コンピュータビジョン

担当: 井尻 敬

○ 講義の概要:

画像処理は，産業・自然科学・エンタテインメントなど，多種多様な分野の発展に関わる非常に重要な技術です．2年生時のデジタルメディア処理では，画像処理の基本である『画像データ構造・画像撮影方法・各種フィルタ・拡大縮小・補間』などについて紹介しました．この内容をさらに発展させ，**本コンピュータビジョンでは，計算機が画像を認識する手法**について紹介します．具体的には，画像から目的部分を切り抜く領域分割，画像の特徴を計算機が理解できる形で記述する特徴抽出，および，抽出した特徴を用いて画像を識別するパターン認識について解説します．また，講義後半では，深層学習を用いた画像処理についても紹介します．

本講義にて解説する技術に関して，コーディング可能な深さで理解できるよう，ソースコードを交えながら詳細な技術解説を行ないます．また，Pythonを用いたプログラミング演習を通して画像処理手法のより深い理解を目指します．講義資料は井尻のweb page （takashiijiri.com）に事前にアップロードします．

○ 達成目標：

1. 領域分割 – 画像の領域分割法について主要なアルゴリズムを説明・実装できる
2. 特徴抽出 – 画像認識に必要な特徴抽出の基礎を説明・実装できる
3. パタレコ – 画像に対するパターン認識（顔認識など）の基礎やアルゴリズムを説明・実装できる

○ 成績評価：

毎回の小テスト（50%），プログラミング課題（50%）に基づき評価します．

○講義資料：

講義で紹介した資料・ソースコードは可能な限りWeb上に公開します．

https://takashiijiri.com/classes

https://github.com/Takashiljiri/PythonOpenCVPractice

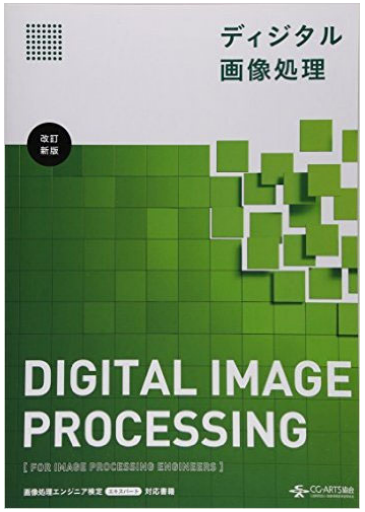
○場所：

豊洲キャンパス

- 座学：教室棟 405
- 演習：未定

○その他：

- あめ・がむ・飲み物などはご自由に。お手洗い許可不要。スマホ通話は教室の外で。
- 周囲に迷惑となる程度の私語は控えて下さい（小さな声で教え合うのはOK）（本学の私担当の講義で私語が問題になったことは無いですが、念の為）。
- 小テストの解答データの共有やコピー、プログラミング課題の解答データの共有やコピーペーストは、カンニングと同等の扱いとするので絶対に行わないでください。（わからない部分の教え合いは奨励します。）



参考書

- CG-Arts協会（画像情報教育進行委員会）
- ディジタル画像処理[改訂新版] 大型本

- 日本語で読める画像処理の教科書です
- 画像や例が多く入門者には最適だと思います

- 網羅性が非常に高い反面，初学者にとっては説明が少ない部分もある → 講義中に丁寧に解説します

※基本的に、スライド資料を中心に講義を進めます

小テストについて

- 1~8回目の講義内にて実施
- 実施方法：scombz
- 解答期間：**講義の最後10分 ~ 20分**
  ※ただし「初回」「講義が長引いた場合」「scombzのトラブル」
  「ややこしい計算問題がある回」は解答期間を延長します
- 内容：
  - その日に扱った手法に関する課題
  - 次回扱う手法に関するごく簡単な課題

- 禁止：他者との協力 / 生成AI
- OK：持ち込み・webの参照

ある手法を『理解する』とは？

- 教科書をおぼえた　　：×
- 人にその手法を説明できる　：△
- 例を挙げて人に説明できる　：○
- プログラムとして記述できる：◎

→ コードを書こう！

※井尻の偏見に基づきます．異論は認めます．
※きちんと理解できていない手法も離散化された数式さえあれば
　コードに落とせることも結構あります。。

ソースコードについて

- 本講義紹介する手法はなるべくソースコードも合わせて提供します
  - takashiijiri.com/classes
  - github.com/TakashiIjiri/PythonOpenCVPractice
- Python & OpenCV または MFC & C++ 環境で書いてあります
- インストール方法・コーディングの基本に関する資料も用意します
  - ただし詳細は講義中には触れません
  - 興味のある人だけ自由に勉強を進めてください

Contents

復習: ディジタルメディア処理

本講義はディジタルメディア処理（2年後期）の知識を前提とする
- 画像とは
- 画像の変形とアファイン変換
- フーリエ変換 (FFT)
- フィルタ処理
- 畳み込み
- 逆畳み込み

※未履修の方は適宜独習してください
※過去資料は, http://takashiijiri.com/classes/index.html

講義を理解するために前提とする数学知識

- 高校までの数学
- 線形代数全般 : ベクトル，ベクトルの内積，行列，行列の積，固有値固有ベクトル
- 最適化数学（主に最急降下法を利用）
- 畳み込みとフーリエ変換

復習をお願いします.

予習 2週目 : テンプレートマッチング

- 入力画像から物体を検出するための手法
- 検出対象の画像（テンプレート）を用意し, 入力画像をラスタスキャンし相違度を評価
- 相違度が閾値以下の領域を出力する
- 相違(類似)度 : SAD, SSD, NCCなど

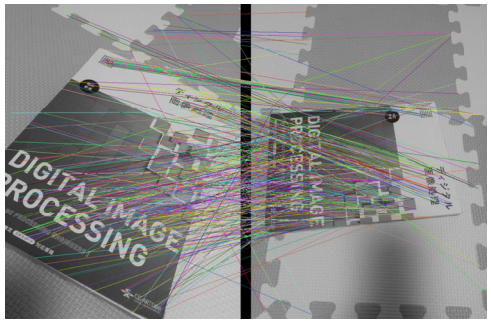
入力画像

テンプレート

サブピクセル精度で検出するための関数フィッティング
高速化のための残差逐次検定・粗密(coarse to fine)探索・chamfer matching

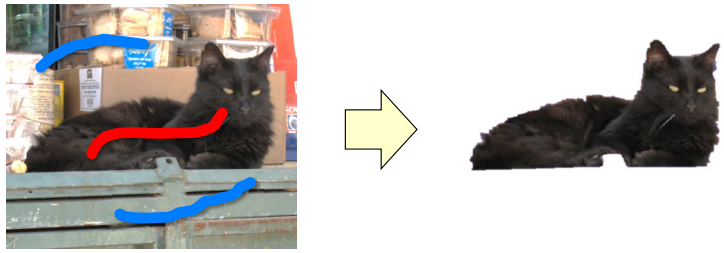
予習 3週目 : 特徴ベクトルとSIFT

- 特徴ベクトルとは何か
  - 検出された特徴点同士を比較するため，特徴点周囲の局所領域をベクトルの形で表すもの.
  - 特徴ベクトルは，SIFT，BRIEF, ORB, SURF, AKAZEなど，沢山の種類がある
  - 特徴ベクトルは目的や対象画像の依存してよいものを選択すべき
- SIFT特徴
  - DoGの極値を特徴点として検出
  - 特徴点のスケールに応じた局所領域を考慮
  - 特徴点周囲の勾配方向に沿って局所窓を回転
  - 局所窓を4分割し，各領域の勾配ヒストグラムを特徴ベクトルとする



予習 4週目：領域分割 (下図はグラフカット法)

前傾制約 背景制約



前景・背景に属する画素を適当に入力すると，これを制約に画像を二値化

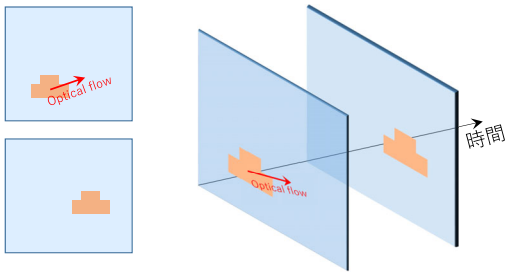
二値化をエネルギー最小化問題として定式化し、フローネットワークの最小カットにより最適解を計算する

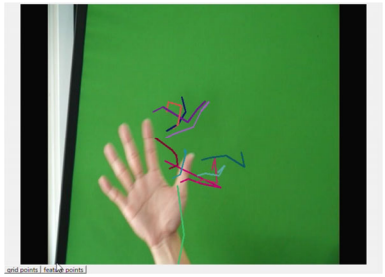
臓器などの塊状領域に対してはかなり強力な領域分割法

OpticalFlowvis.py

予習 5週目：オプティカルフロー

- 2枚の画像内におけるある物体の移動量をベクトルとして表現したもの
- 動画内の『物体追跡』や『物体の動きの解析』などに利用される
- ブロックマッチング法，Lucas-Kanade法など

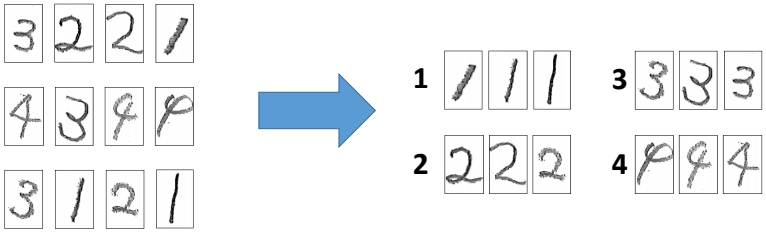




予習 6週目：パターン認識

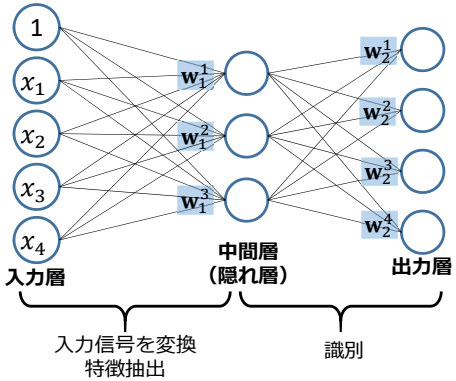
『データの中の規則性を自動的に見つけ出し、その規則性を使ってデータを異なるカテゴリに分類する処理』 *(PRML, C.M. Bishop)*

例) 手書き文字画像の認識



予習 7週目：ニューラルネットワーク

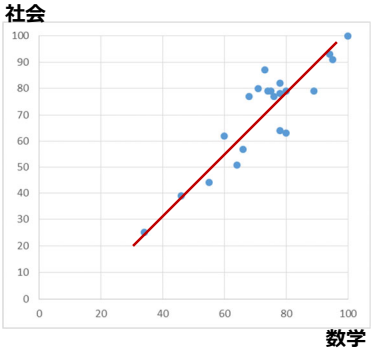
- 複数のユニット（ニューロン）を層状につないだネットワーク
- 入力層が特徴ベクトルを受け取り，出力層から識別結果が出力される

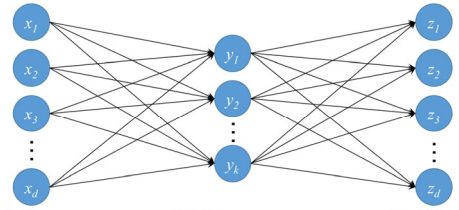


※ ここではニューラルネットワークの計算法を扱います

※ 学習方法（誤差逆伝搬）の詳細、様々な活性化関数、様々な学習方法（Drop out、バッチ正則化）、様々なネットワーク構造（RNN、LSTM、Attention）については扱いません。

予習 8週目：主成分分析とオートエンコーダ





入力
$\mathbf{x} \in R^d$

中間層 $\mathbf{y}$
$\mathbf{y} = \mathbf{f}(\mathbf{W}\mathbf{x} + \mathbf{b})$
$\mathbf{W}$：重み係数
$\mathbf{b}$：バイアス項
$\mathbf{f}(\ )$: 活性化関数

出力層 $\mathbf{z}$
$\mathbf{z} = \mathbf{f}(\bar{\mathbf{W}}\mathbf{y} + \bar{\mathbf{b}})$
$\bar{\mathbf{W}}$: 重み係数
$\bar{\mathbf{b}}$: バイアス項
$\mathbf{f}(\ )$: 活性化関数

画像処理技術に関連する研究紹介

目的
- 学術論文にふれる
- プログラミング課題（発展）を知る
- 深層学習の応用例を知る

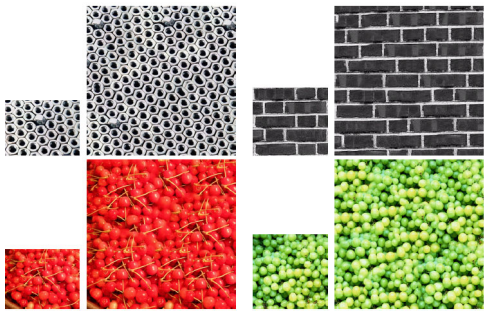
Contents
- Texture Synthesis by Non-parametric Sampling, ICCV 1999
- Seam Carving
- CNNによるImage in-painting

テクスチャ合成

- **テクスチャとは**，ここでは『物体表面に現れる模様』のことを指す
※分野によっては，触感・手触り・歯ざわりなどもテクスチャと呼ばれる

- **テクスチャ合成とは，**例となるテクスチャから新たなテクスチャを生成する技術のこと．

図は[Kwatra et al SIGGRAPH 2005]より



小さなテクスチャから大きなテクスチャを生成

画像は[Simakov et al. CVPR 2008]より

画像リサイズ

IEEE International Conference on Computer Vision, Corfu, Greece, September 1999

Texture Synthesis by Non-parametric Sampling

Alexei A. Efros and Thomas K. Leung
Computer Science Division
University of California, Berkeley
Berkeley, CA 94720-1776, U.S.A.
{efros,leungt}@cs.berkeley.edu

20年前に出たとても有名な論文です．
論文の実装と聞くと難しそうに感じるかもしれませんが実装自体はとても簡単です．
卒研では既存手法の実装も大切になります

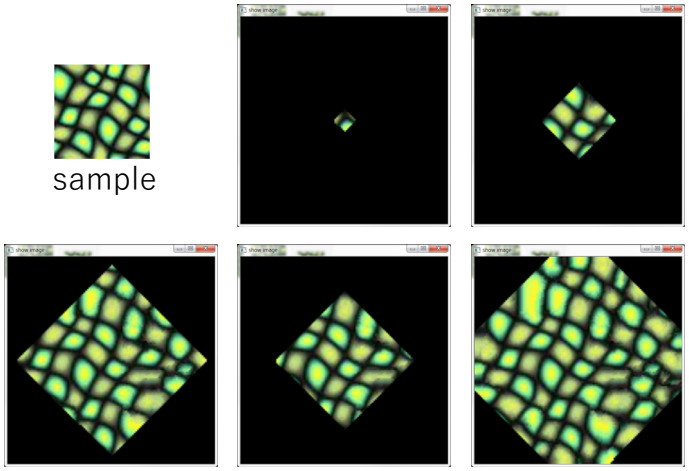
- ICCV 1999

背景：幅広い応用があるため『テクスチャ合成』は需要の高い技術．

課題：当時既存のテクスチャ合成技術は，Markov random fieldや周波数解析に基づくものだったが，多様なテクスチャに対してよい合成は難しかった．

提案：中心から徐々に合成を進めるアルゴリズムを提案．新しい画素値を埋めるときに入力画像から似た近傍領域を持つ画素をサンプリングする

実装例 – プログラミング課題として出題する予定



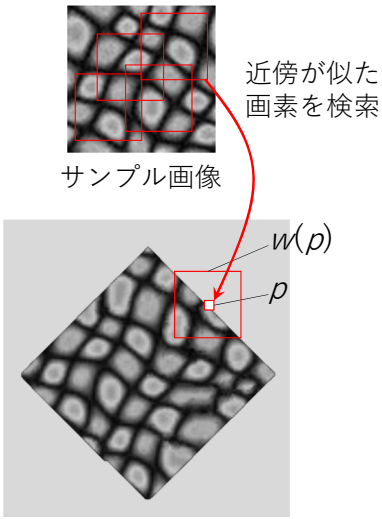
アルゴリズム（簡易版）

- 中央からテクスチャを成長させる

入力：サンプル画像 I_{smp}，窓サイズ k
出力：合成画像 I

1. 画像 I の中心3x3画素をランダム初期化
2. 以下を繰り返す
   - 2.1 既合成部分の隣接画素 p を選択
   - 2.2 pの近傍 $w(p)$ と最も似た領域 w_{best} を I_{smp} より検索
   - 2.3 w_{best}の中央画素値を p に代入
   - 2.4 全画素の合成がなされたら終了

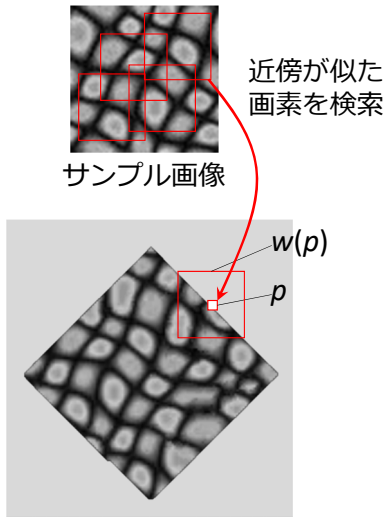
本当はもう少し複雑（次頁へ）



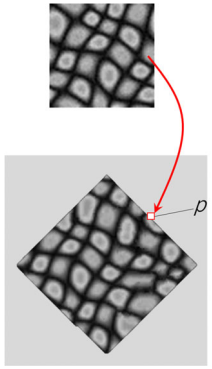
アルゴリズム

入力：サンプル画像 I_{smp}，近傍サイズ k
出力：合成画像 I

1. 画像 I の中心3x3画素をランダム初期化
2. 以下を繰り返す
   - 2.1 既合成部分の隣接画素 p を選択
   - 2.2 pの近傍 $w(p)$ と最も似た領域 w_{best}を I_{smp} より検索
   - 2.3 $d(w(p),w') \leq 1.1 * d(w(p),w_{best})$ を満たすすべての w'を I_{smp} より検索
   - 2.4 発見した複数のw'の中央画素値からヒストグラムを作成し，最も頻度が高いものを p に代入
   - 2.4 全画素の合成がなされたら終了

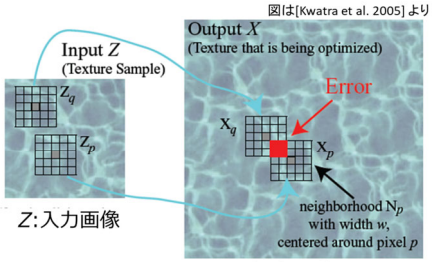


Texture合成技術



似たパッチを検索し
中心画素をコピー

図は[Kwatra et al. 2005]より

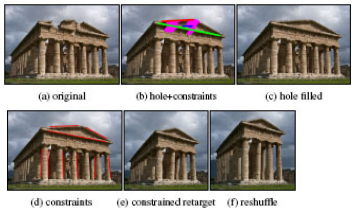


Z:入力画像

X：合成画像(出力)

似たパッチを検索しパッチをコピー
片方向・双方向の検索手法あり
Kwatra 2005/Simakov 2008

PatchMatch [Barnes2009]



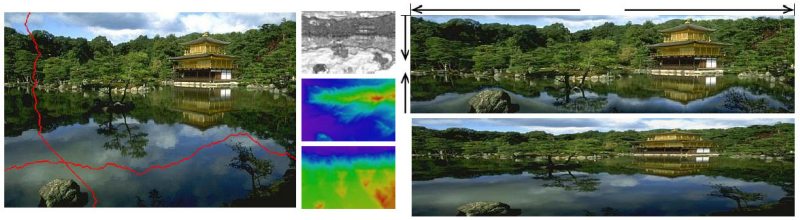
高速な類似パッチ検索による実時間画像生成
https://www.youtube.com/watch?v=fMe19oTz6vk

Seam Carving

Seam Carving for Content-Aware Image Resizing

Shai Avidan
Mitsubishi Electric Research Labs

Ariel Shamir
The Interdisciplinary Center & MERL



背景：コンテンツの縦横比を変化させたいことが多くある（image retargetingと呼ばれる）

問題：画像のretargetingにおいて単純に横方向に伸縮させると，撮影されたものの縦横比も変化してしまう

提案：画像全体の縦横比を変化させながら映ったものの縦横比を変化させないretargeting手法

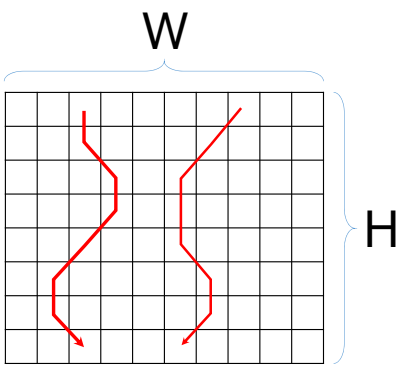
アイディア

- 各画素に重要度を定義（例：$I_x^2 + I_y^2$ or $|I_x| + |I_y|$ ）
- 横方向に縮める場合は，以下の条件を満たすseamを計算しそれを間引く
  - 画像の上辺から下辺へ画素をひとつずつつないだもの
  - ひとつ下に行く際，一画素分横方向に移動できる
  - 重要度の総和が最小となる

重要度マップ

[図は論文より]

解きたい問題



- サイズ W x H の画像に対して『通る画素値の総和が最小となるSeam』を求めよ
- ただし…
  - Seamは画素をつなぐものとする
  - Seamの始点は画像の上端，終点は画像の下端の画素とする
  - 一行分上から下に移動する際，左右に１画素分移動してもよい

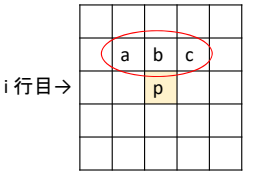
※画素にはその画素の重要度が入っている

- Seamの数は非常に多いので，すべてのseamを作成し，画素値の総和が最も小さいものを見つけるのは非現実的

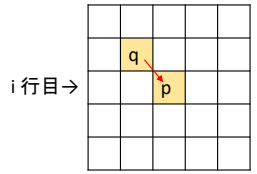
→ 動的計画法 (Dynamic Programing) が利用できる

解き方

- 各画素に以下の情報を書き込む
  - その画素に到達するSeamの画素値の総和
  - その画素に到達する直前の画素

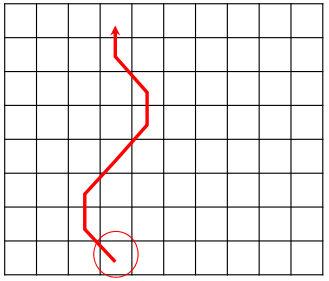


Pの上の行にあるabcから総和が最小のものを検索



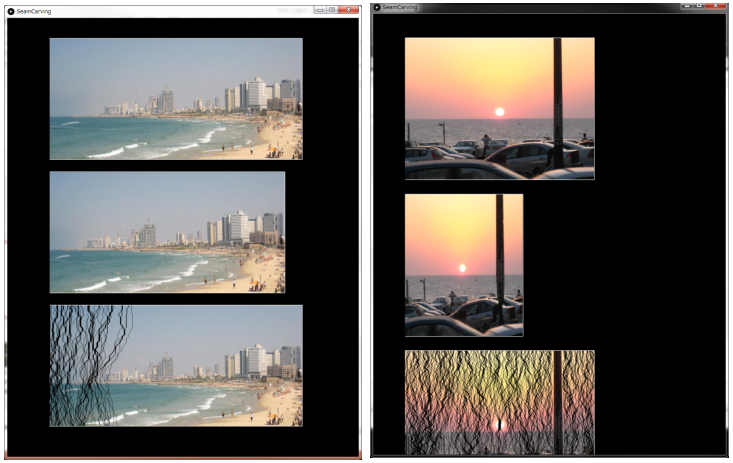
発見した画素をマークし，pの総和を更新

- Seamの検索アルゴリズム
  - 1行目の画素について、その画素値を総和として記入
  - 2行目から下端の行まで以下を繰り返す
    - 今 i-1 行目までは計算が終わっているものとする
    - i 行目のある画素 *p* に着目する
    - *p*の左上，上，右上のうち総和が最小のものを*q*とする
    - *q*までの総和と*p*の画素値を，*p*までの総和として記入
    - *画素pの直前の画素として，qをマーク*
  - *H行目において総和が最小の画素から上向きにたどると，Seamが得られる*



一番下の行の総和最小の画素から上向きにたどる

実装例 – プログラミング課題として出題します



Globally and Locally Consistent Image Completion

SATOSHI IIZUKA, Waseda University
EDGAR SIMO-SERRA, Waseda University
HIROSHI ISHIKAWA, Waseda University

画像補完を行なう深層学習の枠組みの提案　SIGGRPH 2017

研究背景と目的

研究背景
- Image Completionは非常に大きな需要を持つ課題
  - Image completion : 画像の一部を異なる画像埋める手法の総称
  - 応用先 → 不要なコンテンツの除去 / 汚れの除去

課題と目的
- 既存手法はテクスチャ合成（画像の一部をコピペ）によるもの
- 精度や速度に課題有り

提案
- CNNによる新たなImage Completion手法を提案
- 3種のネットワークからなる局所情報と大域情報を考慮したGAN型のCNNモデルを提案
  - Completion network
  - Local content discriminator
  - Global content discriminator

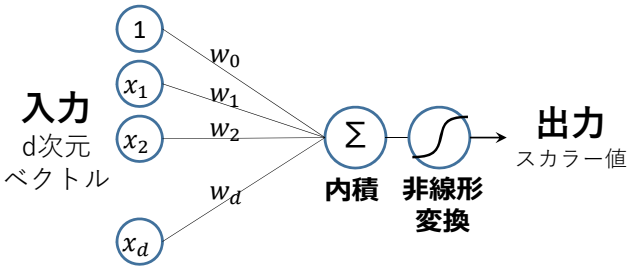
→ https://www.youtube.com/watch?v=5Ua4NUKowPU

ニューラルネットワークとは

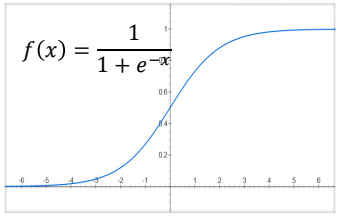
ユニット

- ニューラルネットワークを構成する最小単位をユニットと呼ぶ
- ユニットは，重み係数$\mathbf{w}$と非線形関数fを持つ
- 入力信号$\mathbf{x}$を受け取り，係数との内積$\mathbf{w}^T\mathbf{x}$を計算し，非線形関数にかけて，[0,1]の実数値を返す
- **係数$\mathbf{w}$は学習により最適化する**

訓練: 入出力の組をたくさん用意し、これらを満たすよう重みを調整



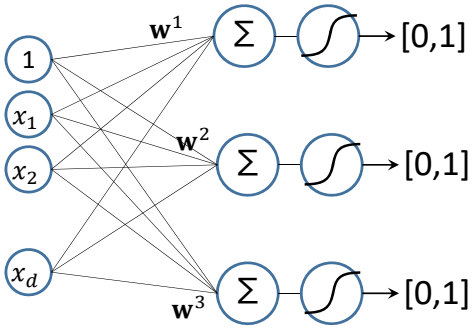
非線形変換部分には、シグモイド関数などが利用される



シグモイド関数

ユニットを並列につなぐ

- ユニットは入力信号が重みと似ているときに大きな値をかえすので…
- 複数のユニットを並列に並べたら，ベクトルを出力できる（複数クラスを扱える）



入力ベクトルが$\mathbf{w}^1$に似てるとき大きい値に

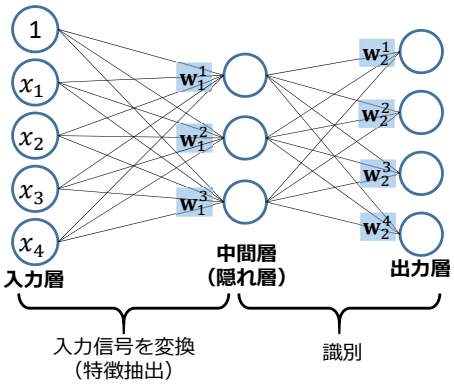
入力ベクトルが$\mathbf{w}^2$に似てるとき大きい値に

入力ベクトルが$\mathbf{w}^3$に似てるとき大きい値に

ユニットを直列につなぐ

- 入力層と出力層の間に中間層をはさみこむ
- 入力ベクトルを識別しやすい形に変換してから識別する

→ 線形分離不可能な問題にも対応できる



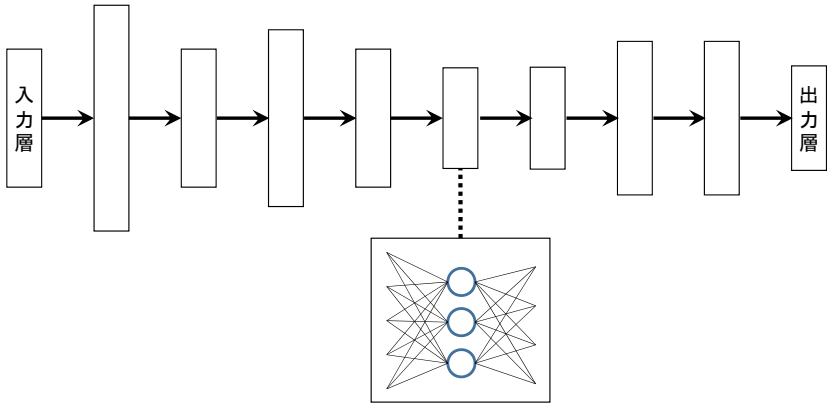
左図では非線形関数を省略（実際は計算）

重み係数が未知数で，これを教師データから学習する

→左図では 5*3 + 3*4 = 27個の未知数

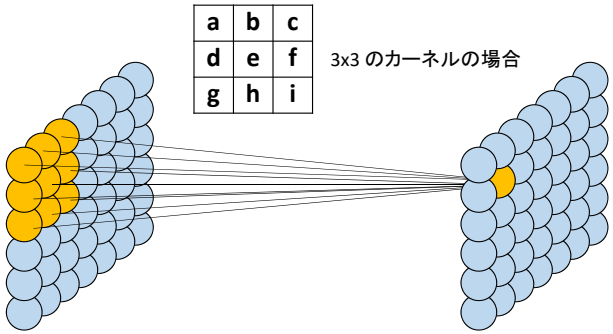
深層学習：Deep learning

- 7層 ~ 20層など，多層構造を持つニューラルネットワークのこと
- 特徴抽出を繰り返し，最後に識別を行なう（最近は違うものも多い）



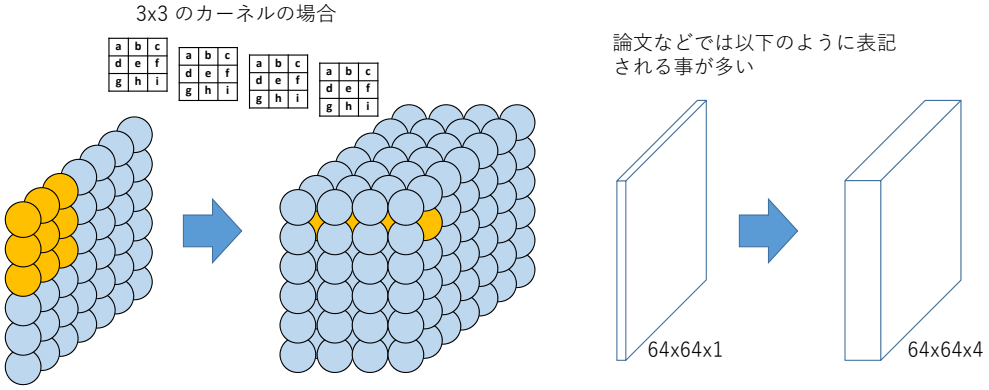
Convolutional Neural Network

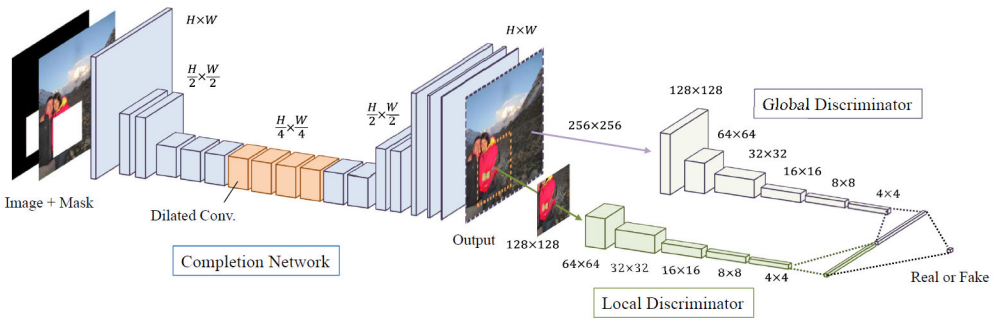
- 主に画像データに対して用いられるNeural Networkの一種
- 畳み込み（線形フィルタ）で入力画像を出力画像に変換
- フィルタの重みが変数で、これを訓練データより調整する



Convolutional Neural Network

- 線形フィルタを4個用意すると4枚の画像が出力される → channel方向に積み重ねて表現される

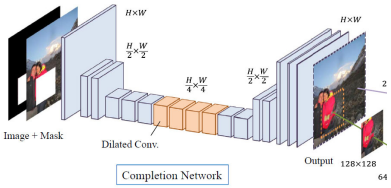




3種のCNNからなるネットワークモデルを提案する

- **Completion Network** : FCNNで，画像とマスク画像を入力とし，マスク部分を埋めた画像を出力する．
- **Global Discriminator** : 画像全体を受け取りそれが，元画像か生成画像かを判定
- **Local Discriminator** :画像の一部を受け取りそれが，元画像か生成画像かを判定

Completion Network



- Fully CNN
  - Input: RGB image (欠損有り)+ binary mask
  - Output: RBG image
  - ただし，実際の出力時には，mask以外の部分はinput画像で置き換える
- 基本的にencoder-decorder形式
- サイズを小さくするPoolingは2回だけ

Type	Kernel	Dilation (η)	Stride	Outputs
conv.	5 × 5	1	1 × 1	64
conv.	3 × 3	1	2 × 2	128
conv.	3 × 3	1	1 × 1	128
conv.	3 × 3	1	2 × 2	256
conv.	3 × 3	1	1 × 1	256
conv.	3 × 3	1	1 × 1	256
dilated conv.	3 × 3	2	1 × 1	256
dilated conv.	3 × 3	4	1 × 1	256
dilated conv.	3 × 3	8	1 × 1	256
dilated conv.	3 × 3	16	1 × 1	256
conv.	3 × 3	1	1 × 1	256
conv.	3 × 3	1	1 × 1	256
deconv.	4 × 4	1	1/2 × 1/2	128
conv.	3 × 3	1	1 × 1	128
deconv.	4 × 4	1	1/2 × 1/2	64
conv.	3 × 3	1	1 × 1	32
output	3 × 3	1	1 × 1	3

Completion Network

SKIP

Dilated convolutionを利用

$$\mathbf{y}_{u,v} = \sigma\left(\mathbf{b} + \sum_{i=-k'_h}^{k'_h}\sum_{j=-k'_w}^{k'_w} W_{k'_h+i,k'_w+j}\,\mathbf{x}_{u+\eta i,v+\eta j}\right),$$

$$k'_h = \frac{k_h - 1}{2}, \qquad k'_w = \frac{k_w - 1}{2}, \qquad (1)$$

※ kw, kh : カーネルサイズ，xuv : input map, yuv: output map, Wi,j:kernel, η :dilation量

※カーネルの影響領域を η 倍する

パラメータの量は同じでより広い領域から情報を集められる → image completionのようなタスクには非常に重要

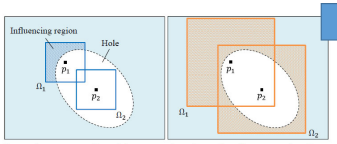


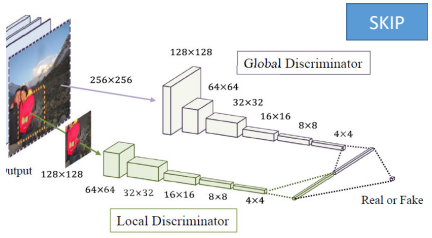
Fig. 3. Importance of spatial support. In order to be able to complete large regions, the spatial support used to compute an output pixel must include pixels outside of the hole. On the left, the pixel p_1 is computed from the influencing region in the spatial support Ω_1, while the pixel p_2 cannot be calculated since the supporting area Ω_2 does not contain any information outside of the hole. However, on the right side, the spatial support is larger than the hole, allowing the completion of the center pixels.

Type	Kernel	Dilation (η)	Stride	Outputs
conv.	5 × 5	1	1 × 1	64
conv.	3 × 3	1	2 × 2	128
conv.	3 × 3	1	1 × 1	128
conv.	3 × 3	1	2 × 2	256
conv.	3 × 3	1	1 × 1	256
conv.	3 × 3	1	1 × 1	256
dilated conv.	3 × 3	2	1 × 1	256
dilated conv.	3 × 3	4	1 × 1	256
dilated conv.	3 × 3	8	1 × 1	256
dilated conv.	3 × 3	16	1 × 1	256
conv.	3 × 3	1	1 × 1	256
conv.	3 × 3	1	1 × 1	256
deconv.	4 × 4	1	1/2 × 1/2	128
conv.	3 × 3	1	1 × 1	128
deconv.	4 × 4	1	1/2 × 1/2	64
conv.	3 × 3	1	1 × 1	32
output	3 × 3	1	1 × 1	3

Context discriminators

SKIP

- Global discriminator
  - Input 画像全体を256x256にスケールしたもの
  - Output 1024次元ベクトル
- Local discriminator
  - Input 画像のpatch 128x128
  - Output 1024次元ベクトル
  - 学習時には以下を利用
    - fake : maskの中心を中心とした128x128
    - true : ランダムにパッチを選択
- 最終層で2048を1次元ベクトルに



(a) Local Discriminator

Type	Kernel	Stride	Outputs
conv.	5 × 5	2 × 2	64
conv.	5 × 5	2 × 2	128
conv.	5 × 5	2 × 2	256
conv.	5 × 5	2 × 2	512
conv.	5 × 5	2 × 2	512
FC	-	-	1024

(b) Global Discriminator

Type	Kernel	Stride	Outputs
conv.	5 × 5	2 × 2	64
conv.	5 × 5	2 × 2	128
conv.	5 × 5	2 × 2	256
conv.	5 × 5	2 × 2	512
conv.	5 × 5	2 × 2	512
conv.	5 × 5	2 × 2	512
FC	-	-	1024

SKIP

Training : Loss function

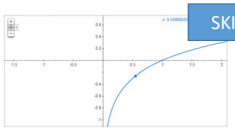
二種のLossを考慮

Mean squared error

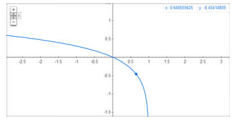
$$L(x, M_c) = \| M_c \odot (C(x, M_c) - x) \|^2 \ ,$$

※生成結果と元画像の差がコスト

Log x



Log 1-x



Generative adversarial network loss

$$\min_C \max_D \ \mathbb{E}[\ \log D(x, M_d) + \log(1 - D(C(x, M_c), M_c)\ ] \ ,$$

※true画像xに D(x,Md)＝1 と出力したい

※fake画像C(x,Mc)に D(C(x,Mc),Mc)＝0と出力したい

C(x,Mc): 入力画像 x, マスクMcに対するcompletion networkの出力

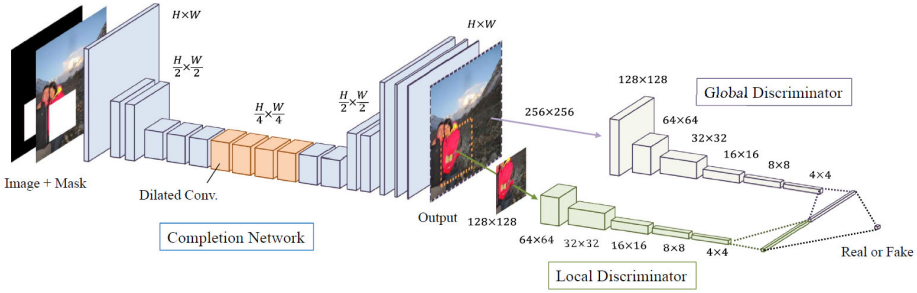
D(x,Md):入力画像 x, マスクMdに対するdiscriminatorの出力、trueなのでなるべく1にしたい

D(C(x,Mc),Mc):completion networkの出力に対するdiscriminatorの出力、fakeなのでなるべく0にしたい

SKIP

Training : stabilize

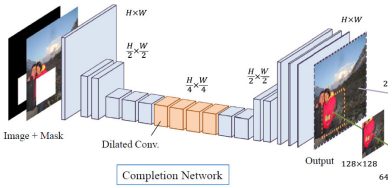
- Generatorとdiscriminatorの両方をバランスしながら訓練する必要があるがこれは結構難しい
- 以下の手順で訓練を進める
  - 1. 先ずgeneratorのみを訓練 (MSE Lossのみを利用)
  - 2. generatorをfixしてdiscriminatorを訓練 (GAN lossのみ)
  - 3. generatorとdiscriminatorを同時に訓練 (MSE + GAN loss)



Training data & Results

- 訓練データ作成

1. 写真データを用意
2. ランダムにscalingして 256x256の領域を抽出
3. ランダムにmaskを生成（1辺が96~128の範囲に）



結果:

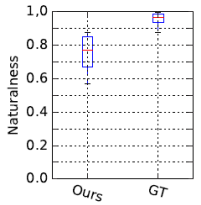
- Place2 dataset（ 8M枚の訓練画像）を用意
- Training
  - Completion network only – 90,000 iteration
  - discriminator only – 10,000 iteration
  - both 500,000 iteration
  - 2 Months on four tesla K80 GPUs
- 訓練後に画像補完にかかる時間は右図

Image Size	Pixels	CPU (s)	GPU (s)	Speedup
512 × 512	409,600	2.286	0.141	16.2×
768 × 768	589,824	4.933	0.312	15.8×
1024 × 1024	1,048,576	8.262	0.561	14.7×

User Study

- 10人の参加者
- 提案法で生成したものとデータセットのセレブの顔画像をみせ，どちらが生成したものかを質問
- 右下図は，”real”であると回答された割合

- Real 画像 : 96.5% が realと判断された
- 生成画像 : 77.0%がrealと判断された



Method	Naturalness (median)
GT	96.5%
Ours	77.0%

Fig. 11. Result of our user study evaluating the naturalness of the image completion on the CelebA dataset. The numbers are the percentage of the images that are deemed to be real by 10 different users for the Ground Truth (GT) and the result of the completion by our approach.

まとめ

Globally and Locally Consistent Image Completion

SATOSHI IIZUKA, Waseda University
EDGAR SIMO-SERRA, Waseda University
HIROSHI ISHIKAWA, Waseda University

Fig. 1. Image completion results by our approach. The masked area is shown in white. Our approach can generate novel fragments that are not present elsewhere in the image, such as needed for completing faces; this is not possible with patch-based methods. Photographs courtesy of Michael D Beckwith (CC0), Mon Mer (Public Domain), davidgsteadman (Public Domain), and Owen Lucas (Public Domain).

- Image completion taskに非常に良い精度を発揮できる深層学習法を提案した
- Globally and locally discriminatorを導入したnetwork architectureを提案した
- 実際に多様な画像に対してimage completionを行いその精度を示した

まとめ

- 本科目の実施方法
  - 01~08回目：対面 + 小テスト
  - 09~14回目：プログラミング演習

- 画像処理・画像認識に関する研究紹介
  - テクスチャ合成
  - シームカービング
  - 深層学習によるImage in-painting

前期14週の間、よろしくおねがいします！