

# ディープネットワークを用いた 白黒写真の自動色付け

---

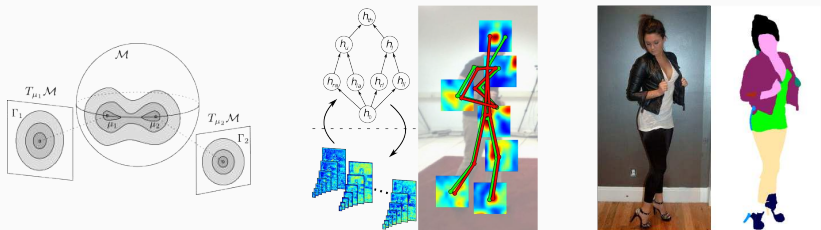
シモセラ エドガー

2018年1月24日(水)

早稲田大学

# 自己紹介

- 2015年7月に BarcelonaTech で博士号を取得
- 2015年8月から 2017年3月まで早稲田大学の研究院助教
- 2017年4月から 同大学の研究院講師

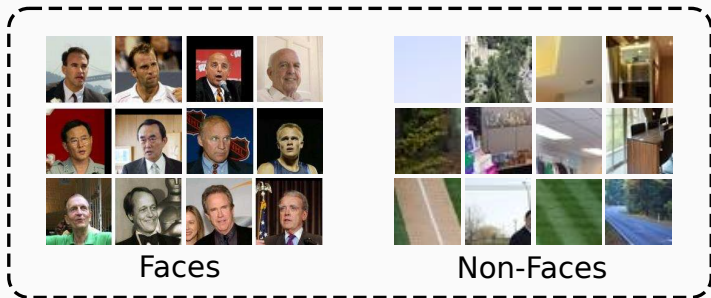


1. ディープラーニングにおける画像変換
2. 白黒写真の自動色付け
3. GAN について
4. 画像補完

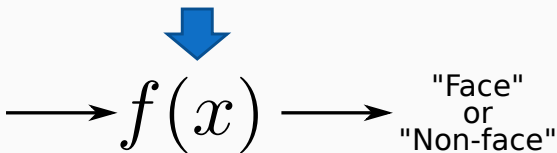


# ディープラーニングにおける画像変換

---

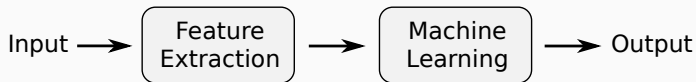


Training Data



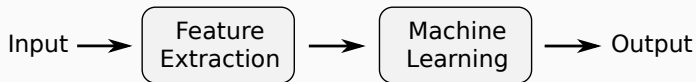
# ディープラーニングと機械学習

- ・ 手動で決めた特徴を入力
- ・ 機械学習モデルは特徴を出力に変換
- ・ 精度は特徴に拘束されている

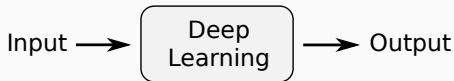


# ディープラーニングと機械学習

- ・ 手動で決めた特徴を入力
- ・ 機械学習モデルは特徴を出力に変換
- ・ 精度は特徴に拘束されている

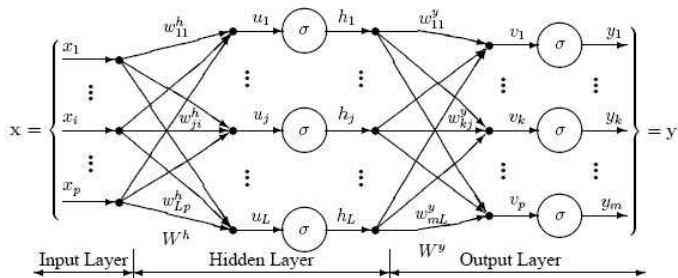


- ・ ディープラーニングは出力と入力のマッピングを学習
- ・ ヒューリスティックを避けて、データ依存になる



# ディープラーニング

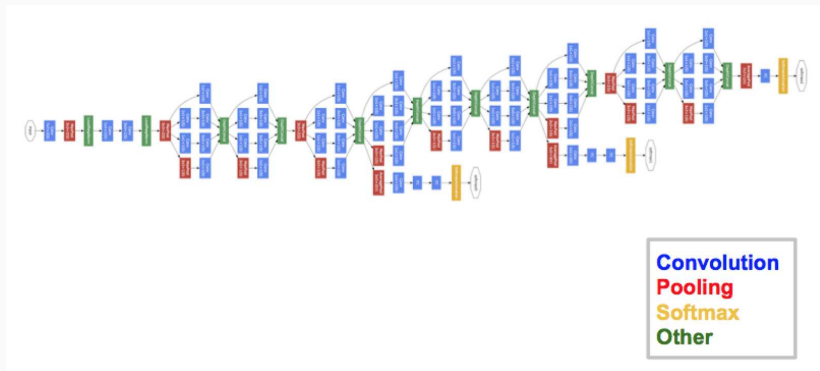
- ・ 現代のニューラルネットワーク
  - ・ GPU による計算効率
  - ・ 大規模なデータセット
- ・ パラメータを学習
- ・ ハイパーパラメータを手動で設定





# ディープラーニングの特徴

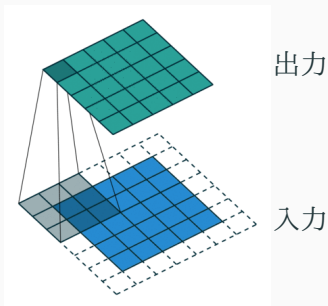
- ・ 深い多層モデルを使用
- ・ 学習が数日あるいは数週間かかる
- ・ GPUが必要



Szegedy, Christian, et al. *Going deeper with convolutions*. CVPR, 2015.

# 畳み込み層

- ・ 畳み込み演算で層をつなぐ
- ・ 同時に複数のフィルタをかける
- ・ ハイパーパラメーター：カーネル、パディング、ストライド
  - ・ 重みはカーネルで表現
  - ・ パディングで入力サイズを保持
  - ・ ストライドで解像度を変換



# 畳み込みの種類

ストライド数による解像度が変わる

## 1. Flat-convolution

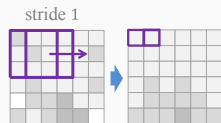
1.1 カーネル  $3 \times 3$ , パディング  $1 \times 1$ , ストライド 1

## 2. Down-convolution

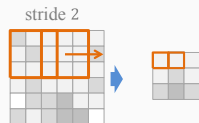
2.1 カーネル  $3 \times 3$ , パディング  $1 \times 1$ , ストライド 2

## 3. Up-convolution

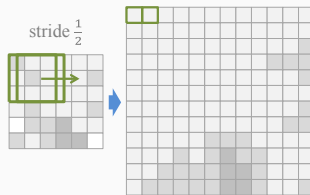
3.1 カーネル  $3 \times 3$ , パディング  $1 \times 1$ , ストライド  $1/2$



Flat-convolution



Down-convolution



Up-convolution

# 畳み込みの種類

ストライド数による解像度が変わる

## 1. Flat-convolution

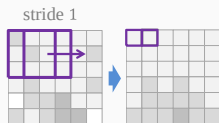
1.1 カーネル  $3 \times 3$ , パディング  $1 \times 1$ , ストライド 1

## 2. Down-convolution

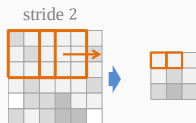
2.1 カーネル  $3 \times 3$ , パディング  $1 \times 1$ , ストライド 2

## 3. Up-convolution

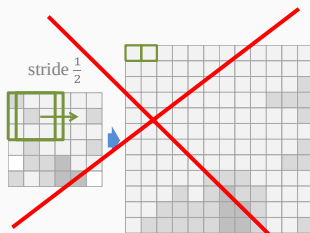
3.1 カーネル  $3 \times 3$ , パディング  $1 \times 1$ , ストライド  $1/2$



Flat-convolution



Down-convolution



Up-convolution

# 畳み込みの種類

ストライド数による解像度が変わる

## 1. Flat-convolution

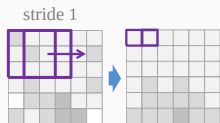
1.1 カーネル  $3 \times 3$ , パディング  $1 \times 1$ , ストライド 1

## 2. Down-convolution

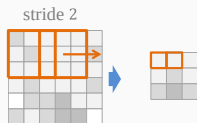
2.1 カーネル  $3 \times 3$ , パディング  $1 \times 1$ , ストライド 2

## 3. Up-convolution

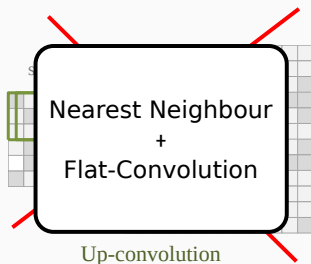
3.1 カーネル  $3 \times 3$ , パディング  $1 \times 1$ , ストライド  $1/2$



Flat-convolution

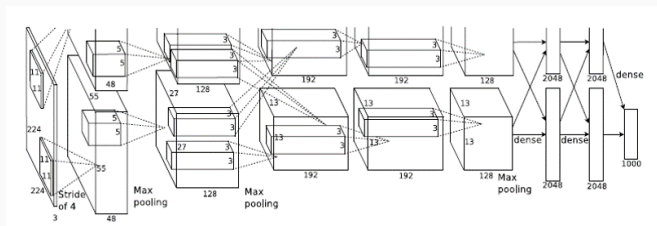


Down-convolution



# 全層畳み込みニューラルネットワーク

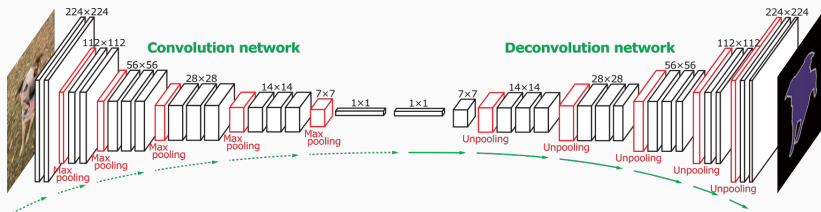
- ・ 全結合層を使うと入力サイズが固定される
- ・ 全層畳み込み層で任意の解像度を入力可



Krizhevsky et al. *ImageNet Classification with Deep Convolutional Neural Networks*. NIPS, 2012.

# 全層畳み込みニューラルネットワーク

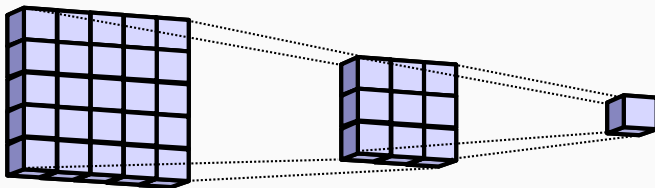
- ・ 全結合層を使うと入力サイズが固定される
- ・ 全層畳み込み層で任意の解像度を入力可
- ・ エンコーダー・デコーダー型
  - ・ 最初に解像度を下げる → メモリ削減、学習加速
  - ・ 必要なら元の解像度に戻す



Noh et al. *Learning Deconvolution Network for Semantic Segmentation*. ICCV, 2015.

# 全層畳み込みニューラルネットワーク

- ・ 全結合層を使うと入力サイズが固定される
- ・ 全層畳み込み層で任意の解像度を入力可
- ・ エンコーダー・デコーダー型
  - ・ 最初に解像度を下げる → メモリ削減、学習加速
  - ・ 必要なら元の解像度に戻す
- ・ 畳み込み層
  - ・ 3 x 3 カーネルで重みを減らす
  - ・ 空間サポートが大事





## 白黒写真の自動色付け

---

- 落書きベース [Levin+ 2004; Yatziv+ 2004; An+ 2009; Xu+ 2013; Endo+ 2016]
  - 色を落書きで設定
  - 手動入力必要
- 参考画像ベース [Chia+ 2011; Gupta+ 2012]
  - 他の写真を参考にして色付け
  - 似てる画像必要



Levin+ 2004



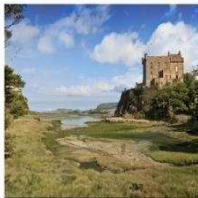
Input

Reference

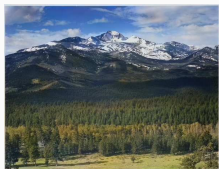
Output

Gupta+ 2012

- End-to-end で学習できる 大域特徴と 局所特徴の統合による自動色付けのネットワーク
  - 新しい統合レイヤで大域特徴を局所特徴に練り込む
  - 学習で分類ラベルを利用
  - 百年前の写真を現実的に色付ける



# 自動色付け



Colorado Park, 1941 年



Textile Mill, 1937 年



Berry Field, 1909 年



Hamilton, 1936 年

The Lost World (1925)



# データについて

- ・ 簡単に作れる
- ・ 現代のカラー写真を白黒にする
- ・ MIT の Places データセットを使用 [Zhou et al. 2014]



Abbey



Airport terminal



Aquarium



Baseball field

...



Dining room



Forest road



Gas station



Gift shop

...

## 問題の難度

- ・ パッチをだけ見ると空と天井を 区別しにくい
- ・ モデルを改良する必要



正解データ

普通の CNN

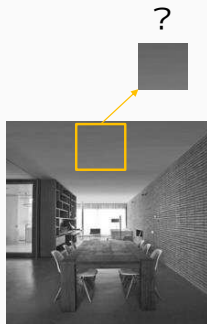
# 問題の難度

- ・ パッチをだけ見ると空と天井を 区別しにくい
- ・ モデルを改良する必要



正解データ

普通の CNN





## 問題の難度

- ・ パッチをだけ見ると空と天井を 区別しにくい
- ・ モデルを改良する必要



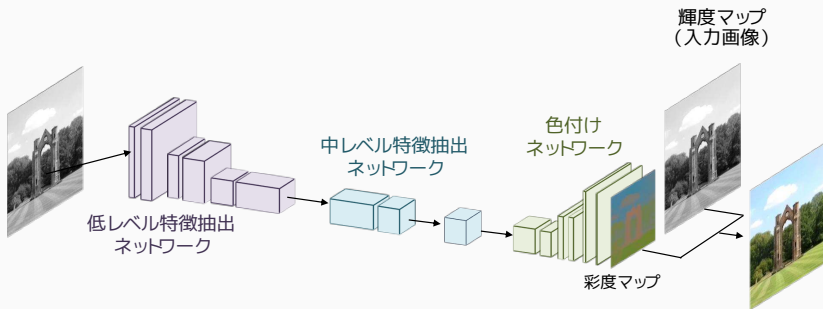
正解データ

普通の CNN

提案手法

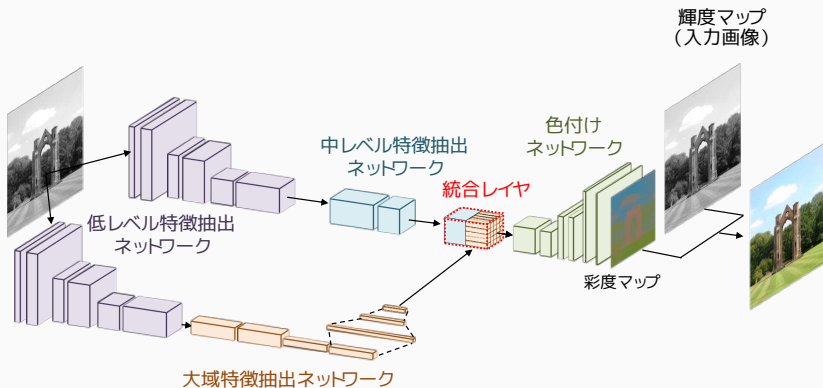
# モデルの改造

- ・ モデルは色を予想し、白黒写真に加える
- ・ 問題はパッチの情報が足りない



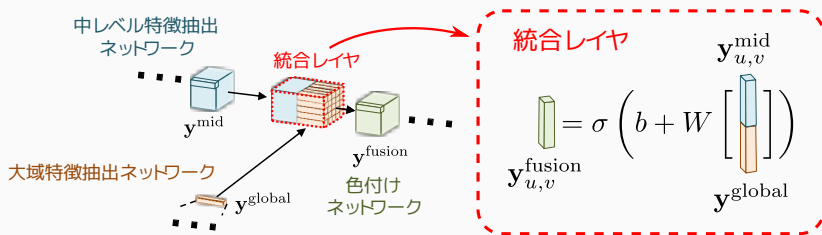
# モデルの改造

- ・ モデルは色を予想し、白黒写真に加える
- ・ 問題はパッチの情報が足りない
- ・ 解決方法：大域特徴を利用
- ・ 提案の統合レイヤでパッチと大域特徴を結合



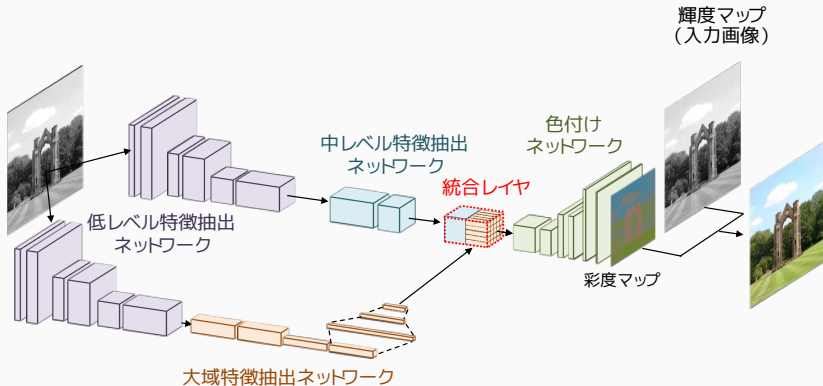
# 統合レイヤ

- ・ 大域特徴と局所特徴を統合する層
- ・  $W$ 、 $b$ を学習させる
- ・ 大域特徴を局所特徴に練り込む
- ・ どんな解像度でも対応



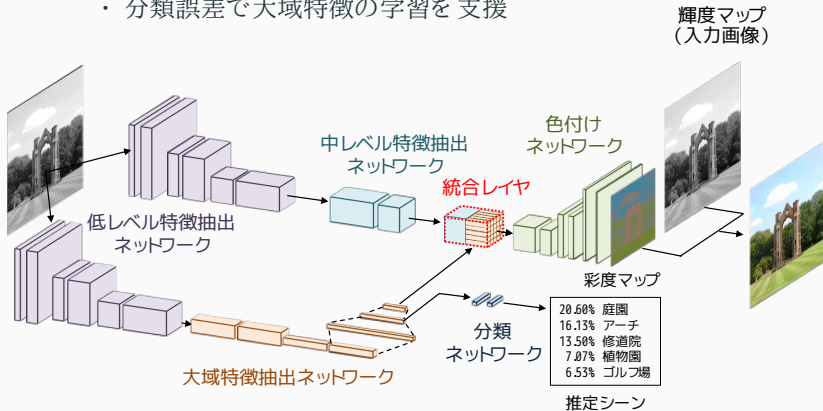
# モデルと学習

- ・ 全層をランダムから学習
- ・ ADADELTAを使用



# モデルと学習

- ・ 全層をランダムから学習
- ・ ADADELTAを使用
- ・ 2つのロスを使用
  - ・ MSEロスで色付けを学習させ
  - ・ 分類誤差で大域特徴の学習を支援



# 色空間の影響



正解



RGB



YUV



L\*a\*b\*

- ・ 以外と影響がない
- ・ L\*a\*b のほうがちょっといい

# 大域特徴でスタイル変換

局所

大域

局所

大域

局所

大域

