

# 3. デジタル画像

(コンピューターサイエンス)

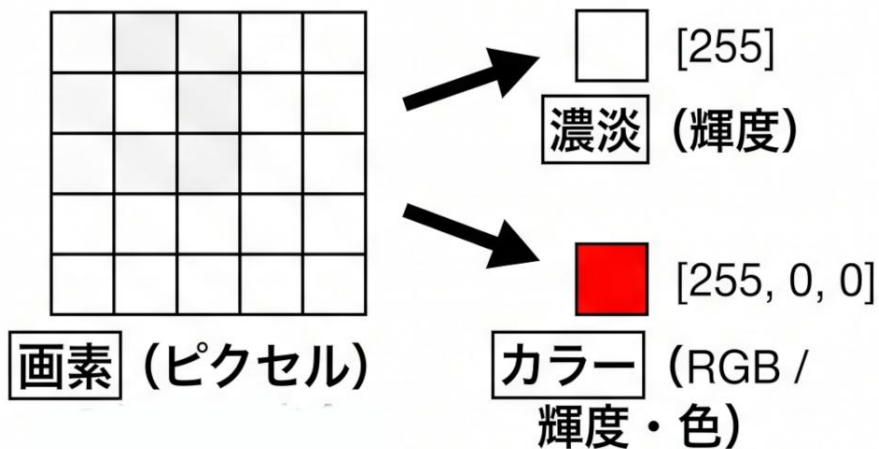
金子邦彦



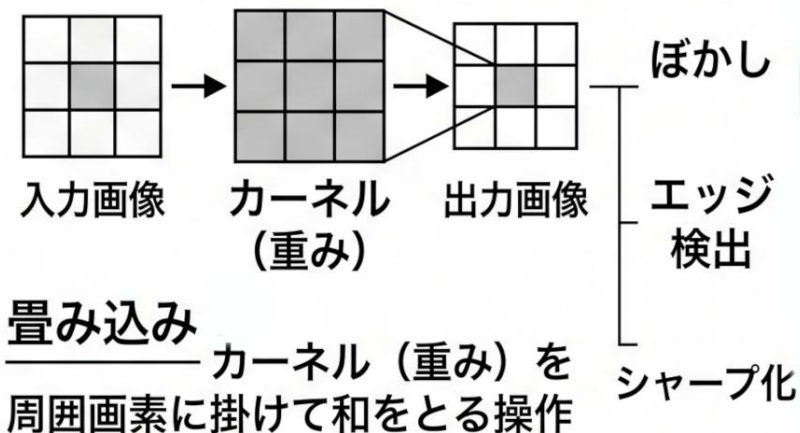
# 全体内容



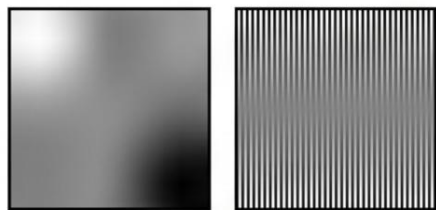
## 1. デジタル画像の構成



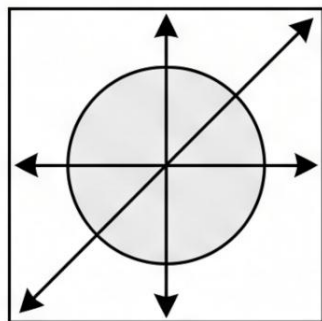
## 2. 畳み込み (フィルタ)



## 3. 空間周波数とスペクトル

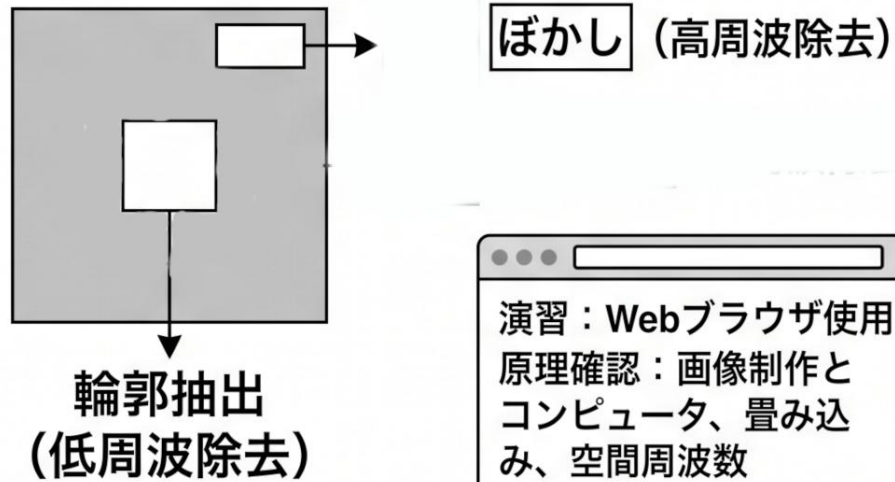


空間周波数：  
明暗変化の細かさ



中心：低周波  
周辺：高周波

## 4. 周波数フィルタ

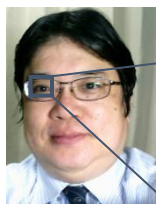


# 3-1 デジタル画像

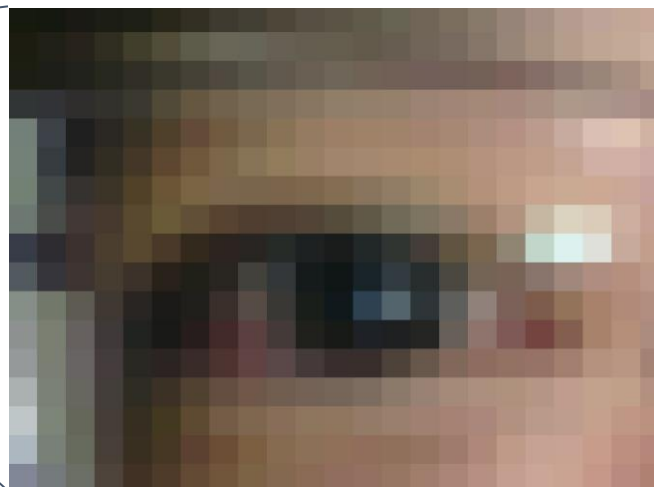
# デジタル画像の構造：画像と画素



- **画像は格子状に並んだ画素**（画像を構成する最小の要素）から構成される。
- 画像を拡大すると，個々の画素が確認できる。



画像



それぞれの格子が画素

# 画像の種類



カラー画像

**輝度（明るさの度合い）  
と色**の情報を持つ画像







濃淡画像

**輝度**のみの情報を持つ画像

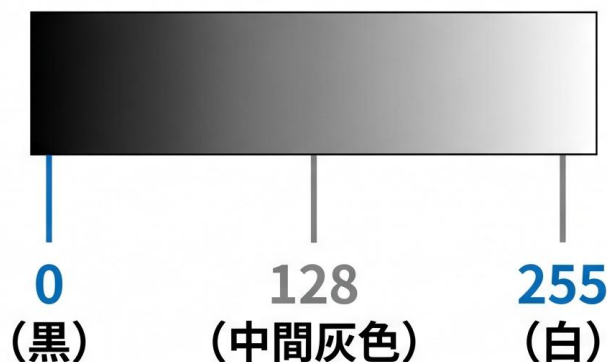
# 濃淡画像の基礎



## 1. 画像の輝度を数値化する (4段階の例)

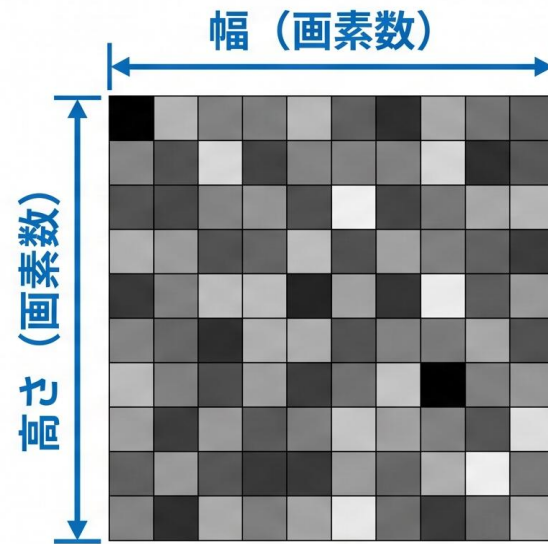
濃淡見本	対応する数値 (コード)
	0
	1
	2
	3

## 2. 一般的な256階調 (0~255) の表現



デジタル画像では通常、濃淡を0~255の数値で表す。

## 3. 画像のサイズ (画素数)



画像は『画素 (ピクセル)』の集まりであり、サイズは幅と高さの画素数で表される。

# カラー画像の成分

カラー画像の成分の表し方は2種類ある。

- R (赤) 成分, G (緑) 成分, B (青) 成分で考える場合

各成分は、画素ごとに**1つの数値**を持つ。すべて合わせて、画素ごとに**3つの数値**となる。



R (赤) 成分



G (緑) 成分



B (青) 成分

- 輝度成分, 色成分で考える場合

輝度成分は明るさの情報, 色成分は色の情報を表す。



輝度成分

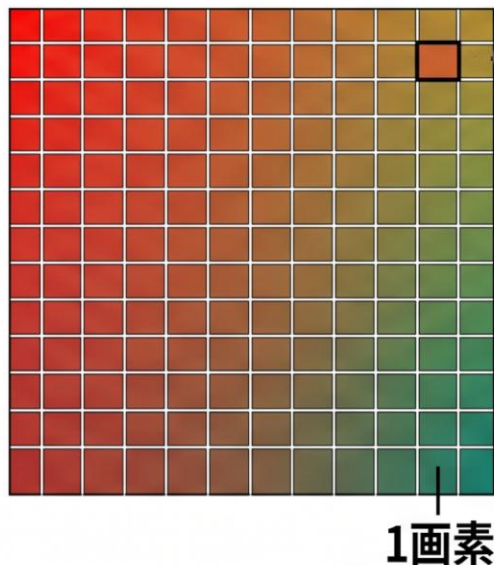


色成分

# カラー画像の基礎

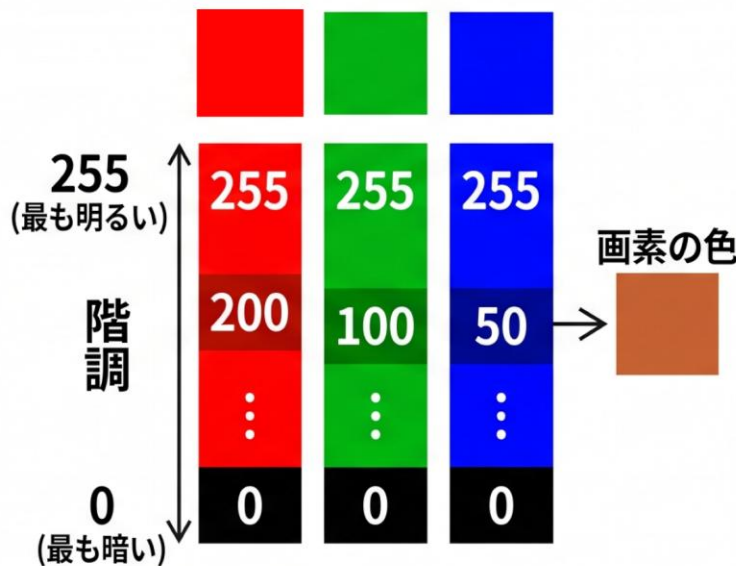


## 画像全体の構成



画像は格子状に並んだ画素（ピクセル）の集まり

## 画素の表現 (R, G, B)



光の三原色 (R, G, B) で表す各成分 0~255 の整数 (256 階調)

	(0, 0, 0)	黒
	(255, 255, 255)	白
	(200, 100, 50)	
	(255, 0, 0)	赤
	(0, 255, 0)	緑
	(0, 0, 255)	青

## 3-2 画像処理入門

# 画像処理

画像処理は、コンピュータで画像を加工・分析する技術

(例)

畳み込み (フィルタ処理)

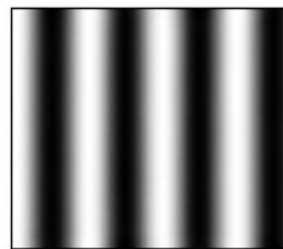
空間周波数



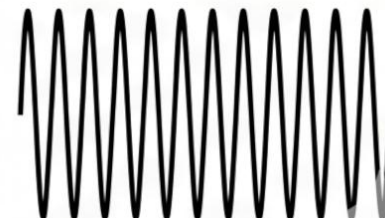
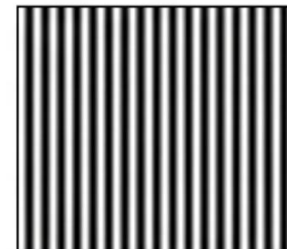
ノイズ除去



エッジ抽出



低周波



高周波

# 畳み込み (フィルタ処理)



- ある画素の値を, その周囲の画素の値を使って計算することで, 画像全体に効果を与える.
- 周囲の画素をどのような重みで使うかを決めた小さな表 (フィルタ, カーネルと呼ばれる) を用いる.

入力画像

2	4	1	0	3
4	4	1	2	3
1	2	3	0	0
0	4	1	3	2
0	0	1	2	3

カーネル (Kernel)  
小さな重みの表

0	-1	0
-1	5	-1
0	-1	0

$$\begin{aligned} & 4 \times 0 + 1 \times -1 + 2 \times 0 + \\ & 2 \times -1 + 3 \times 5 + 0 \times -1 + \\ & 4 \times 0 + 1 \times -1 + 3 \times 0 \\ & \text{結果 } 11 \end{aligned}$$

出力画像

		11		

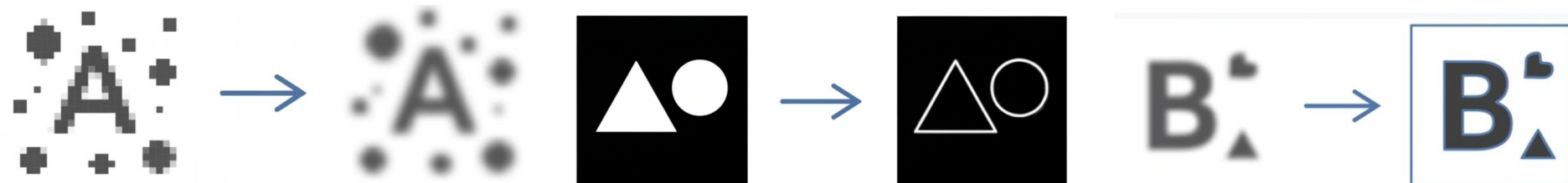
# 畳み込み（フィルタ処理）の種類と効果



**ぼかし**：周囲の画素の平均をとることで、画像を滑らかにする

**エッジ検出**：明るさが急に変わる部分（輪郭線など）を見つけて出す

**シャープ化**：輪郭を強調して、画像をくっきりさせる



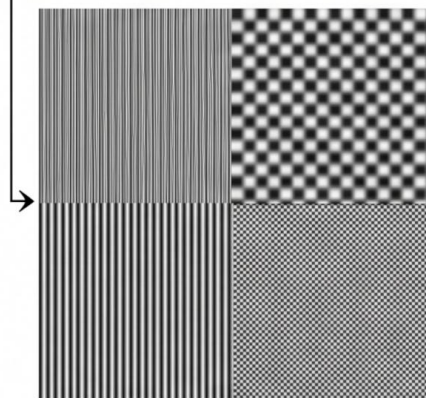
畳み込みの考え方は、画像処理だけでなく、AI（畳み込みニューラルネットワーク：CNN）にも応用されており、画像認識の基盤技術となっている。

# 空間周波数

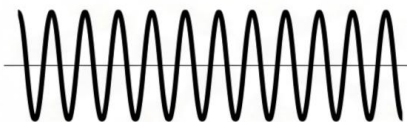
- 画像の中には、**明暗の変化が細かい部分**と**緩やかな部分**がある。
- 空間周波数は、明暗の変化の細かさに対応

## ○ 空間周波数が高い部分

明暗が細かく変化する部分

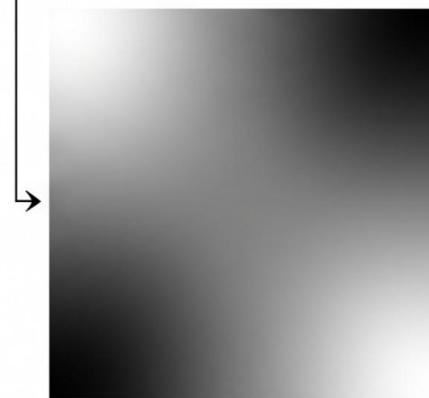


輪郭線  
テクスチャ（表面の模様や質感）の細部など

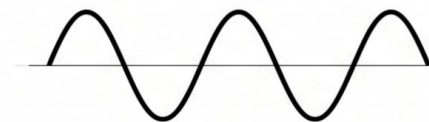


## ○ 空間周波数が低い部分

明暗が緩やかに変化する部分



空や壁のような広い領域など



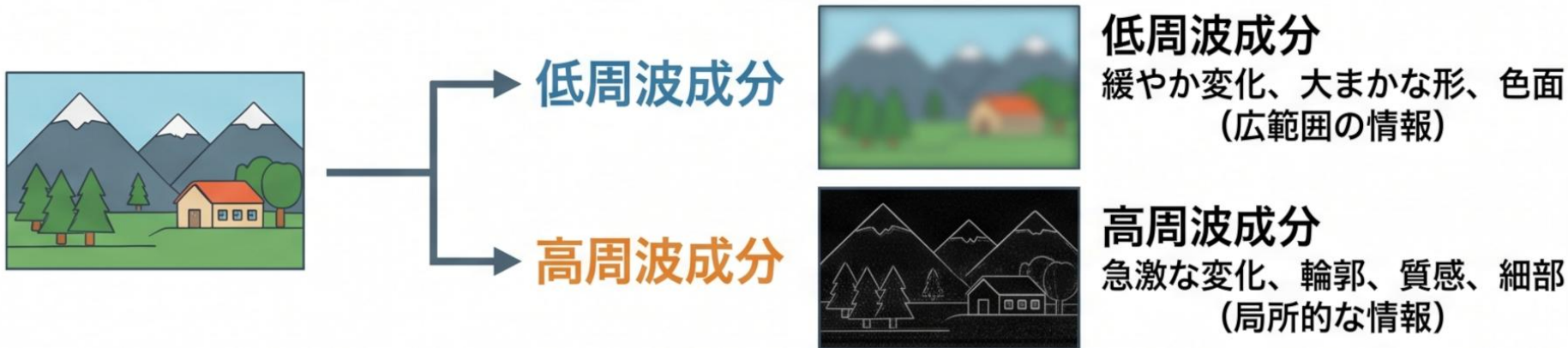
# 空間周波数



画像を空間周波数の観点で分析することで、**画像のぼかしや圧縮（人の目に見えにくい周波数成分の削減）**が可能になる。

高周波成分の除去はぼかし、低周波成分の除去は輪郭抽出に対応する。

## 基本概念：画像の空間周波数分解



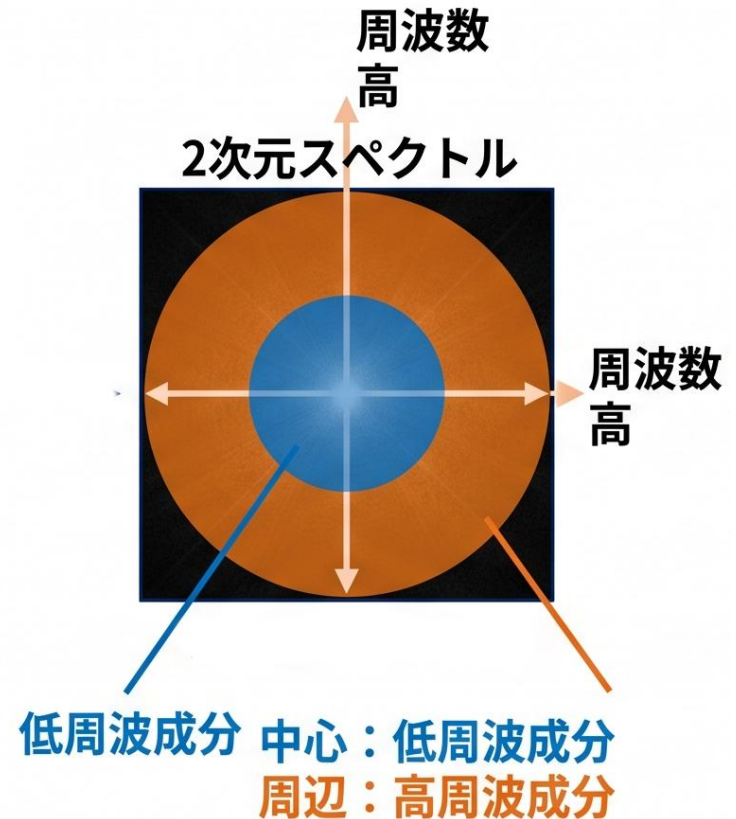
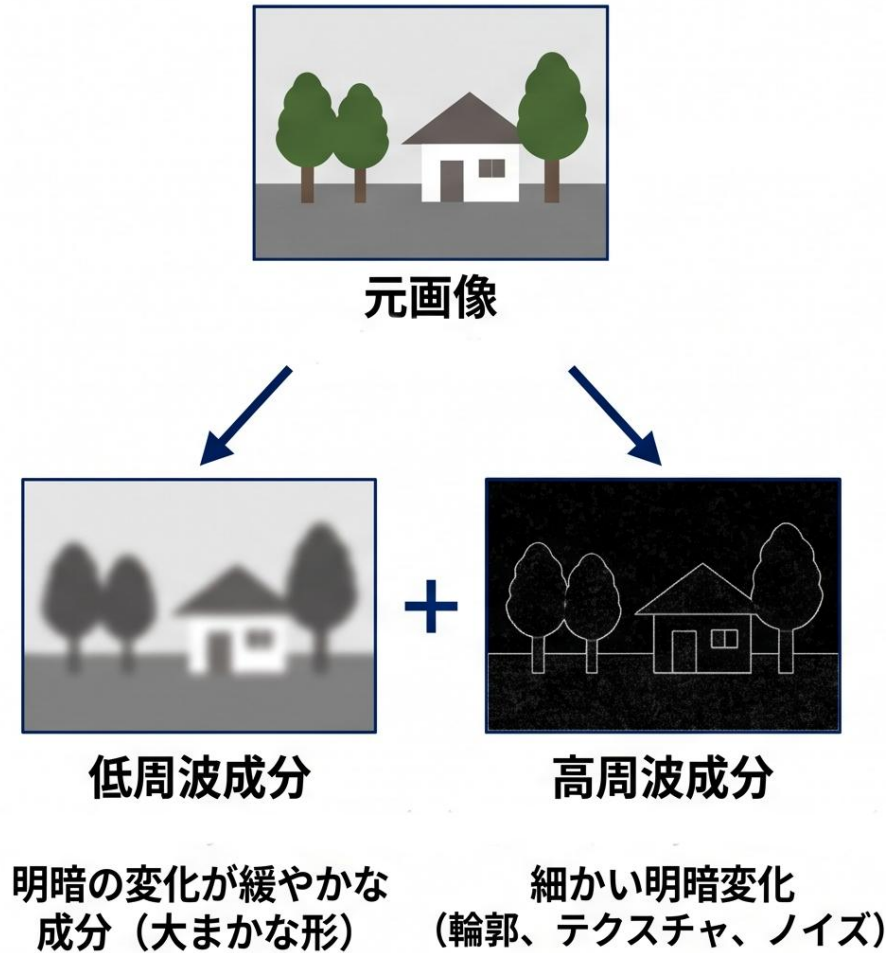
画像のぼかし  
(高い周波数の除去)



ぼかした画像

## 画像の構成：周波数成分の加算

## 2次元空間周波数スペクトル



画像のフーリエ変換により得られ、中心からの距離で周波数を表す。分布図。

# 演習

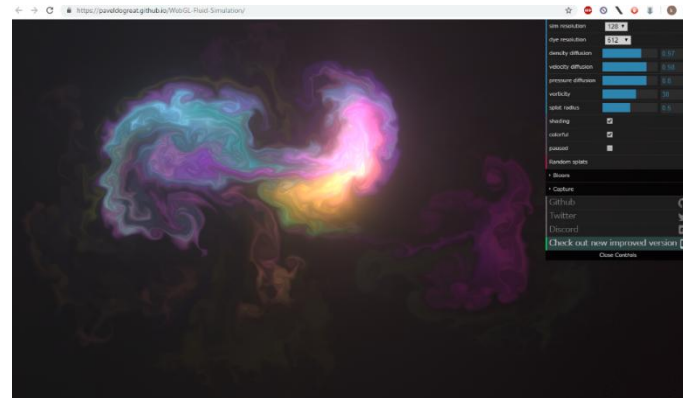
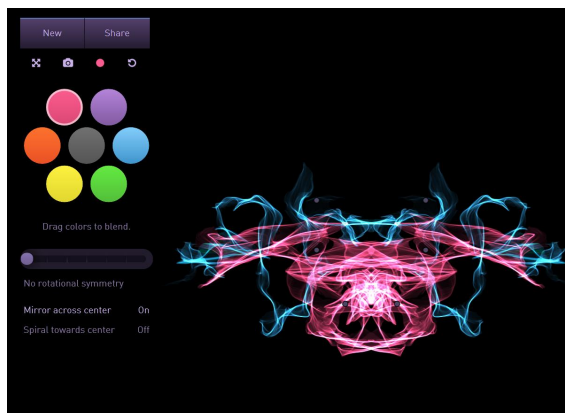
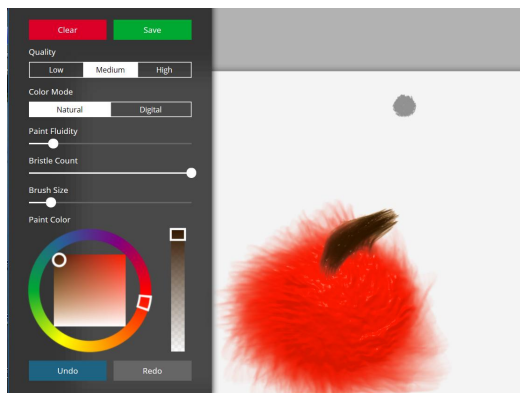


# 画像制作とコンピュータ（演習①～③）



パソコンのWebブラウザを使って画像制作を体験する。（これらの作者に感謝する）

- Fluid Paint: <https://david.li/paint/>
- Silk: <http://weavesilk.com/>
- WebGL Fluid Simulation: <https://paveldogreat.github.io/WebGL-Fluid-Simulation/>



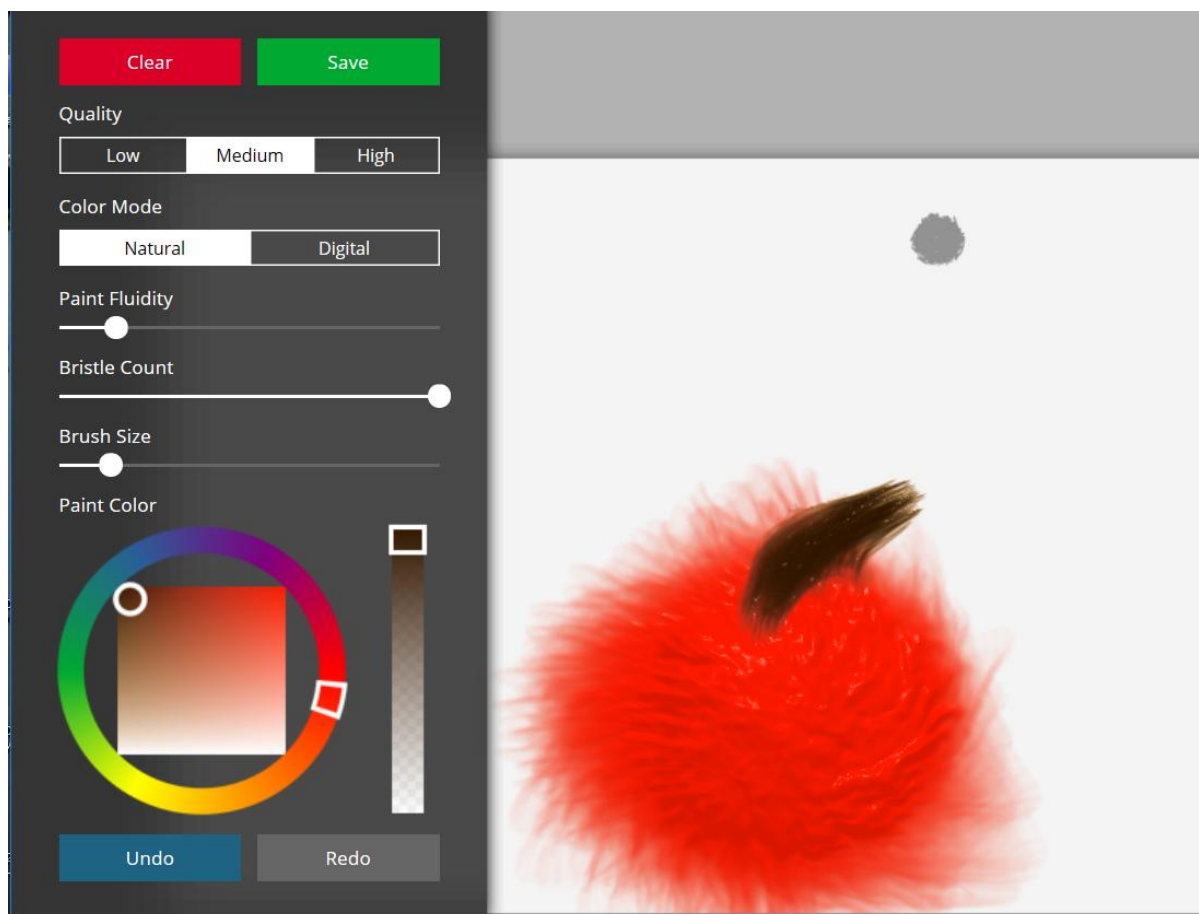
# 演習① ペイントソフト Fluid Paint



筆の各種設定が可能である。

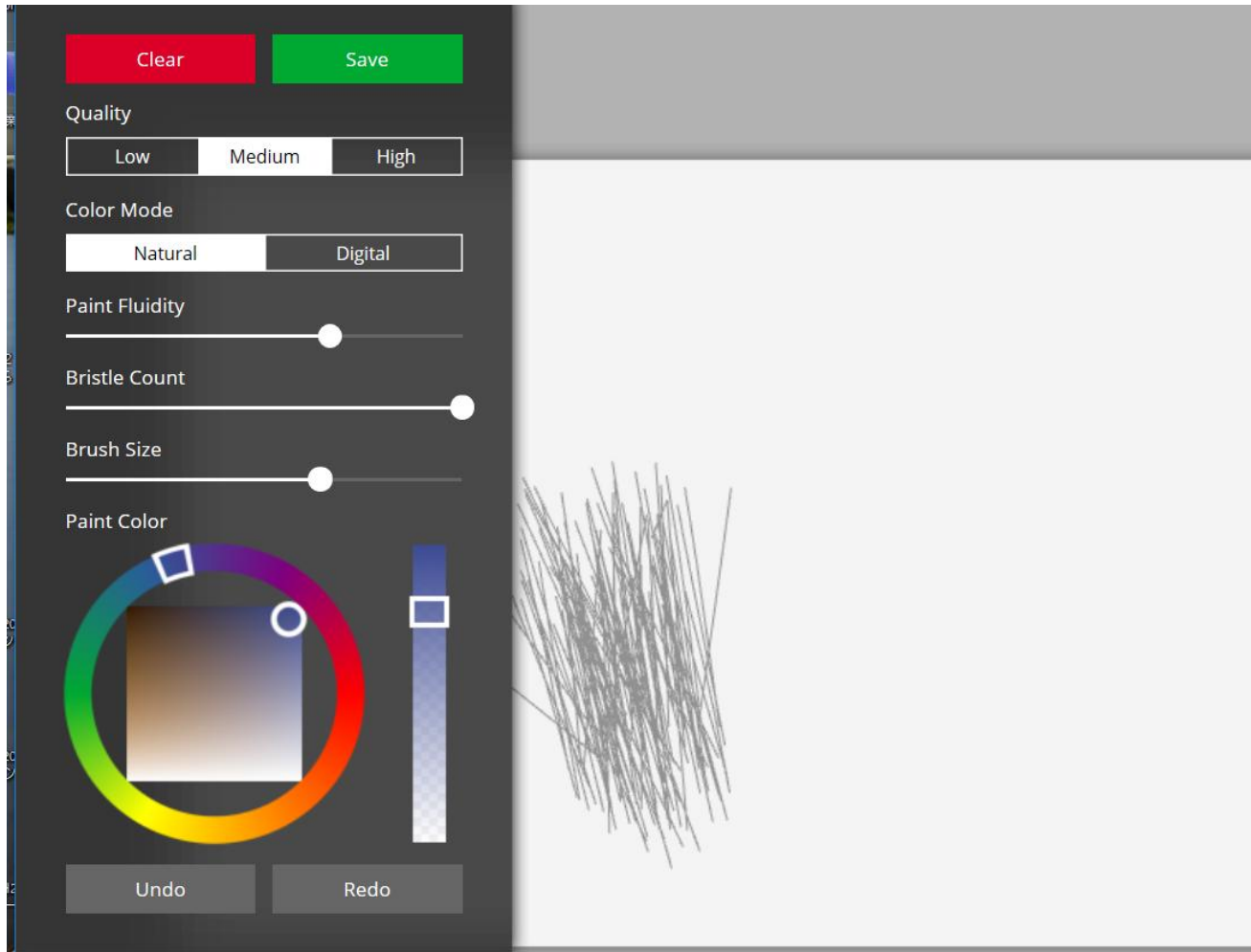
設定可能な項目：色，サイズ，毛先の多さ，毛先の流動性など。

<https://david.li/paint/>



# Fluid Paint の操作

筆の設定は左側のメニューで行う。書き直すときは「Clear」をクリックする。

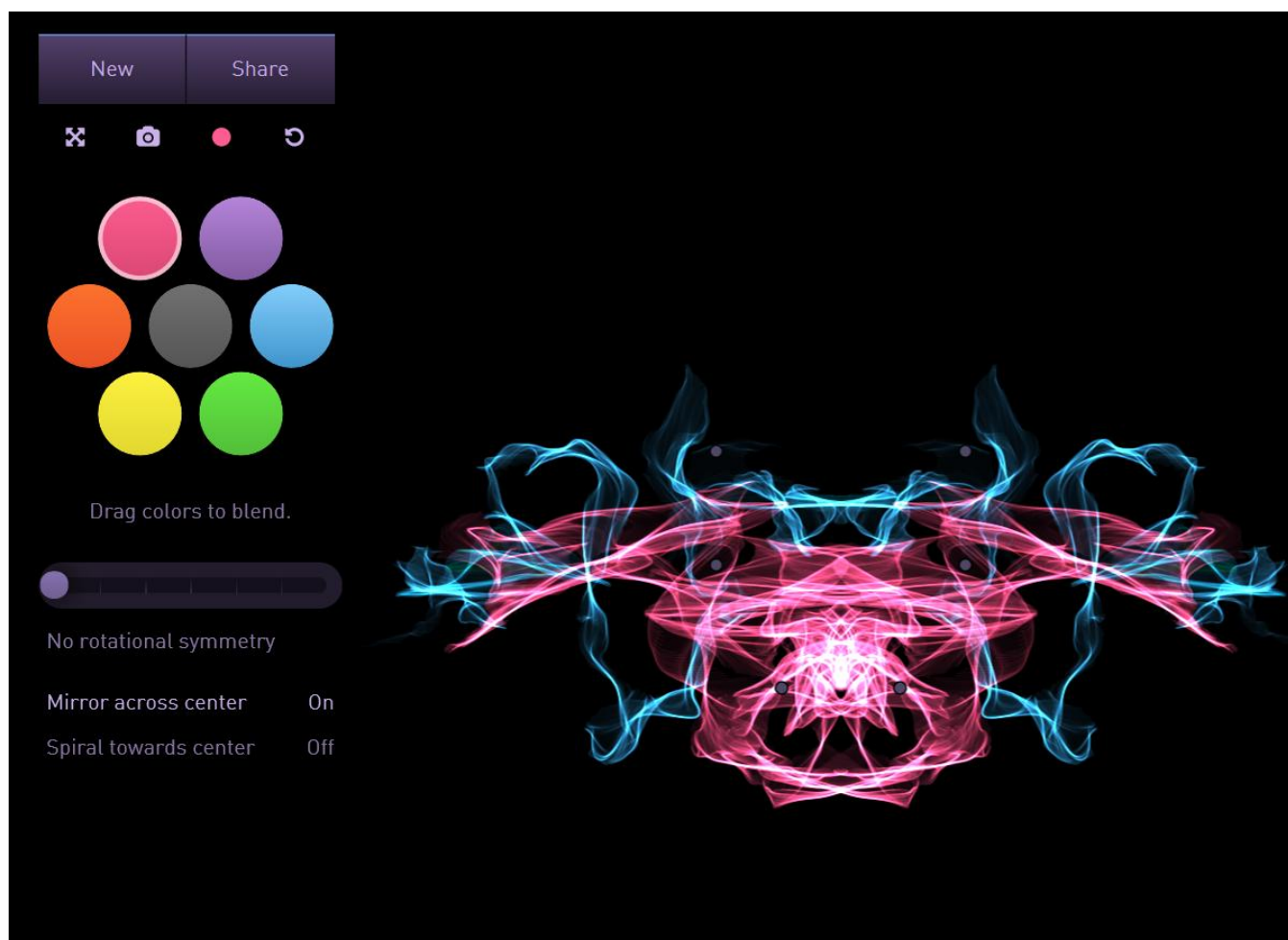


# 演習② アート作品制作サイト Silk



マウス操作でアート作品を作成できるサイト

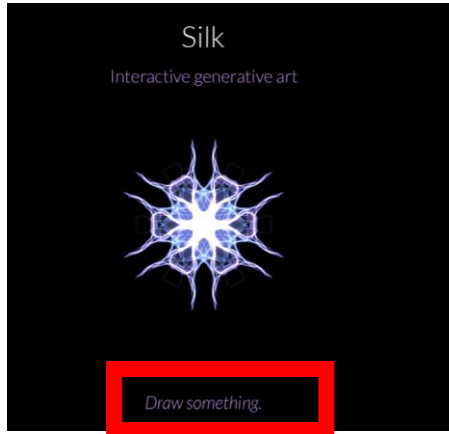
<http://weavesilk.com/>



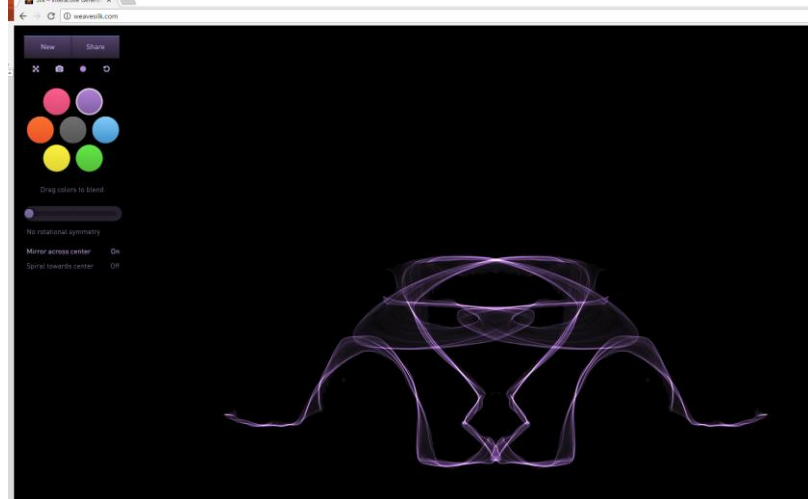
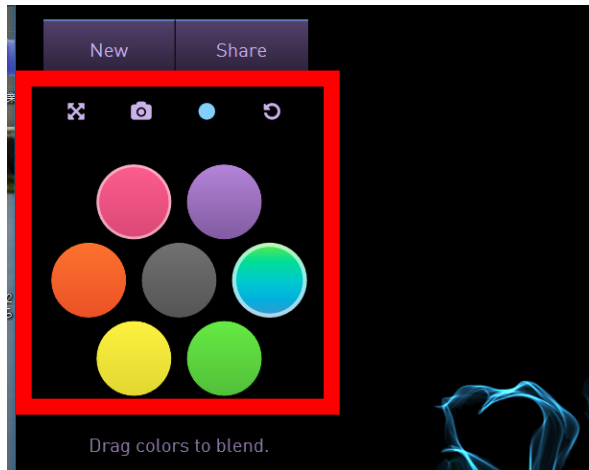
# Silk の操作



開始方法： Silkのページを開いたら、「**Draw Something**」をクリックして開始する。



色の選択： 左上のメニューで、色を選ぶことができる。

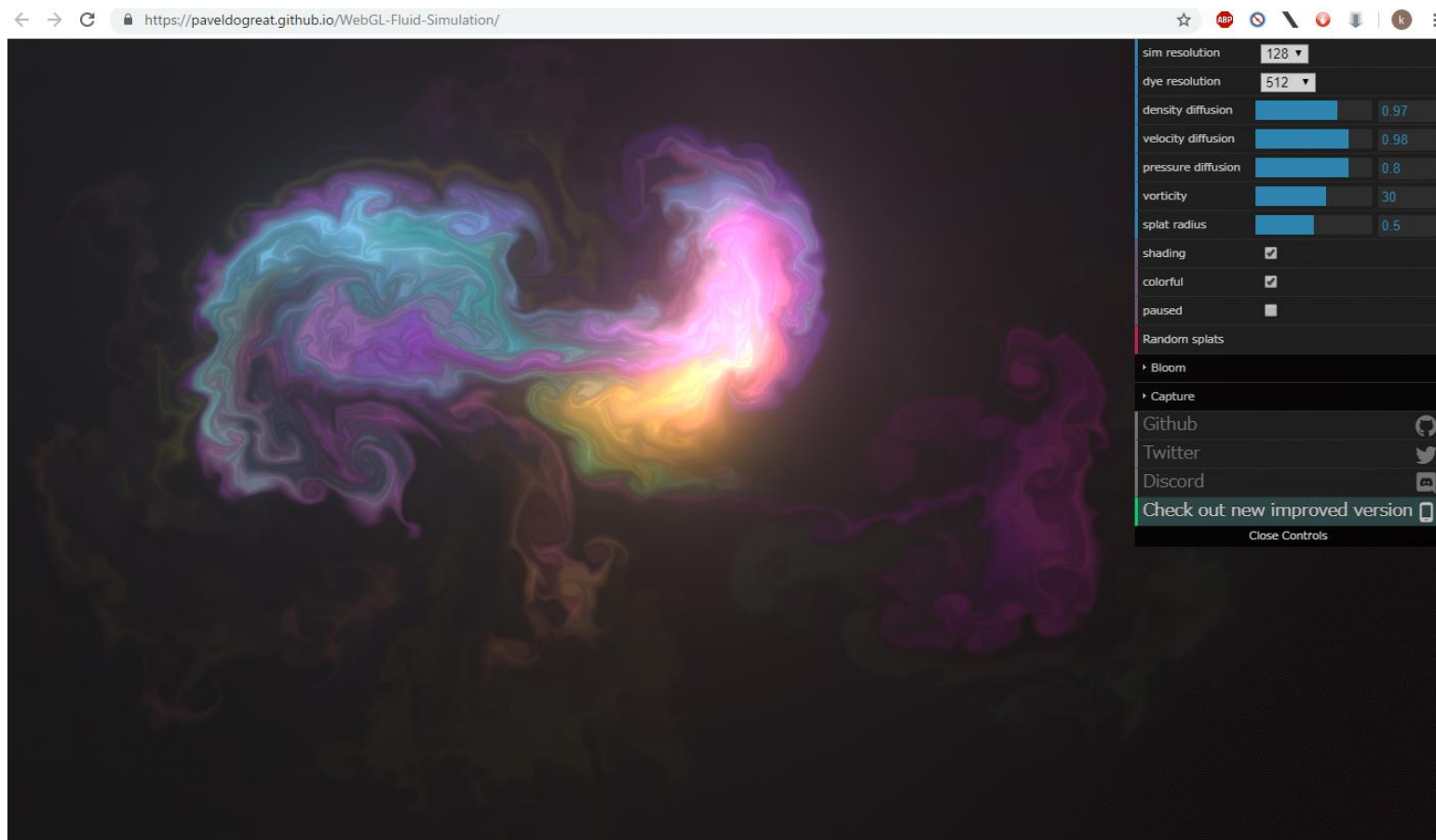


# 演習③ 流体シミュレーション WebGL Fluid Simulation



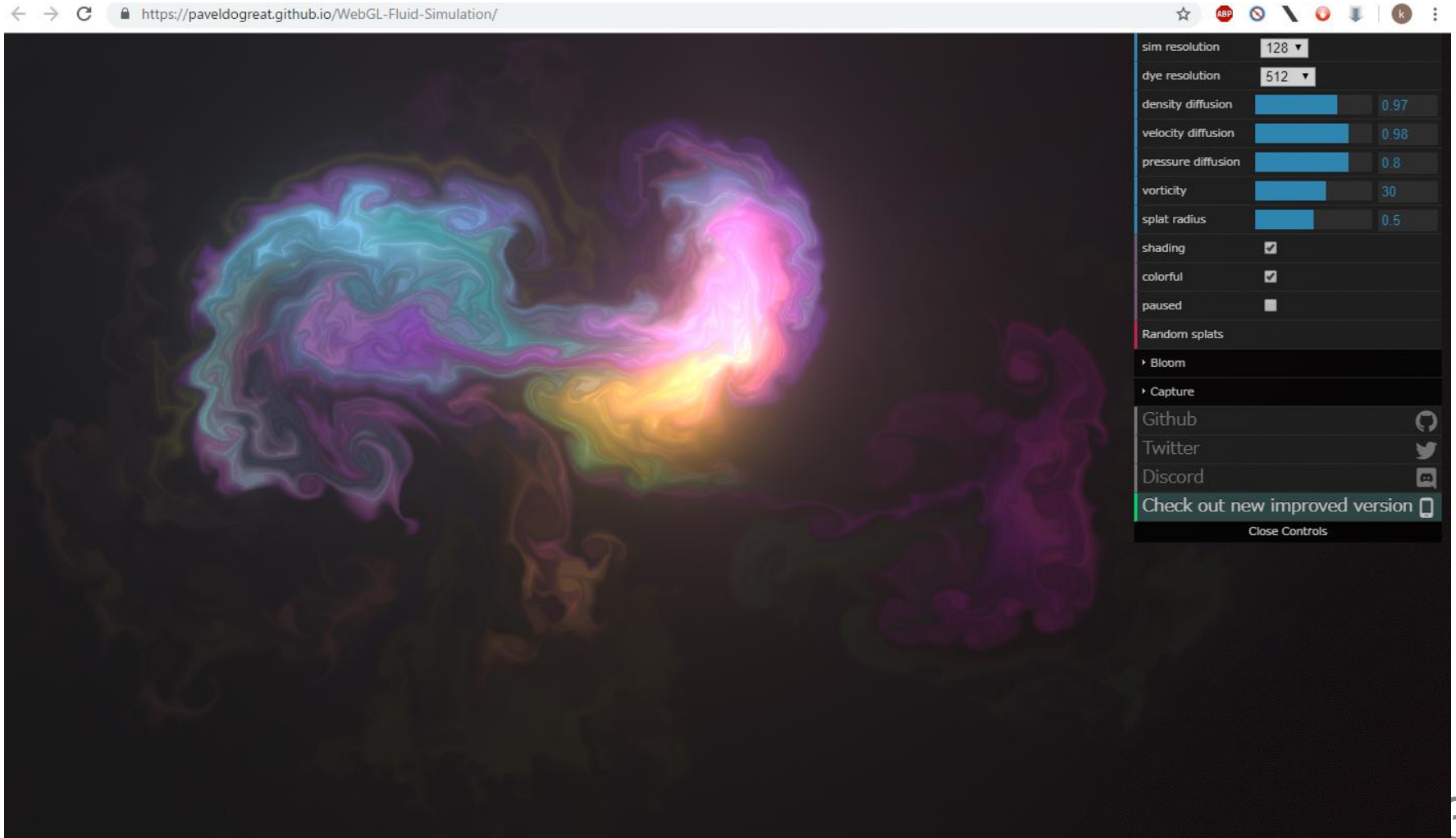
- 光の渦などを作成できるサイトである。

<https://paveldogreat.github.io/WebGL-Fluid-Simulation/>



# WebGL Fluid Simulation の操作

WebGL Fluid Simulationのページを開いたら、マウス操作で描く。

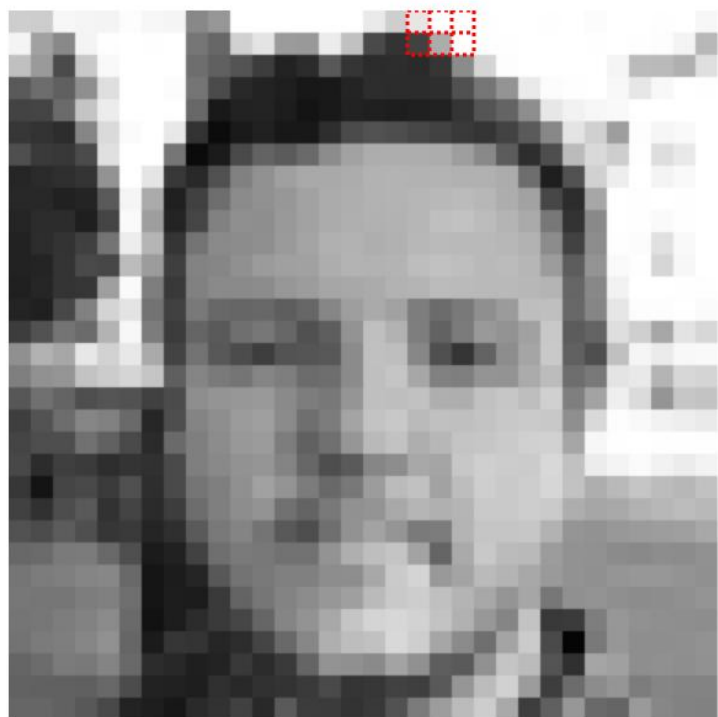


# 演習④ Setosa.io : 画像にフィルタを適用



フィルタの適用によって画像が変化することを、操作を通じて確認

<https://setosa.io/ev/image-kernels/>

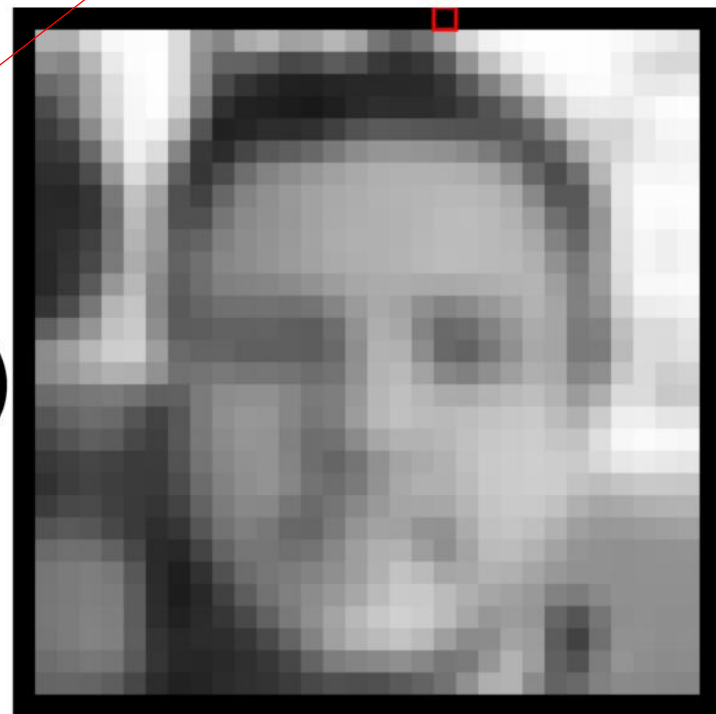


input image

$$\left( \begin{array}{l} \begin{array}{ccc} ? & + & ? & + & ? \\ \times 0.0625 & \times 0.125 & \times 0.0625 \end{array} \\ + \\ \begin{array}{ccc} 231 & + & 255 & + & 254 \\ \times 0.125 & \times 0.25 & \times 0.125 \end{array} \\ + \\ \begin{array}{ccc} 66 & + & 173 & + & 255 \\ \times 0.0625 & \times 0.125 & \times 0.0625 \end{array} \end{array} \right) = \begin{array}{c} ? \\ \text{kernel:} \\ \text{blur} \end{array}$$

フィルタの種類を選ぶメニュー

畳み込みの式



output image

## 【手順】

- 顔画像とその画素値（0～255）が表示されている
- ドロップダウンメニューで**フィルタを選ぶ**.  
**ぼかし（blur）, 鮮鋭化（sharpen）, エッジ検出（outline, sobel）**など.
- フィルタを切り替えて出力画像の変化を確認する.
- **出力画像の任意の画素にマウスを合わせる**. その画素値が、**入力画像の3×3近傍の画素値**による**畳み込み（フィルタ処理）**で算出される過程が表示される.

## 【学びの内容】

- 画像は画素値を並べた2次元配列.
- 畳み込みは、画像上の各位置で近傍の画素値とフィルタの値の要素ごとの積の和を計算する操作

# 演習⑤ deeplizard : 畳み込みを1ステップずつ確認する



## deeplizard : 畳み込みを1ステップずつ確認する

<https://deeplizard.com/resource/pavq7noze2>

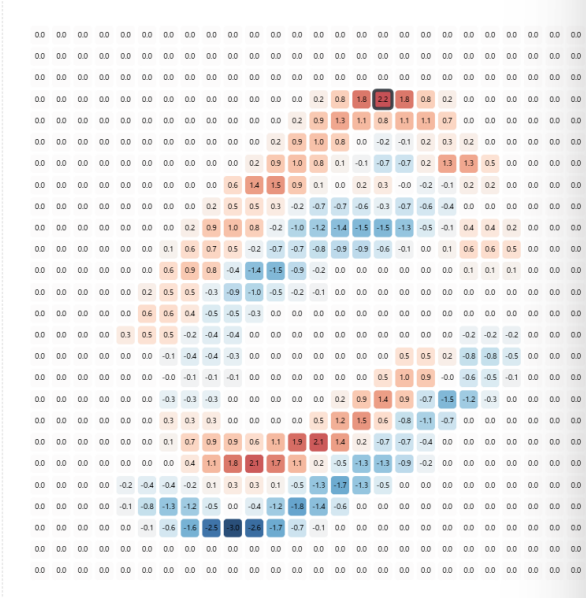
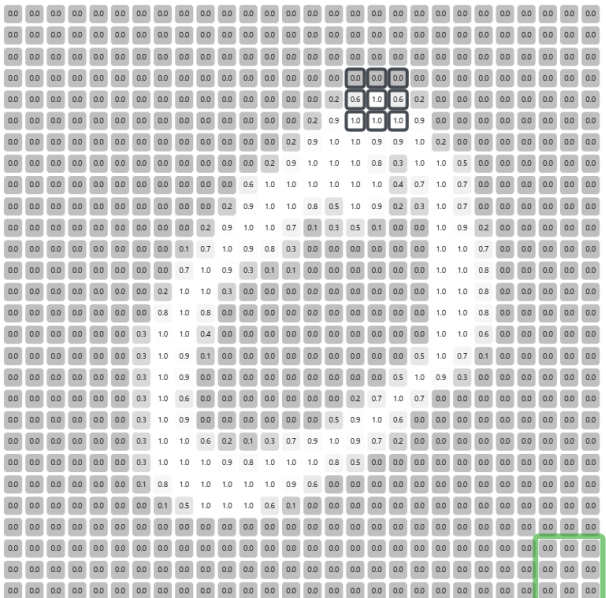
Full Screen | MNIST | 0 - Zero

Top Edge Filter | Step | Play

Window:  $\begin{bmatrix} 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 \end{bmatrix} * \begin{bmatrix} -1.0 & -1.0 & -1.0 \\ 1.0 & 1.0 & 1.0 \\ 0.0 & 0.0 & 0.0 \end{bmatrix} = \begin{bmatrix} 0.0 \end{bmatrix}$

Input:  $\begin{bmatrix} 0.0 & -1.0 \\ 0.0 & -1.0 \\ 0.0 & -1.0 \\ 0.0 & 1.0 \\ 0.0 & 1.0 \\ 0.0 & 1.0 \\ 0.0 & 0.0 \\ 0.0 & 0.0 \\ 0.0 & 0.0 \end{bmatrix} + \dots + \begin{bmatrix} 0.0 & 0.0 \\ 0.0 & 0.0 \\ 0.0 & 0.0 \end{bmatrix} = \begin{bmatrix} 0.0 \end{bmatrix}$

Output



## 【手順】

- **入力，畳み込みのフィルタ（ $3 \times 3$ ），出力を確認。**
- 「**Play**」をクリック。フィルタが入力行列上を1マスずつスライドする過程を確認。
- 出力行列の画素にマウスを合わせると，対応する入力の $3 \times 3$ 領域とフィルタの要素ごとの積の和が表示される。
- **出力値の正は赤，負は青で色分けされる。フィルタが大きな出力を生む位置と入力中のパターンとの対応を確認する。**

## 【学びの内容】

- 出力値が正に大きい位置は，**入力のパターンとフィルタのパターンが一致する位置**に対応する。
- これはAIの畳み込みニューラルネットワーク（CNN）が画像から特徴を抽出する原理

# 演習⑥ 2D Inverse Fourier Transform Playground : 空間周波数を操作



スペクトル上の位置と画像の空間周波数成分との対応を、スペクトルを直接編集して確認

<https://monman53.github.io/2dffft/>

## Original image

Original image or the result of backward Fourier transform of forward Fourier transform (spectrum).

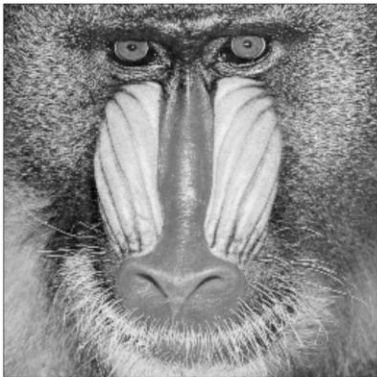
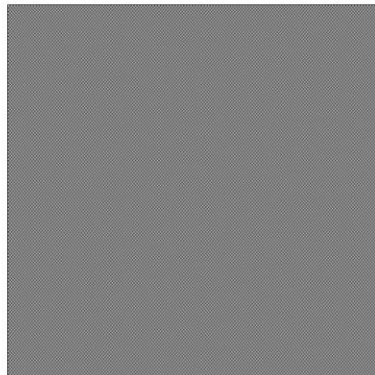


Image URL:   
or  選択されていません



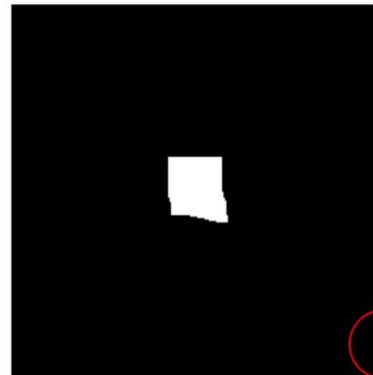
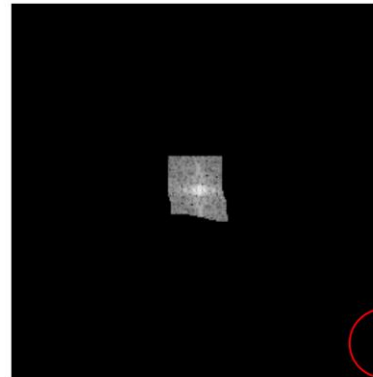
## Result

The result of backward (inverse) Fourier transform of masked spectrum.



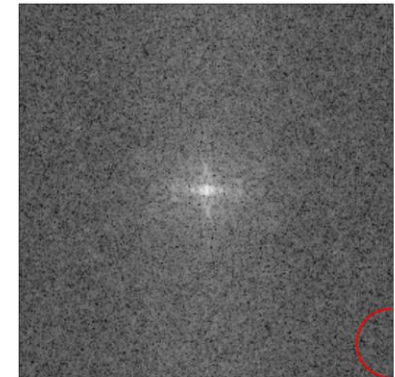
## Masked spectrum (edit here!)

- **Left-drag**: Scratch mask
- **Right-drag**: Restore mask
- **Wheel**: Change circle size

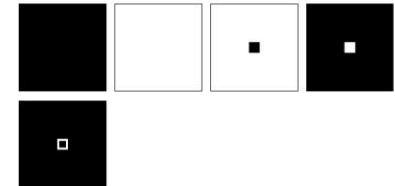


## Spectrum

The result of forward Fourier transform of original image. The actual values are complex numbers, but [absolute values](#) are used here for visualization.



Mask URL:   
or  選択されていません





## 【手順】

- **原画像，マスク適用後の逆フーリエ変換（周波数成分から画像を再構成する変換）結果，編集可能なマスク付きスペクトル，**スペクトルが並んで表示される。
- **右下で「白い四角」を選ぶ。**これは**全通過**（すべての周波数成分を通すため，原画像がそのまま再現される）。Result のところに、現画像が表示されるようになる。
- **スペクトルの周辺部（高周波成分）を左ドラッグで削る。**出力画像が平滑化され，輪郭や細部が失われることを確認する。
- **スペクトルの中心部（低周波成分）を左ドラッグで削る。**画像の大局的な明暗が失われ，輪郭成分が残ることを確認する。

## 【学びの内容】

- **スペクトル上では，中心が低周波成分，周辺が高周波成分**