

# コンピュータビジョン： パターン認識

富岡雄磨

# 本章の目的

- mahotasというパッケージを用いて、画像処理を行う
- 画像のクラス分類を行う

# コンピュータビジョンと画像処理

- コンピュータビジョン

画像という配列データから、アプリケーションに関連する判断を行う

- 画像処理

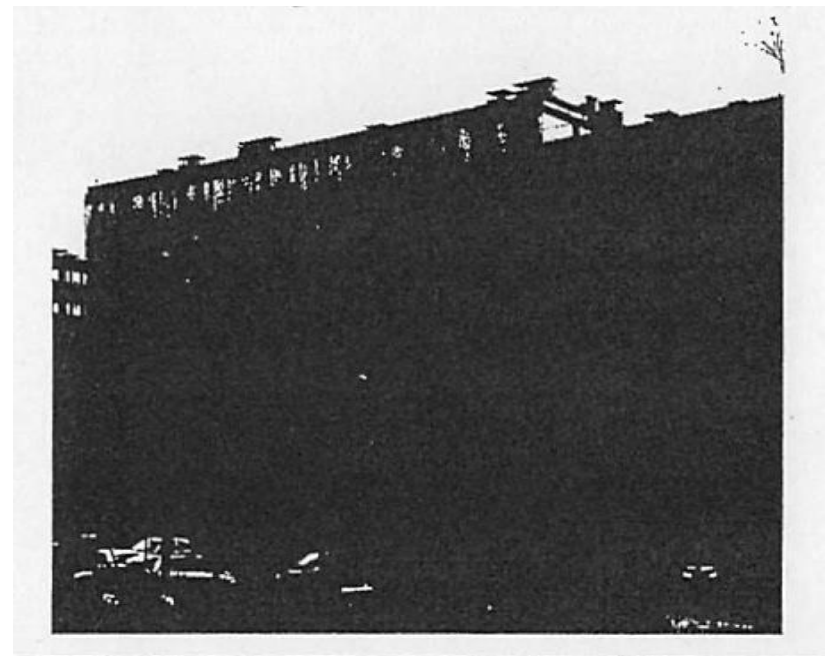
与えられた画像を処理し、異なる画像もしくはよりよい画像を生成する

# 画像処理：閾値処理

- 画像をグレースケール化し、その画像を元に閾値を計算、二値化処理を行う。



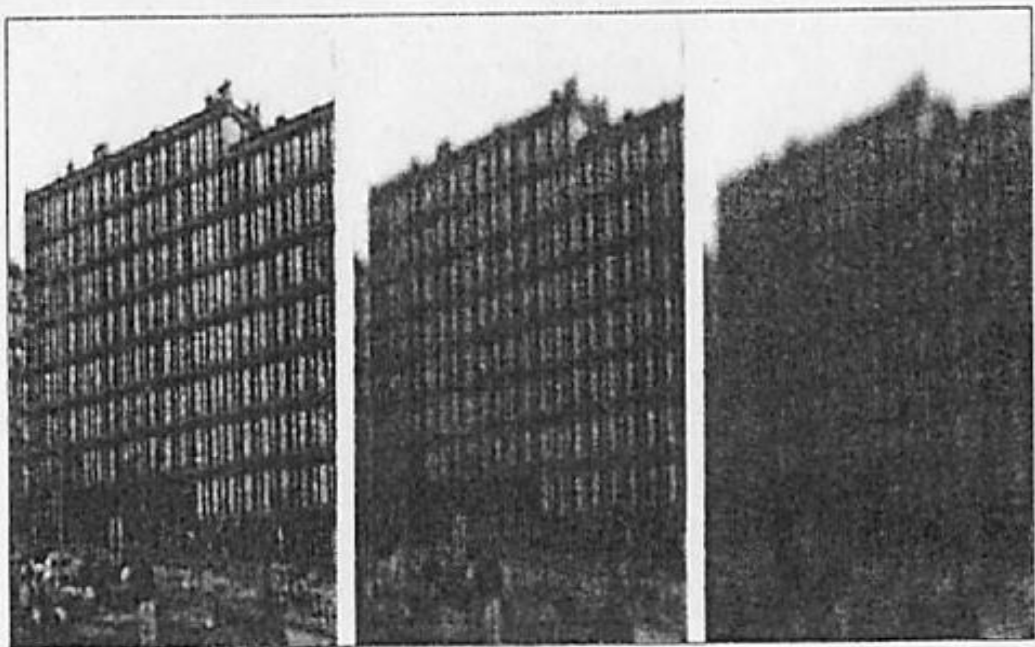
グレースケール化した画像



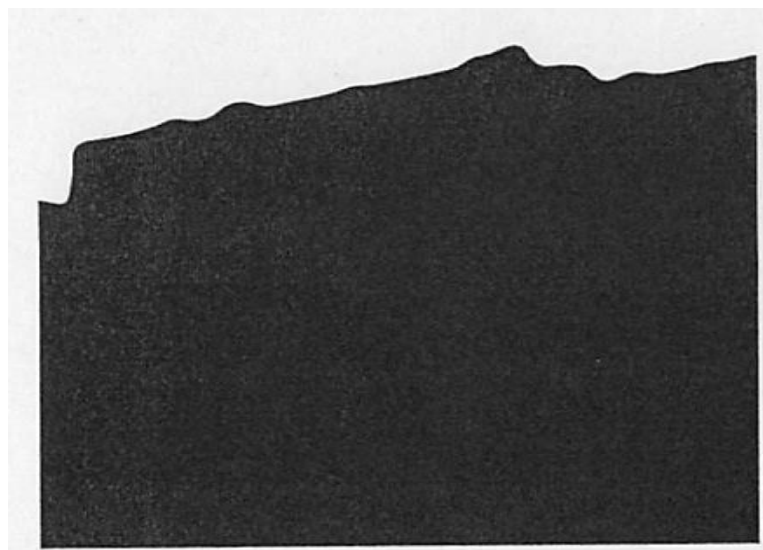
二値化処理をした画像

# 画像処理：ガウシアンぼかし

- 画像をぼかすことで、大きい視点で重要ではない細部を取り除くことができる



ぼかしの度合い小 → 大



ぼかした画像（大）を二値化

# 画像処理：ノイズを加える

- スキャナで読み込んだときのシュミレーションのために、ノイズをかけることもできる



元画像



ノイズをかけた画像

# 画像処理：フォーカスを当てる

- 画像フィルタを利用し、中心部から外部へ重みを小さくしていくことで、中心部は鮮明に、外部はソフトになる



元画像



中心にフォーカスを当てた画像

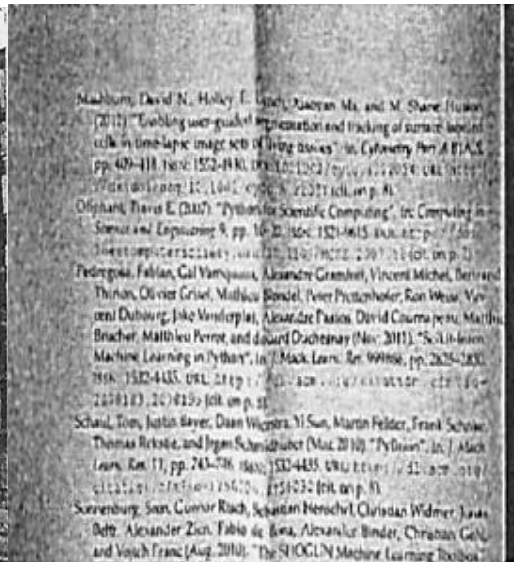
# パターン認識

- 画像におけるクラス分類のことを、パターン認識と呼ぶ
- 画像から特徴量を抽出し、それを用いて分類するのが定番のアプローチである

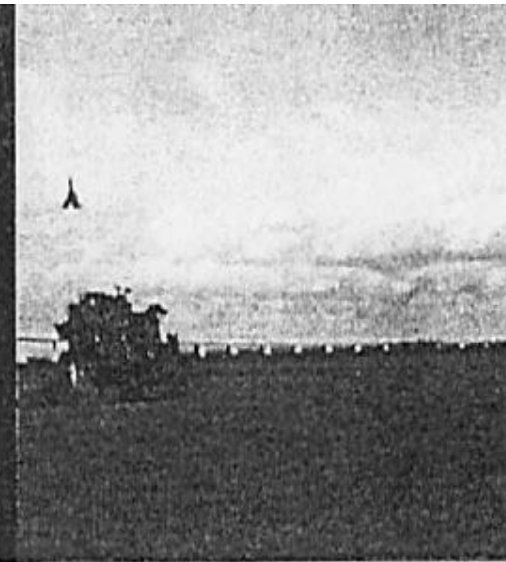
# パターン認識：特徴量の計算



建物



テキスト



風景

mahotasを用いると、以下の式で簡単に計算してくれる

```
haralick_features = np.mean(mh.features.haralick(image), 0)
```

# パターン認識：エッジ検出

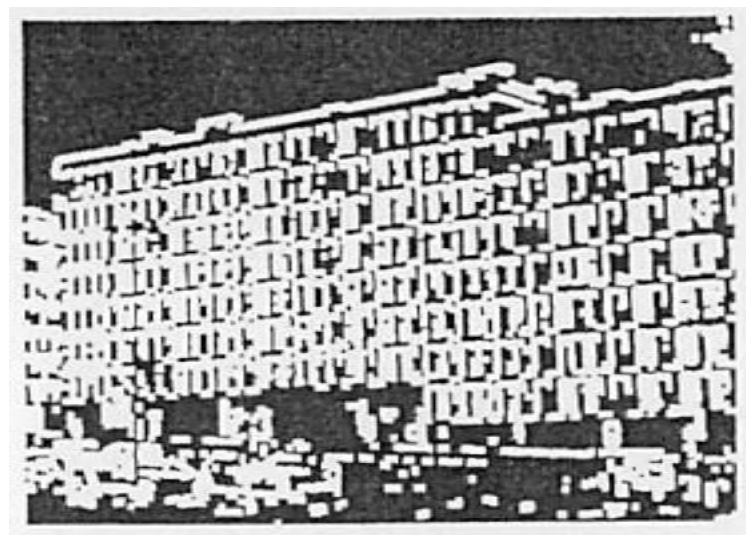
- ソーベルフィルタという手法が有名  
二つの行列で画像をフィルタ処理する手法

$$\begin{pmatrix} 1 & 0 & 1 \\ -2 & 0 & -2 \\ 1 & 0 & 1 \end{pmatrix}$$

垂直方向の処理

$$\begin{pmatrix} 1 & -2 & 1 \\ 0 & 0 & 0 \\ 1 & -2 & 1 \end{pmatrix}$$

水平方向の処理



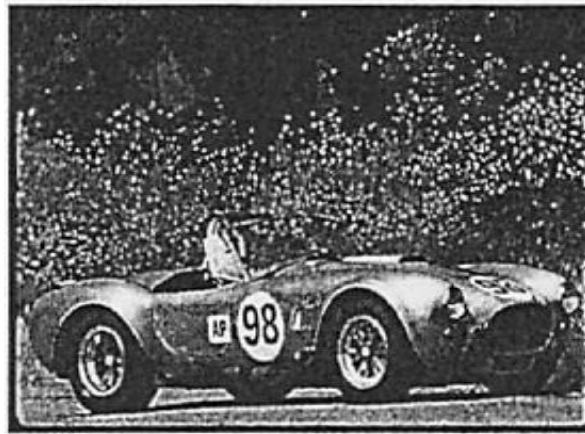
エッジ検出を行った画像

# パターン認識：局所特徴量の検出

- 画像の一部領域の特徴量を計算する



風景



車

