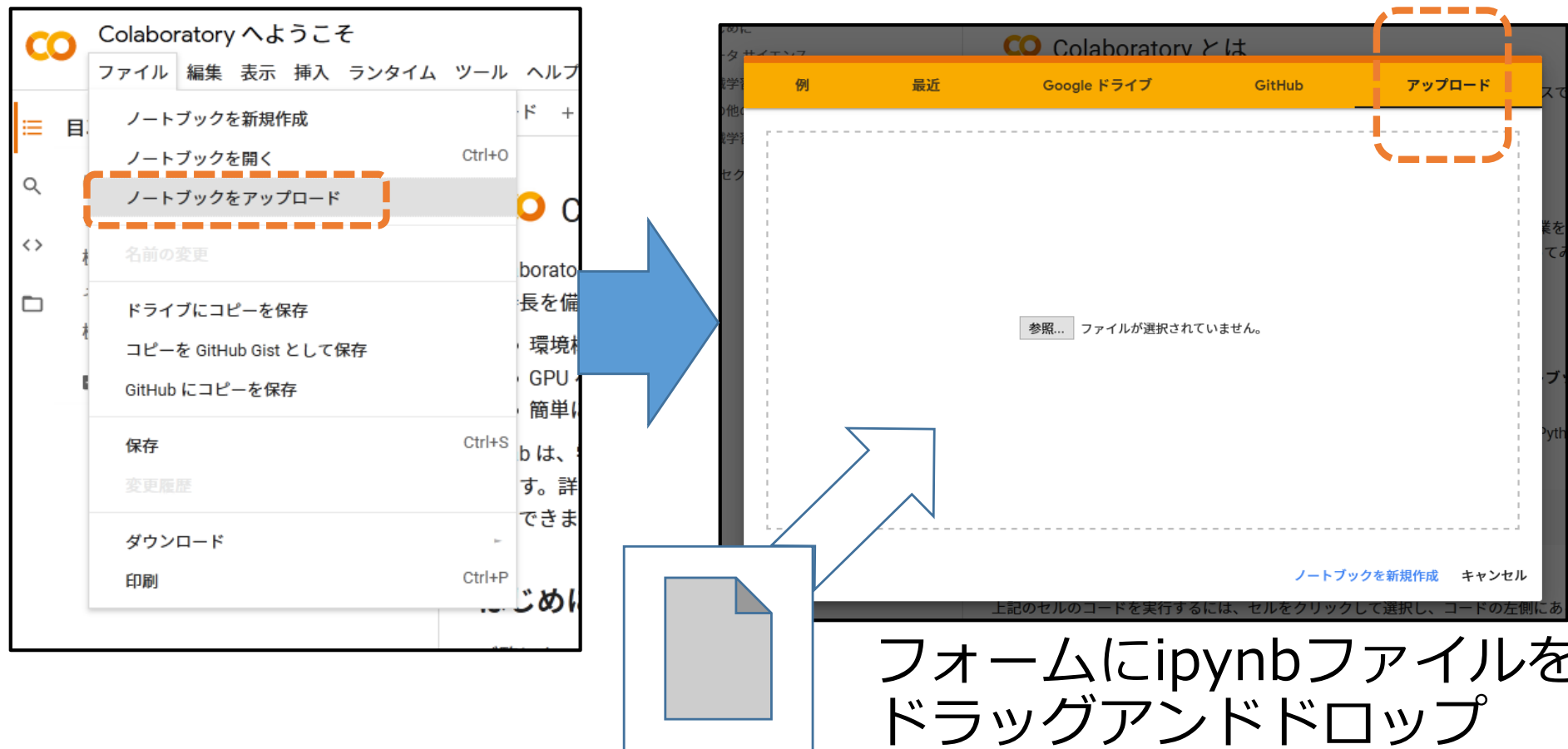
# 最大値に注意



フィルタの出力が255を超えることがある

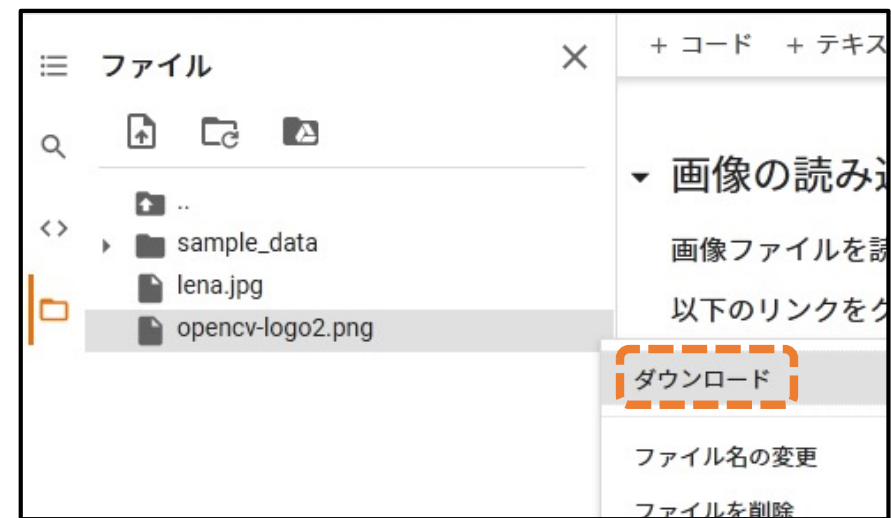
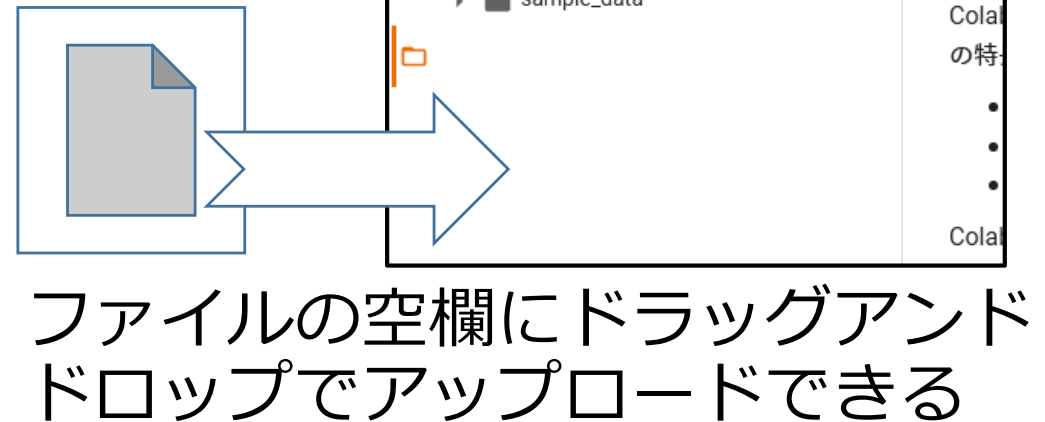
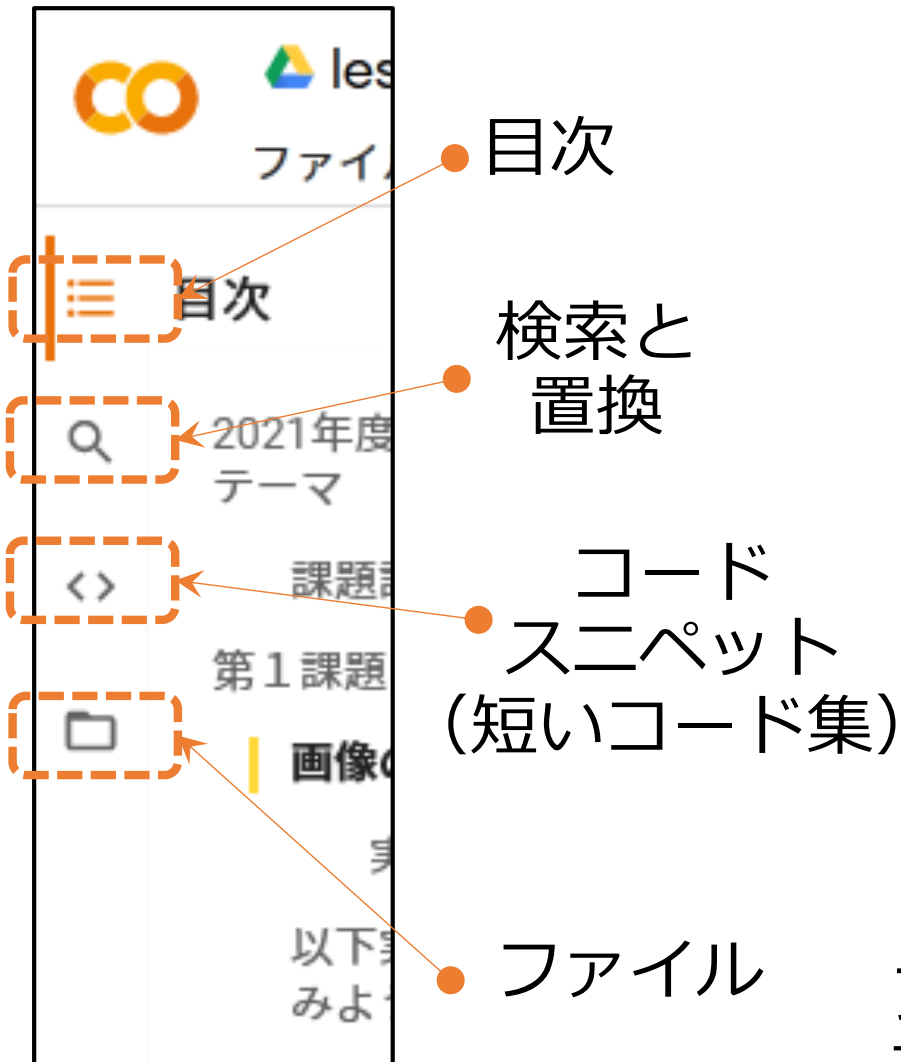
# ipynbファイルのアップロード

Google Colaboratoryを開いた画面で「アップロード」タブを選択するか、メニューの「ファイル」から「ノートブックをアップロード」を選択するとアップロード用のフォームが現れる。



フォームにipynbファイルをドラッグアンドドロップすればアップロードできる。

# Google Colaboratory 基本操作



ファイル名の右の「…」をクリックすればダウンロードを指示できる

# 提出についての補足

- ipynbファイル(実行結果も入ったもの)と保存した画像、利用した元画像を提出してもらいます。
- 回答欄のコードセルには `import` や `imread` (必要であれば)等を含め「**そのセルだけを実行すれば意図した状況が作れる**」ようにしておいてください。  
(他のセルでimportしたライブラリや他のセルで作成した変数に依存させない)
- プログラムが生成するファイルの名前は課題の指示に従う。  
例えば課題1-2なら `lesson1-2.png` とする。追加で保存したい場合は `lesson1-2-1.png`, `lesson1-2-2.png`などのようにする。
- 画像ファイルを保存する課題では、そのファイルを他のプログラムで開いたときにも意図した色で表示されるようにする。(必要であれば`cvtColor()`を使う)