クレジット：

Mathematics and Informatics Center メディアプログラミング入門 2020 山肩洋子

メディアプログラミング入門

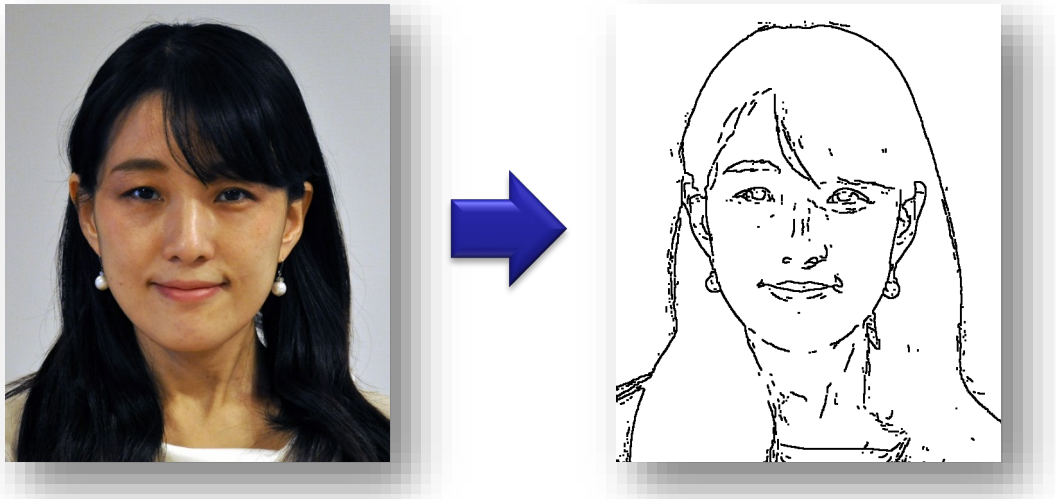
第６回：深層学習による高度な画像認識

火５@本郷 2020年7月7日
情報理工学系研究科 数理・情報教育研究センター
准教授 山肩 洋子

第５回の課題：写真から線画のイラストを作ろう！

今回は原画像を皆さんに用意してもらうことにします。

- 原画像はスマホで撮影してパソコンに取り込んだり、パソコン内蔵のカメラを使って撮影したり、パブリックドメインの画像をWebからダウンロードすることで用意してください
- 原画像はこの課題がおかれているフォルダなどにおいて、その写真のパスを、ノートブックの一番最初のコードセルにある変数`imgfile`に代入
- 各処理において、それぞれパラメータを調整してきれいな線画を作成する

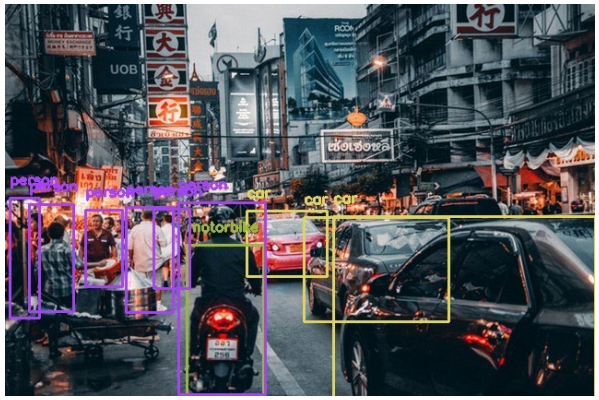


第6回 深層学習による画像処理：顔検出・一般物体認識

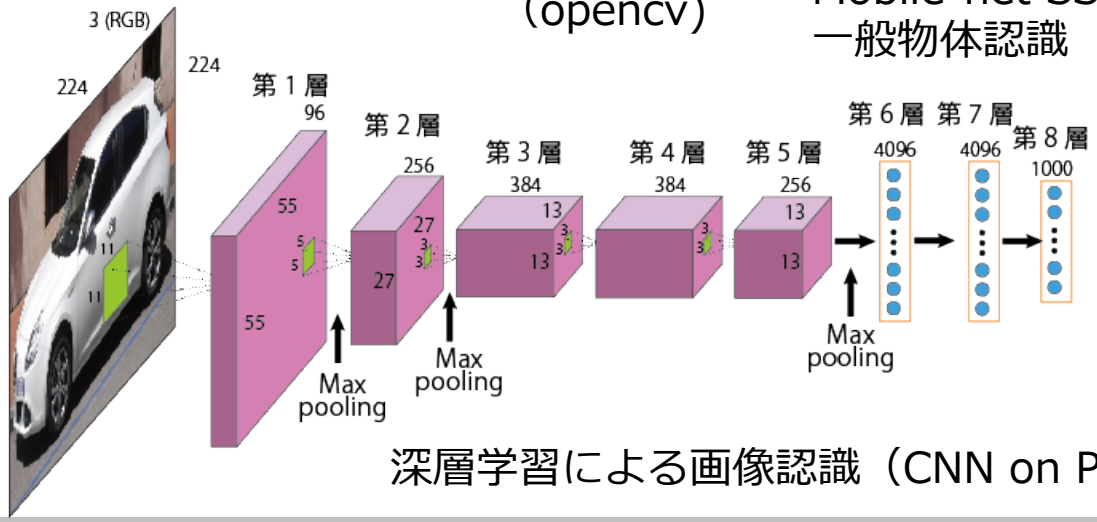
講義内容：高度な画像処理による領域検出や一般物体認識の仕組みを知ろう

演習内容：顔画像検出の手法を理解したのち、深層学習による画像認識手法CNNの仕組みを学ぶ。より発展的な画像認識課題を知り、体験する。

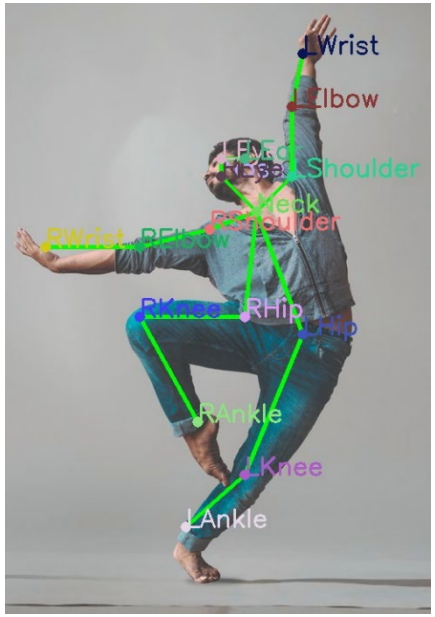
Harr-like特徴量による顔画像検出（opencv）



Mobile-net SSDによる一般物体認識



深層学習による画像認識（CNN on PyTorch）



OpenPoseによる人物姿勢推定

本日の学習内容

- 顔画像検出：Vola-Johnes法
  - Harr-like feature
- 深層学習による画像認識
  - Convolutional Neural Network (CNN)
- 画像認識における様々なタスクと手法
    - 手書き文字認識
    - 一般物体認識、領域抽出
    - 画像や動画からの字幕生成（image/video captioning)
  - 演習
    - 一般物体認識： MobileNet-SSD
    - 人物姿勢推定：OpenPose
- 深層学習による画像生成
  - 敵対的生成ネットワーク：GAN
  - GANを使った様々なタスク
    - 顔画像生成、スタイル変換、線画の着色、フォント生成、超解像度画像生成

Haar-like特徴量による顔画像検出

演習）ImageRecognition1.ipynb

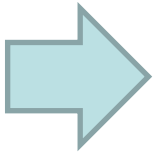
Viola-Jones法：Haar-like特徴量による顔画像検出

- 白と黒の矩形領域の組み合わせでできた様々なパターンを顔に当てはめて一致度を計算
  (矩形の黒領域の輝度の合計) – (白領域の輝度 の合計)
- 様々な顔画像と一致度の高いパターンは顔検出に有効
  - **AdaBoost**によりモデル学習（各特徴量の有効性に応じて重みづけ）
- 入力画像のサイズを変えながら、学習したモデルをあらゆる場所に当てはめることで、顔っぽい領域を見つけていく

著作権等の都合上、ここに挿入されていた画像を削除しました。

特徴量プロトタイプ

Fig.2



著作権等の都合上、ここに挿入されていた画像を削除しました。

AdaBoostによる最初と2番目に抽出された特徴量

Fig.3

Rainer Lienhart and Jochen Maydt. “An Extended Set of Haar-like Features for Rapid Object Detection.” IEEE ICIP 2002, Vol. 1, pp. 900-903, Sep.2002
https://ieeexplore.ieee.org/document/1038171

ref. Paul Viola and Michael Jones, “Rapid object detection using a boosted cascade of simple features, CVPR2001, 2001.

アンサンブル学習 (Ensemble learning)

三人寄れば文殊の知恵

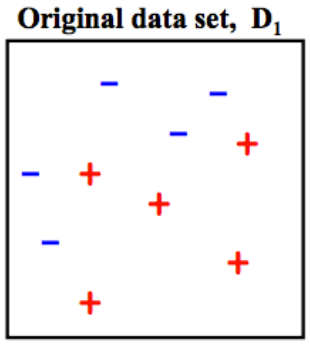
「特別に頭の良い者でなくても三人集まって相談すれば何か良い知恵が浮かぶものだ、という意味。」(ウィクショナリーより)

- すべての学習データを使って強い判別器を１つつくるのではなく、その学習データをいくつかに分けたり、繰り返し使ったりしながら、**複数の弱い判別器**を作る
- その複数の弱い判別器が出した結果を統合して、一つの結果を導く
  - バギング（例：ランダムフォレスト）
  - **ブースティング（例：アダブースト）**

ブースティング

教訓：人ができないことができる人間になろう！

- 判別に失敗するデータを判別できる識別器を作る
- 判別に失敗したデータの重み（重要さ）を大きくして識別器を学習

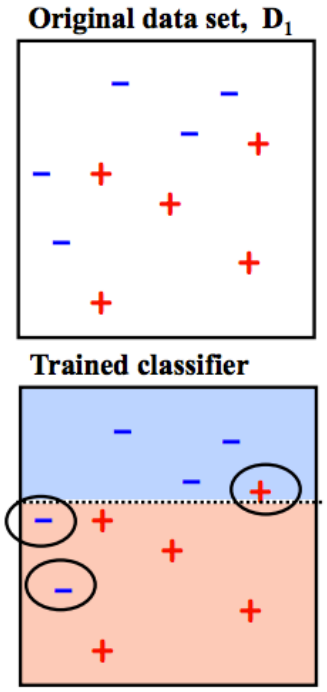


“adaboost” by Alexander Ihler
https://ucirvine.instructure.com/files/1095922/download?download_frd=1

ブースティング

教訓：人ができないことができる人間になろう！

- 判別に失敗するデータを判別できる識別器を作る
- 判別に失敗したデータの重み（重要さ）を大きくして識別器を学習

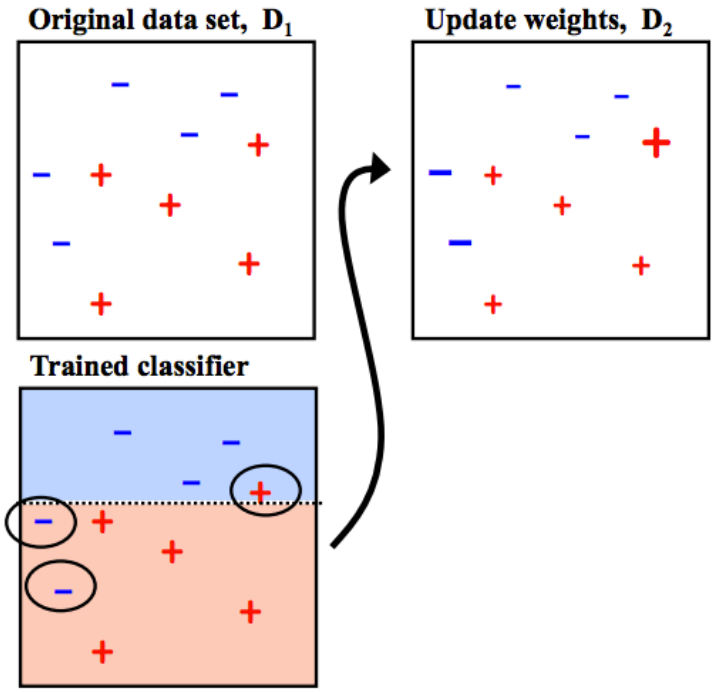


“adaboost” by Alexander Ihler
https://ucirvine.instructure.com/files/1095922/download?download_frd=1

ブースティング

教訓：人ができないことができる人間になろう！

- 判別に失敗するデータを判別できる識別器を作る
- 判別に失敗したデータの重み（重要さ）を大きくして識別器を学習

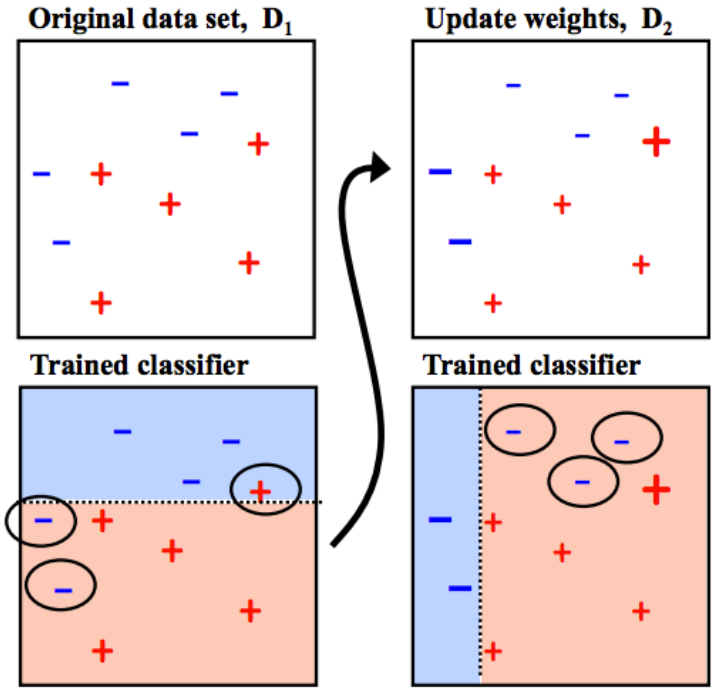


“adaboost” by Alexander Ihler
https://ucirvine.instructure.com/files/1095922/download?download_frd=1

ブースティング

教訓：人ができないことができる人間になろう！

- 判別に失敗するデータを判別できる識別器を作る
- 判別に失敗したデータの重み（重要さ）を大きくして識別器を学習

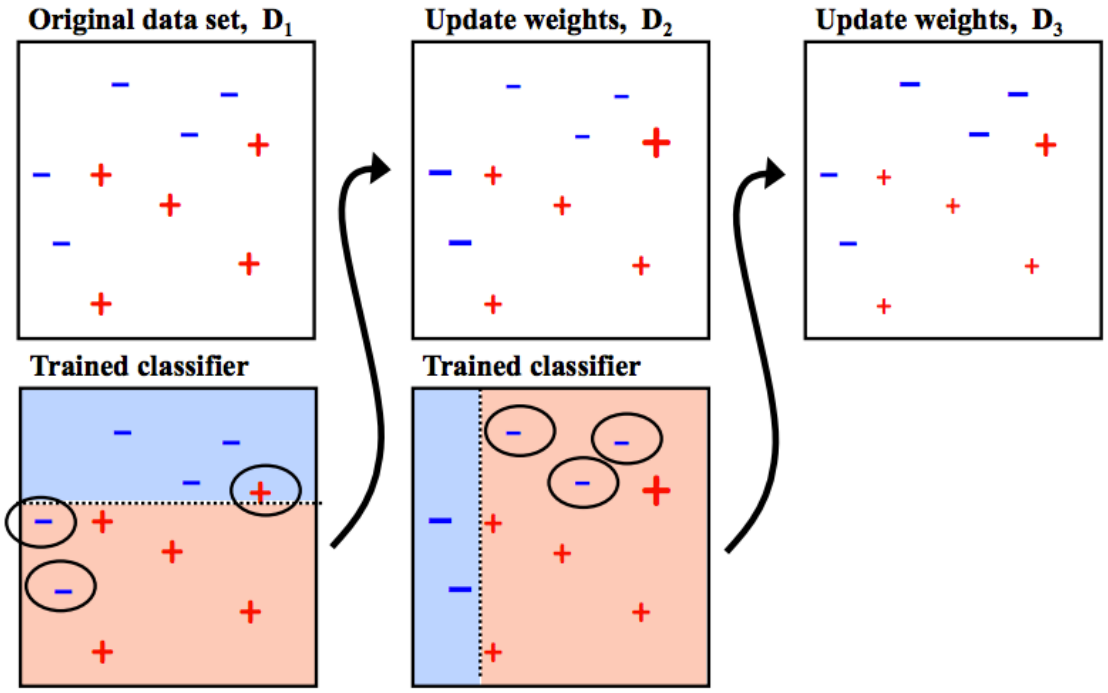


“adaboost” by Alexander Ihler
https://ucirvine.instructure.com/files/1095922/download?download_frd=1

ブースティング

教訓：人ができないことができる人間になろう！

- 判別に失敗するデータを判別できる識別器を作る
- 判別に失敗したデータの重み（重要さ）を大きくして識別器を学習

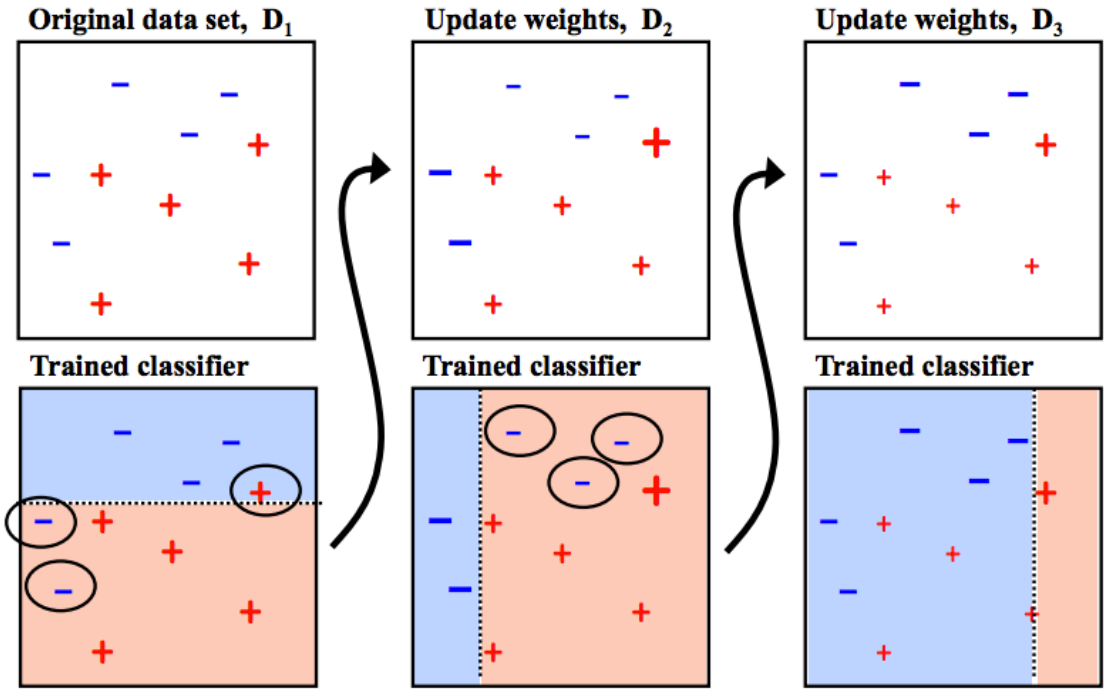


“adaboost” by Alexander Ihler
https://ucirvine.instructure.com/files/1095922/download?download_frd=1

ブースティング

教訓：人ができないことができる人間になろう！

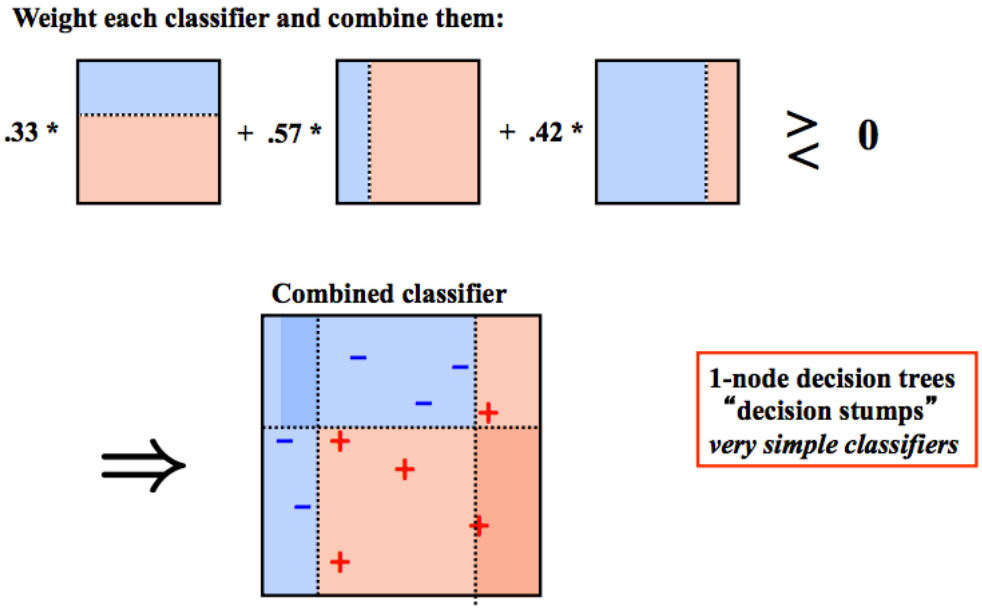
- 判別に失敗するデータを判別できる識別器を作る
- 判別に失敗したデータの重み（重要さ）を大きくして識別器を学習



"adaboost" by Alexander Ihler
https://ucirvine.instructure.com/files/1095922/download?download_frd=1

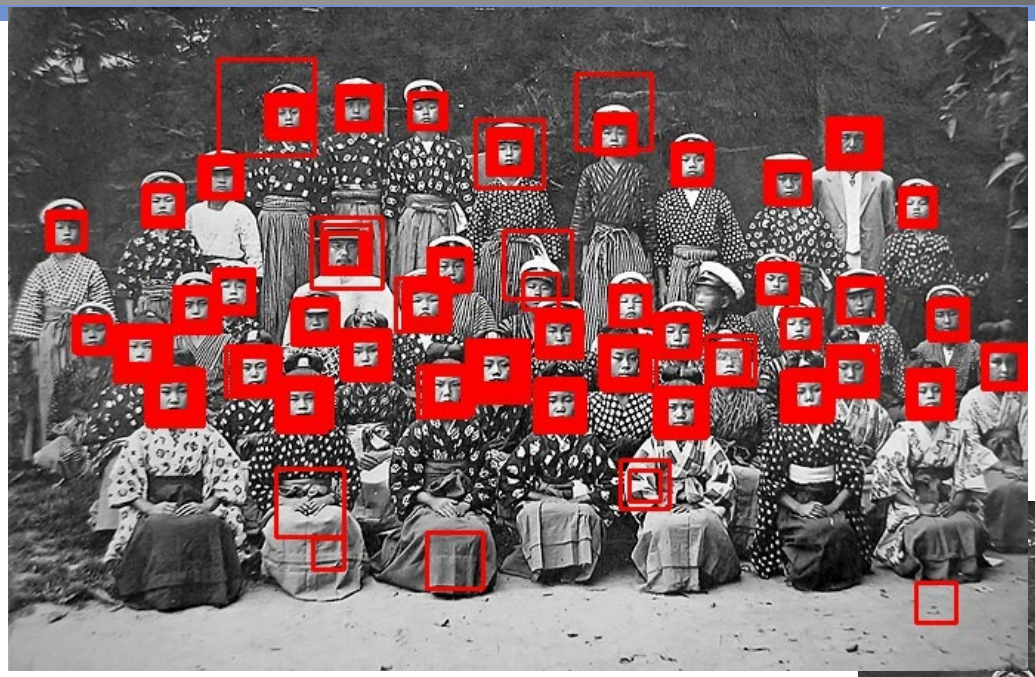
ブースティング

- 識別器の精度によって重みをかける
  - ただし、重みが大きいデータ（間違い安いデータ）を識別できるものをより重用
- 個々の識別器が出した結果を重み付きで投票して、多い方（重みの総和が大きい方）の結果を選ぶ



"adaboost" by Alexander Ihler
https://ucirvine.instructure.com/files/1095922/download?download_frd=1

顔領域の検出結果



- 画像サイズを徐々に小さくしながらモデルと適合するかをスキャン
- 顔ではないところも誤検出される
- 真の顔領域であれば、画像のサイズを変えるたびに、顔領域として何度も検出される

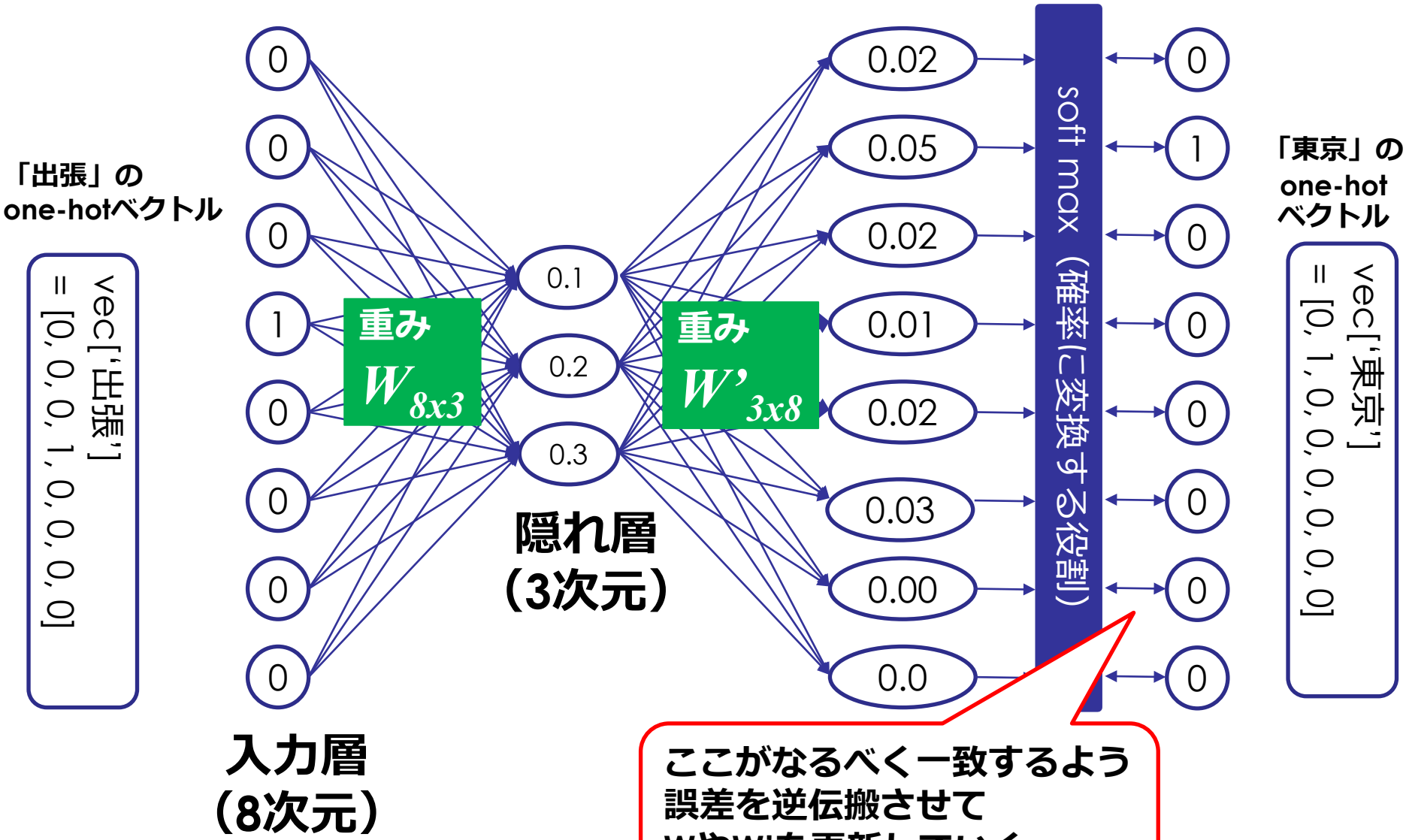
2回以上検出された領域を
顔として検出

ref. (2020/4/6): Wikimedia commons: File:Kasahara Saitama Kasahara Jinjo Elementary School 1920 1.jpg パブリックドメイン
https://ja.wikipedia.org/wiki/%E3%83%95%E3%82%A1%E3%82%A4%E3%83%AB:Kasahara_Saitama_Kasahara_Jinjo_Elementary_School_1920_1.jpg

ニューラルネットワーク

演習：ImageRecognition2.ipynb

[再掲] Word2vec: Skip-gram



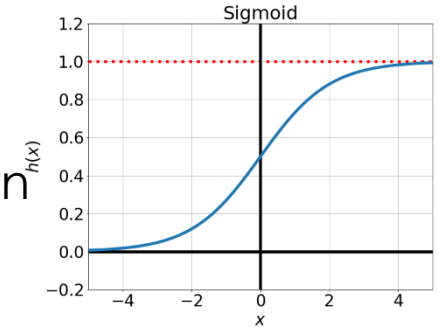
ニューラルネットワーク

- 入力3次元、出力2次元
- 中間層1層（ユニット数２）

$h(x)$は活性化関数
よく使う活性化関数：

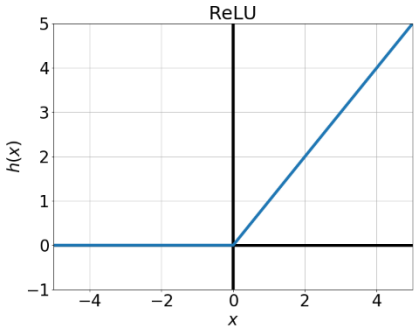
- sigmoid function

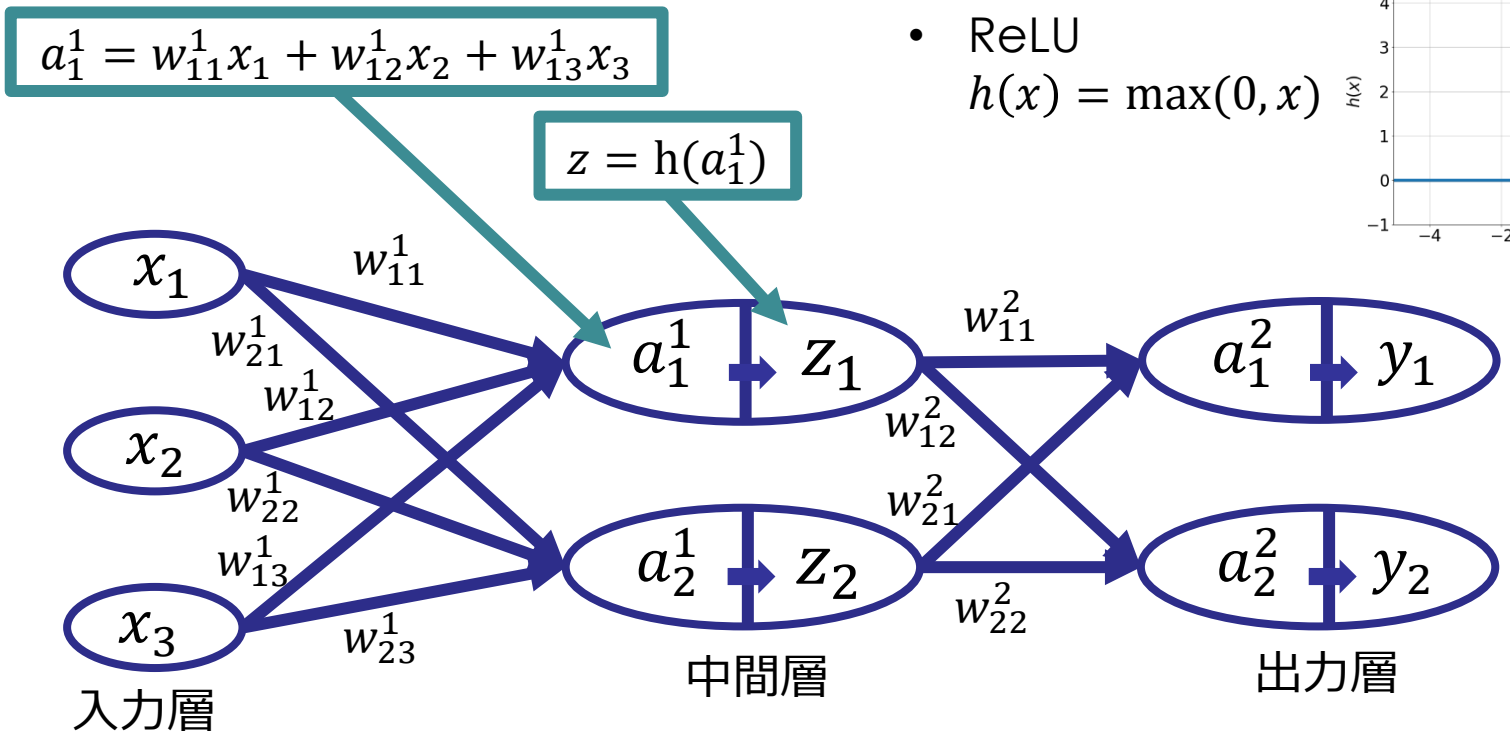
$$h(x) = \frac{1}{1 + e^{-x}}$$



- ReLU

$$h(x) = \max(0, x)$$





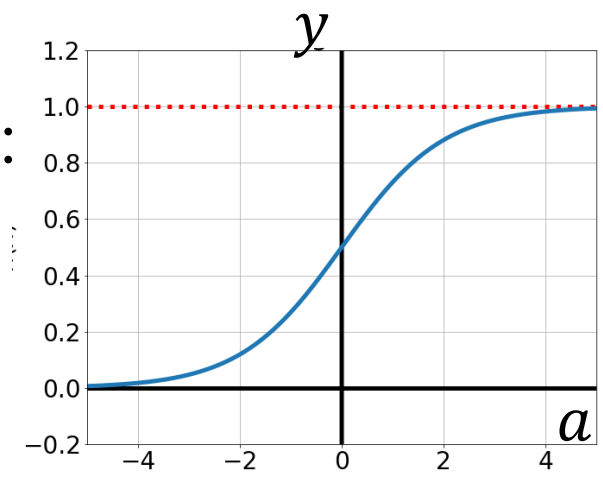
ref. File:Activation logistic.svghttps://commons.wikimedia.org/wiki/File:Activation_logistic.svg
ref. File:Activation rectified linear.svg, CC BY-SA 4.0, https://commons.wikimedia.org/wiki/File:Activation_rectified_linear.svg

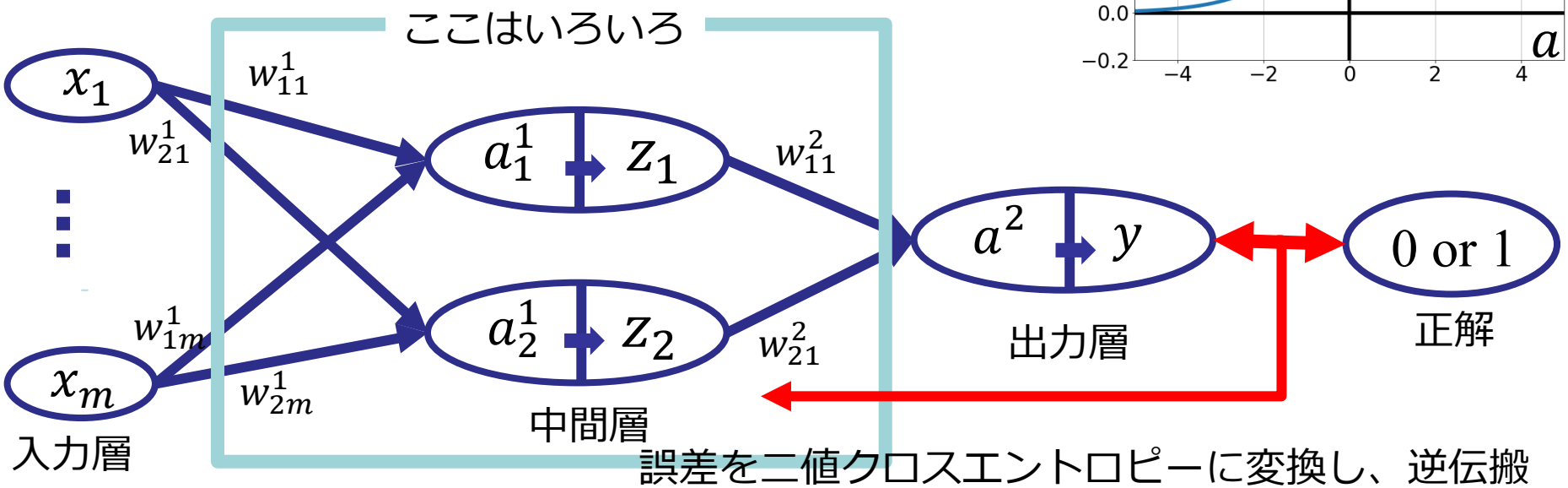
二値分類（正解は0 or 1）

- 出力層のユニット数は１
- 最終層の活性化関数はシグモイド
  - 値が曖昧でも0か1かに割り振ってくれる

sigmoid function:

$$y = \frac{1}{1 + e^{-a}}$$

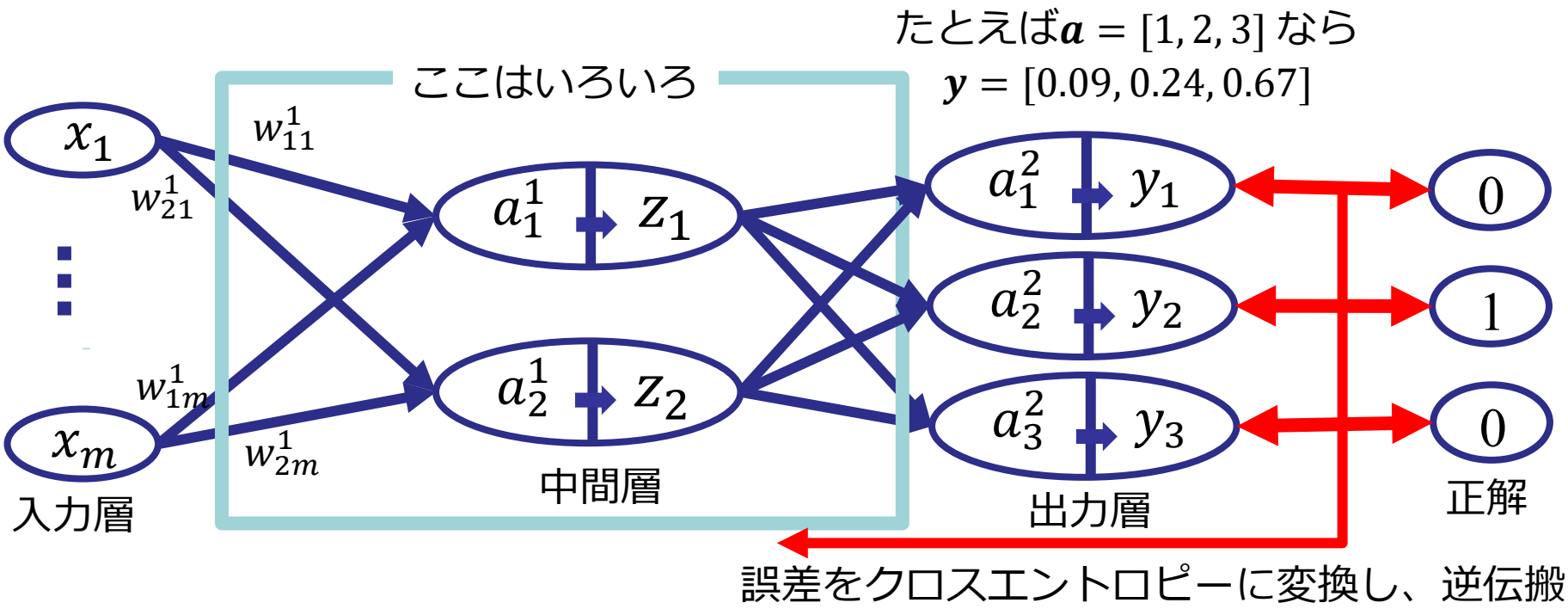




多値分類（正解はone-hot vector）

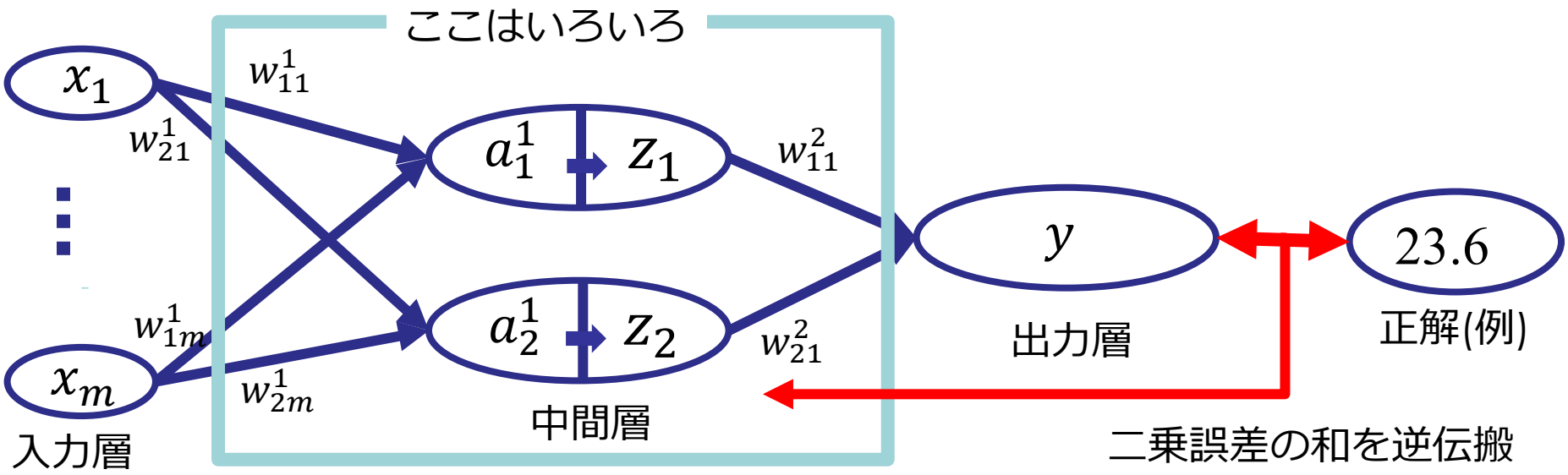
- 出力層のユニット数はクラス数（3クラス分類なら3個）
- 最終層の活性化関数はソフトマックス
  - どんぐりの背比べのとき、差を広げてくれる

softmax function: $y_i = \frac{e^{a_i}}{\sum_{k=1}^{n} e^{a_k}}$



回帰（正解は数値）

- 出力層のユニット数は1
- 最終層の活性化関数はなし



モデルの学習

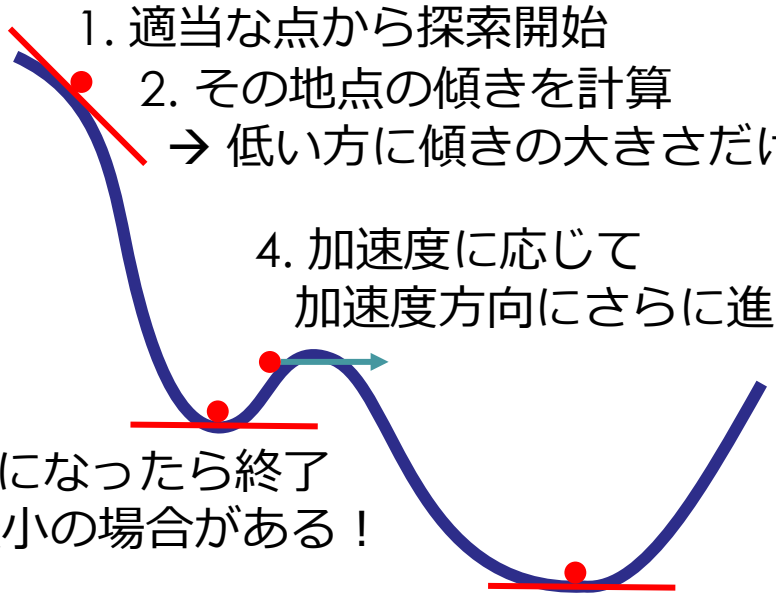
- 重みの初期値はランダムに設定
- 大量のデータを使い、誤差が全体的に小さくなるよう、個々の重みを更新
  - データをミニバッチに分ける（1, 32, 128, 256, 512など）
  - バッチごとに誤差をまとめて逆伝搬
  - すべてのデータを学習し終えて、まだ誤差が大きければ、もう一度すべてのデータを学習する
    - 誤差が小さくなるまで繰り返す
    - 繰り返し回数をエポック（Epoch）と呼ぶ

Q) すべてのデータに対する誤差の和が最小になるような重みのセットをいきなり計算することはできないの？

A) できません。

最小化問題をどう解くか？

- 「yの値が最小となるxを求めよ」
  - 「微分して0になるxを調べる」=解の公式
- 解の公式は常に存在するか？
  - しません。1変数でも4次方程式までしか存在しません。
- 解の公式が存在しない場合、どうやって最小値を導くxを求めるか？
  - 探索的に解きます：最急降下法

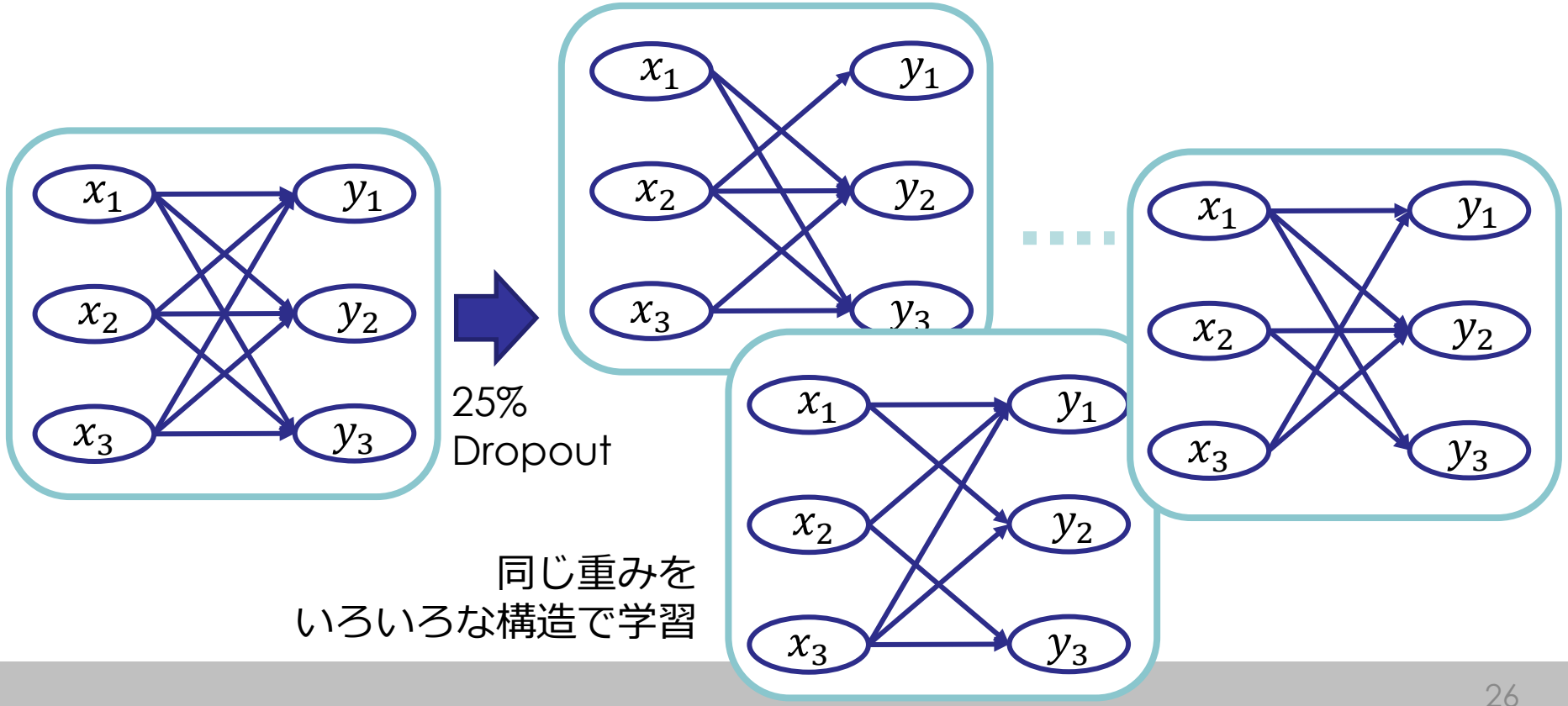


最適化手法

- SDG(Stochastic Gradient Descent; 確率的勾配法)
  - 全データではなく、データをランダムに選んで勾配を計算
  - どれかのデータで極小解に落ちても、別のデータで抜けられる可能性がある
- MomentumSGD
  - 加速度をつけて進むことで、小さな山を越える
- AdaGrad
  - 次元によって勾配が急な次元と緩やかな次元がある
    →勾配が緩やかな次元は学習が進まない
  - 次元ごとに学習率を調整
    （これまで更新量が大きかった次元ほど更新量を小さくする）
- RMSprop: AdaGradの改良版
- Adam: RMSpropの改良版
  - **最もよく使われているアルゴリズム**

汎化性能を高める工夫：Dropout

- 枝をランダムに削除
  - 性能は若干落ちるが、いろんな形のネットワークができる
- アンサンブル学習におけるバギングに似た効果が期待
  - 弱い識別器を複数作り、それらの多数決により最終的な解を決定



バギングがなぜうまくいくのか？

- 株価が上がるか下がるかの二値判別
- ３つのA,B,C識別器は60%の確率で正解を出すとする
  正解が「上がる」のとき「上がる」と予想する確率は？
  1. ３個とも「上がる」と予想：21.6%
     - A正解、B正解、C正解：60% X 60% X 60% = 21.6%
  2. ２個が「上がる」と予想：14.4% x 3 = 43.2%
     - A正解、B正解、C不正解：60% X 60% X 40% = 14.4%
     - A正解、B不正解、C正解：60% X 40% X 60% = 14.4%
     - A不正解、B正解、C正解：40% X 60% X 60% = 14.4%
  3. 1個が「上がる」と予想：9.6 X 3 = 28.8%
     - A正解、B不正解、C不正解：60% X 40% X 40% = 9.6%
     - A不正解、B正解、C不正解：40% X 60% X 40% = 9.6%
     - A不正解、B不正解、C正解：40% X 40% X 60% = 9.6%
  4. ０個が「上がる」と予想
     - A不正解、B不正解、C不正解：40% X 40% X 40% = 6.4%
- 投票で答えを決めるとすると？
  - 「上がる」と予想するのは1.と2.のとき
    → 21.6% + 43.2% = **64.8%** **← 識別器１つ（60%)より精度が高い！**

深層学習による画像認識

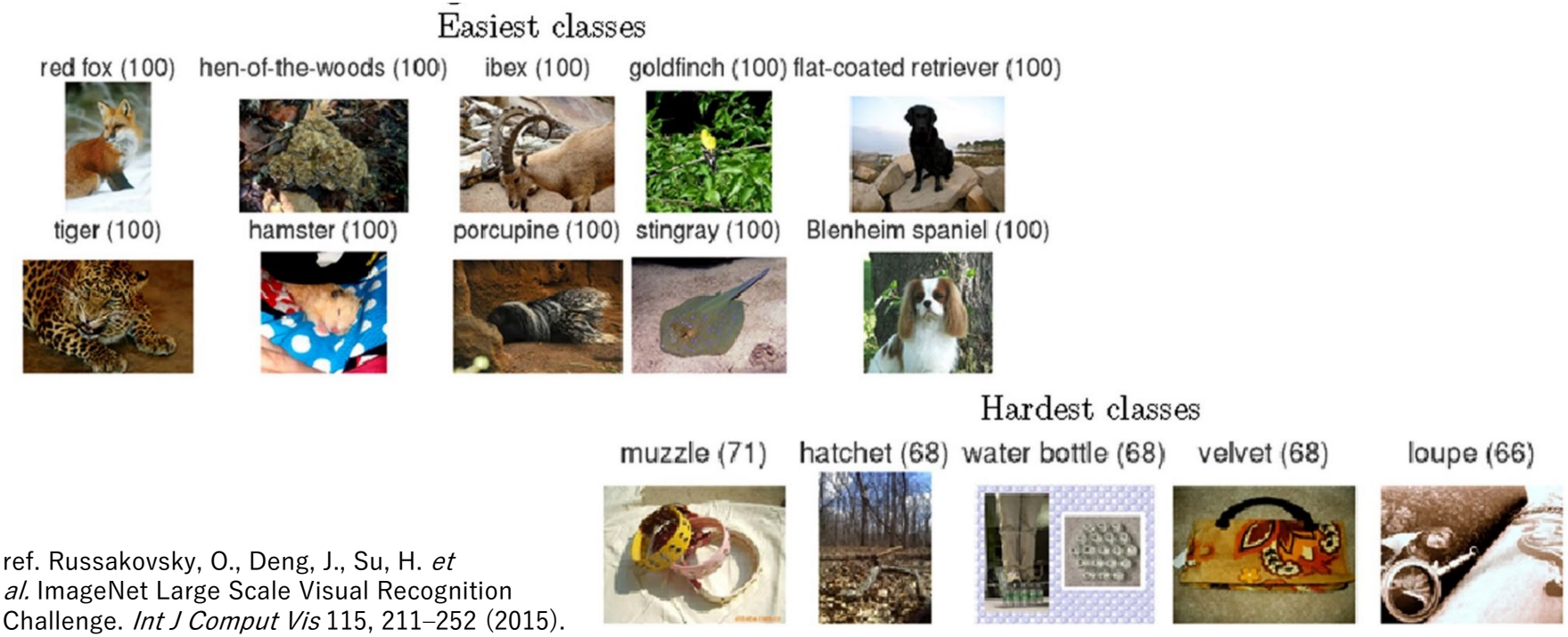
画像認識とは

- 「認識」とは「 人間（主観）が事物（客観・対象）を認め、それとして知るはたらき。(ref. weblio辞書「認識」）
- 画像に写りこんでいる物体や動作、風景等に対し、人間が共通して与えるであろう名前（ラベル）を特定すること
  - 物体の名前を特定する（リンゴが写っている画像を入力すると「リンゴ」というラベルを返す）：物体認識
  - 人間が行っている何らかの動作を特定する（人が手を振っている写真を入力すると「手を振る」というラベルを返す）：動作認識
- 機械学習における「画像認識」とは
  - あらかじめ、認識対象とする各クラスに対し、それと認識されるべき画像の集合が与えられている（訓練データ; training data）
  - データが与えられていないクラスの画像認識は基本的にできない（ただし、未知のクラスをunknownとして認識するタスクもある）
  - どのクラスかわからない画像を適切なクラスに分類するタスクなので、「画像分類 (Image classification)」と表現されることもある
  - 訓練データに含まれない、クラスが未知の画像集合（評価データ; test data）がどれくらい正しく分類できたかで精度を評価

深層学習による画像認識の幕開け

- クラウドソーシングによる大規模データセット：ImageNet
- ImageNet Large-Scale Visual Recognition Challenge (ILSVRC)
  - 1000クラス、学習120万枚、検証5万枚、テスト10万枚
  - マルチラベル：1枚の画像に複数ラベル＋信頼度
- 2011年のILSVRCで優勝したモデルのエラー率は26%

→ 2012年にDeep Learningを使ったモデルが登場→15%に激減！



ref. Russakovsky, O., Deng, J., Su, H. *et al.* ImageNet Large Scale Visual Recognition Challenge. *Int J Comput Vis* 115, 211–252 (2015).

どんな問題を解いたのか？ネコ科大型動物編

これは何ですか？

Photo by Patrick Giraud from Wikipedia CC BY 2.5
https://ja.wikipedia.org/wiki/%E3%83%92%E3%83%A7%E3%82%A6#/media/%E3%83%95%E3%82%A1%E3%82%A4%E3%83%AB:Namibie_Etosha_Leopard_01edit.jpg

ヒョウ？ジャガー？

著作権等の都合上、ここに挿入されていた画像を削除しました。

ヒョウとジャガー　体の模様比較写真

https://ikimono-matome.com/leopard-jaguar/

模様から、これはヒョウです！

http://image-net.org/challenges/LSVRC/2012/browse-synsets

どんな問題を解いたのか？犬編

これは何ですか？

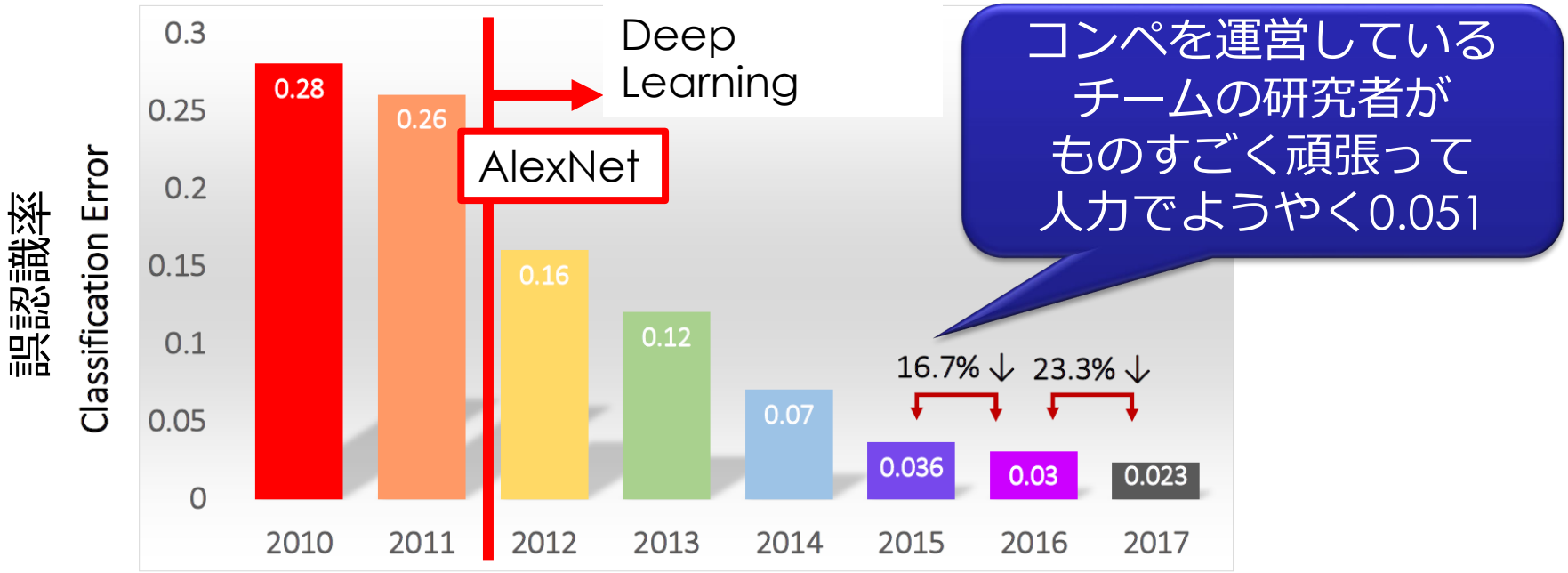
ダルメシアン、ニューファンドランドドッグ、シベリアンハスキー、ハンティングドッグ、ジャーマンシェパード、イングリッシュシープドッグ、フレンチブルドッグ、マルチーズ、グレイトマウンテンドッグ、パグ、テリア、シェットランドシープドッグ

Photo by Harald Urnes, Norway from Wikipedia CC BY-SA 3.0
https://ja.wikipedia.org/wiki/%E3%82%AA%E3%83%BC%E3%83%AB%E3%83%89%E3%83%BB%E3%82%A4%E3%83%B3%E3%82%B0%E3%83%AA%E3%83%83%E3%82%B7%E3%83%A5%E3%83%BB%E3%82%B7%E3%83%BC%E3%83%97%E3%83%89%E3%83%83%E3%82%B0#/media/%E3%83%95%E3%82%A1%E3%82%A4%E3%83%AB:Old_english_sheepdog_Ch_Bobbyclown's_Dare_for_More.jpg

http://image-net.org/challenges/LSVRC/2012/browse-synsets

物体認識精度は人間を超えた!?

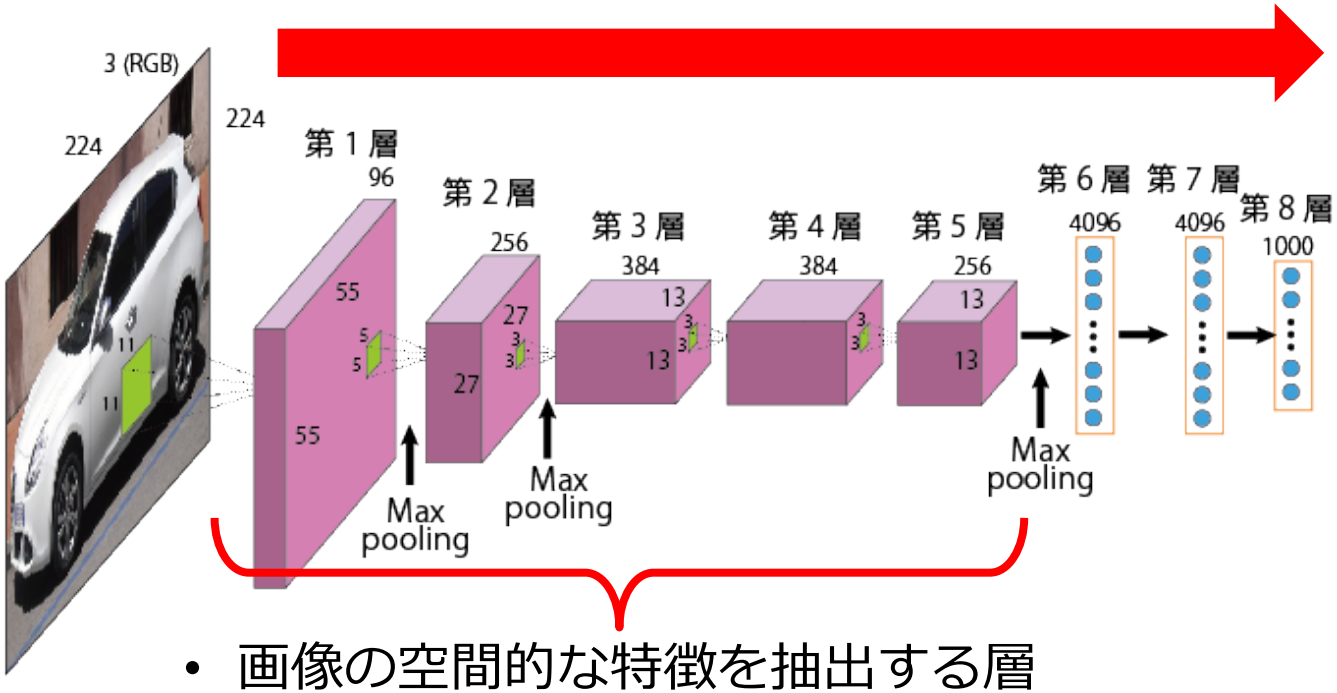
- ILSVRC2012で、トロント大学Geoffrey Hinton教授率いるグループが初めて多層ニューラルネットワークによる画像認識モデルを提案
- 主著者の名前をもじってAlexNetと呼ばれる
- 翌年から上位チームはすべてDeep Learningを採用
- 2015年についに人間の精度を超えた



Excerpted from ImageNet Large Scale Visual Recognition Challenge (ILSVRC) 2017 Overview
Andrej Karpathy, "What I learned from competing against a ConvNet on ImageNet", Sep 2, 2014, http://karpathy.github.io/2014/09/02/what-i-learned-from-competing-against-a-convnet-on-imagenet/

畳み込みニューラルネットワーク（CNN; Convolutional Neural Network)

- 代表的な物体識別モデル
- 実際には様々な実装がある（下図はAlexNet）



- 画像の空間的な特徴を抽出する層
- 途中の層でははどうなっているか？

最終層にsoftmaxを適用した結果

0.00 goldfish
0.00 great white shark
0.00 tiger shark
0.00 hammerhead
0.01 electric ray
...
0.89 car
...
0.00 stinkhorn
0.00 earthstar
0.00 hen-of-the-woods
0.02 bolete
0.00 ear, spike
0.00 toilet tissue

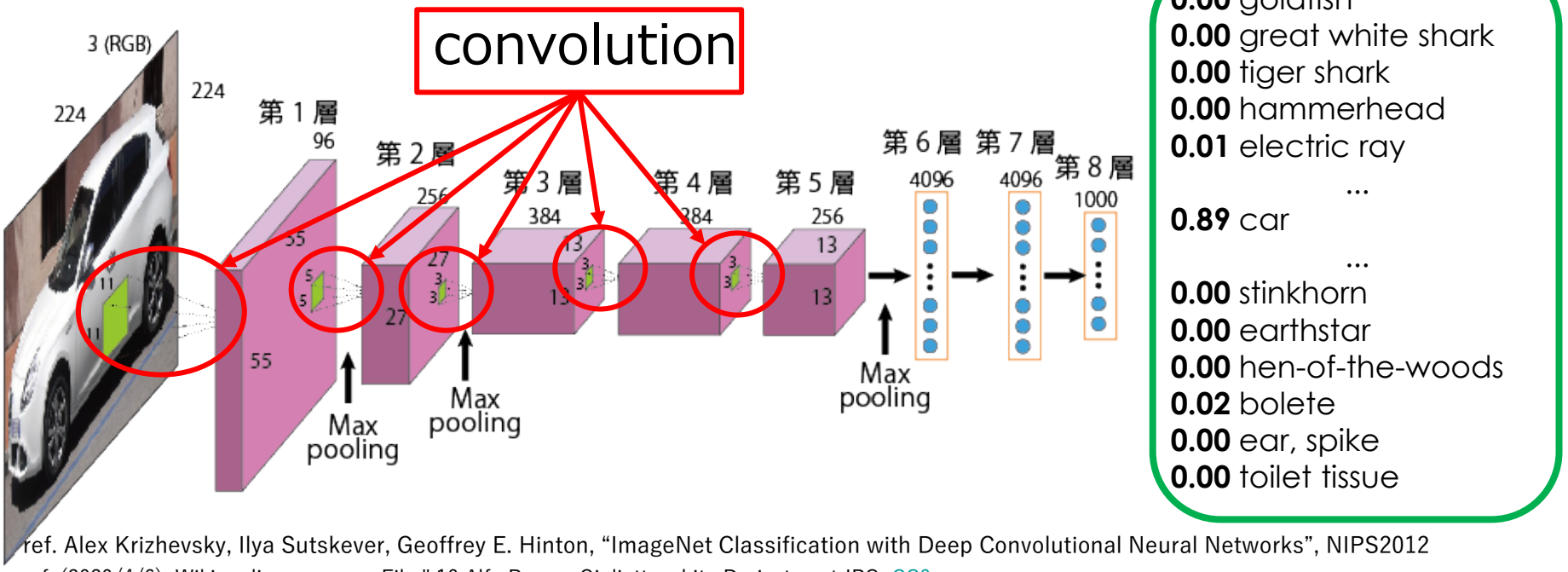
1000クラスのうちcarだけが高く、残りがほぼ0のようなベクトルが出力

ref. Alex Krizhevsky, Ilya Sutskever, Geoffrey E. Hinton, “ImageNet Classification with Deep Convolutional Neural Networks”, NIPS2012

ref. (2020/4/6): Wikimedia commons: File:" 10 Alfa Romeo Giulietta white Derivate cut.JPG CC0
https://commons.wikimedia.org/wiki/Category:CC-Zero?uselang=ja#/media/File:%22_10_Alfa_Romeo_Giulietta_white_Derivate_cut.JPG

畳み込みニューラルネットワークにおける畳み込み演算

- 第1層から第5層までは、2次元デジタルフィルタで説明した**畳み込み演算 (convolution)**を行っている
- ただし、2次元デジタルフィルタではフィルタは人がデザインしていたのに対し、CNNではデータから学習により取得する

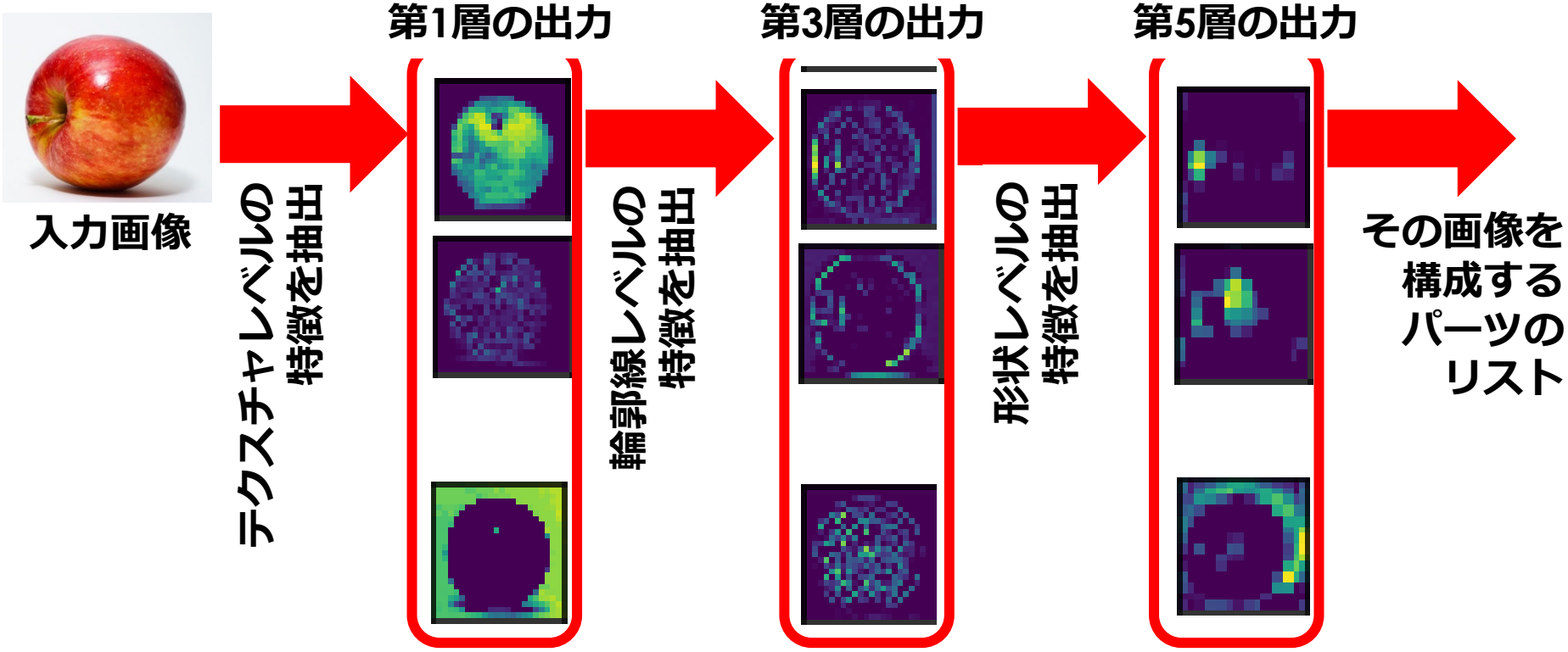


ref. Alex Krizhevsky, Ilya Sutskever, Geoffrey E. Hinton, “ImageNet Classification with Deep Convolutional Neural Networks”, NIPS2012

ref. (2020/4/6): Wikimedia commons: File:" 10 Alfa Romeo Giulietta white Derivate cut.JPG CC0
https://commons.wikimedia.org/wiki/Category:CC-Zero?uselang=ja#/media/File:%22_10_Alfa_Romeo_Giulietta_white_Derivate_cut.JPG

CNNにおける画像のフィルタリング

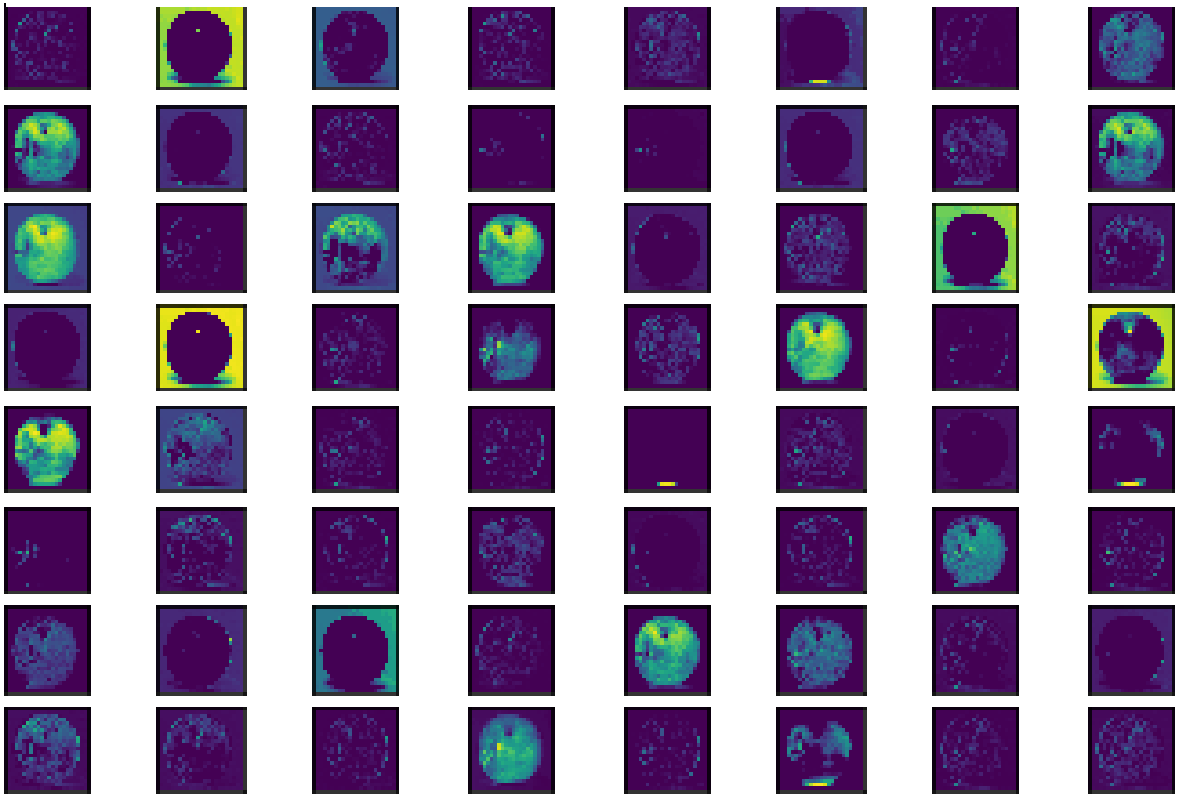
- 層を経るごとにより具体的な形状の特徴を取り出していく
- 第5層になると、「入力画像にどんなパーツが写りこんでいるか（実際には尤度分布）」がわかってくる
- 「入力画像に写っているパーツのセット」が「他のクラスに比べ、車にありがちなパーツのセット」であるならば「車」と判別



ref. (2020/4/3): Wikimedia commons: File:Red Apple.jpg CC BY 2.0 https://commons.wikimedia.org/wiki/File:Red_Apple.jpg

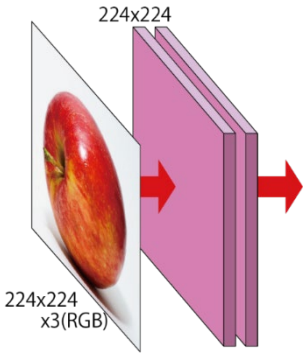
モデルはKeras VGG16 pretrainedを使用

第1層の出力(テクスチャレベルの特徴)



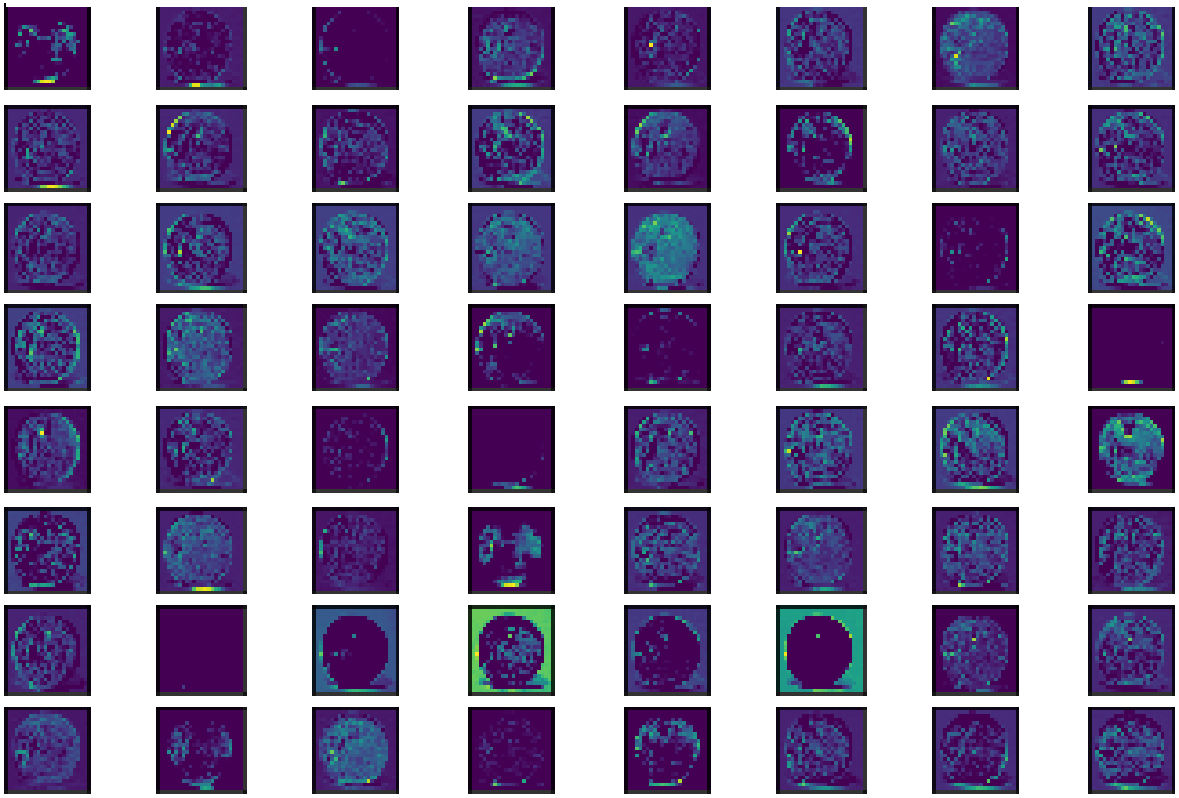
ref. (2020/4/3): Wikimedia commons: File:Red Apple.jpg CC BY 2.0 https://commons.wikimedia.org/wiki/File:Red_Apple.jpg

入力画像



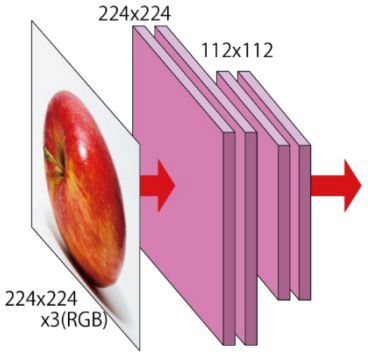
可視化している層

第2層の出力(テクスチャレベルの特徴)



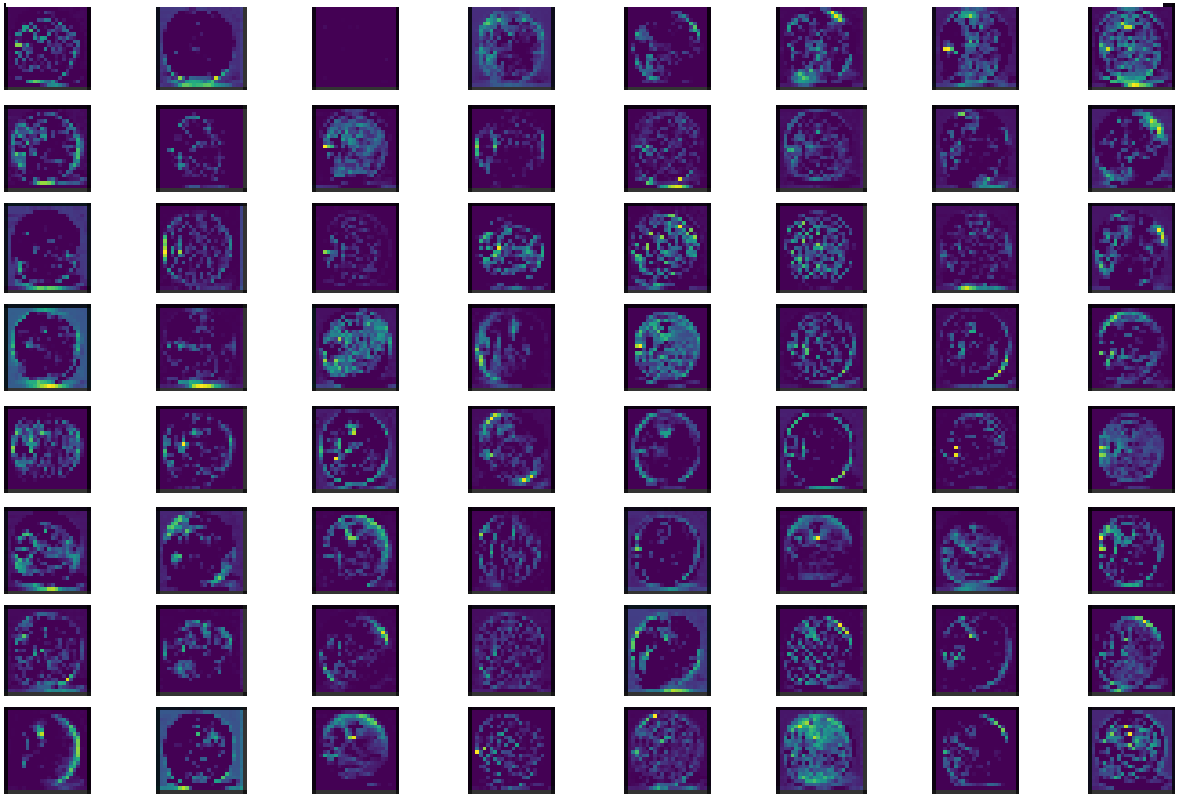
ref. (2020/4/3): Wikimedia commons: File:Red Apple.jpg CC BY 2.0 https://commons.wikimedia.org/wiki/File:Red_Apple.jpg

入力画像



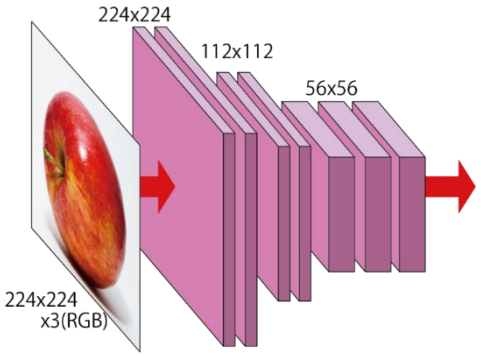
可視化している層

第3層の出力(輪郭線レベルの特徴)



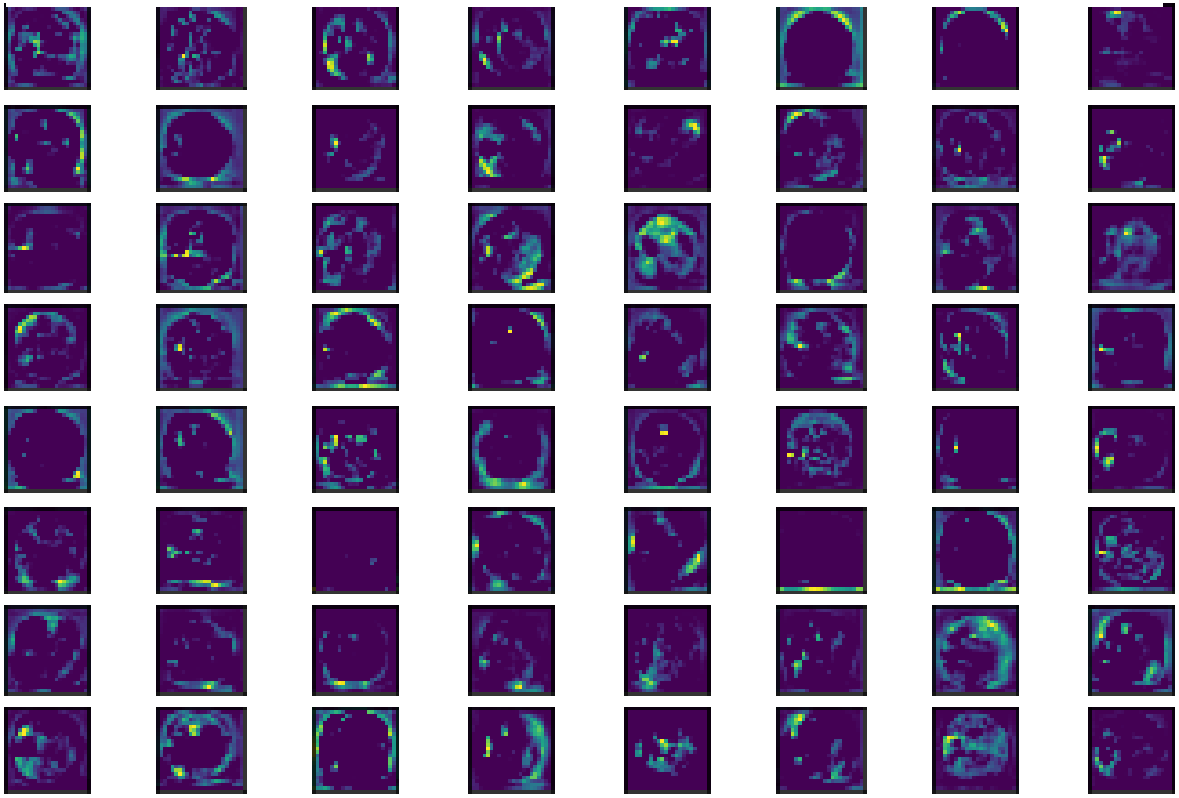
ref. (2020/4/3): Wikimedia commons: File:Red Apple.jpg CC BY 2.0 https://commons.wikimedia.org/wiki/File:Red_Apple.jpg

入力画像



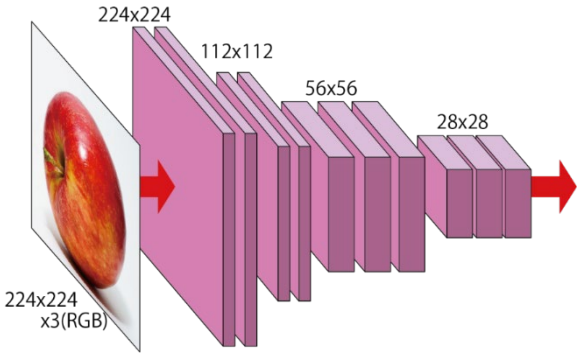
可視化している層

第4層の出力(輪郭線レベルの特徴)



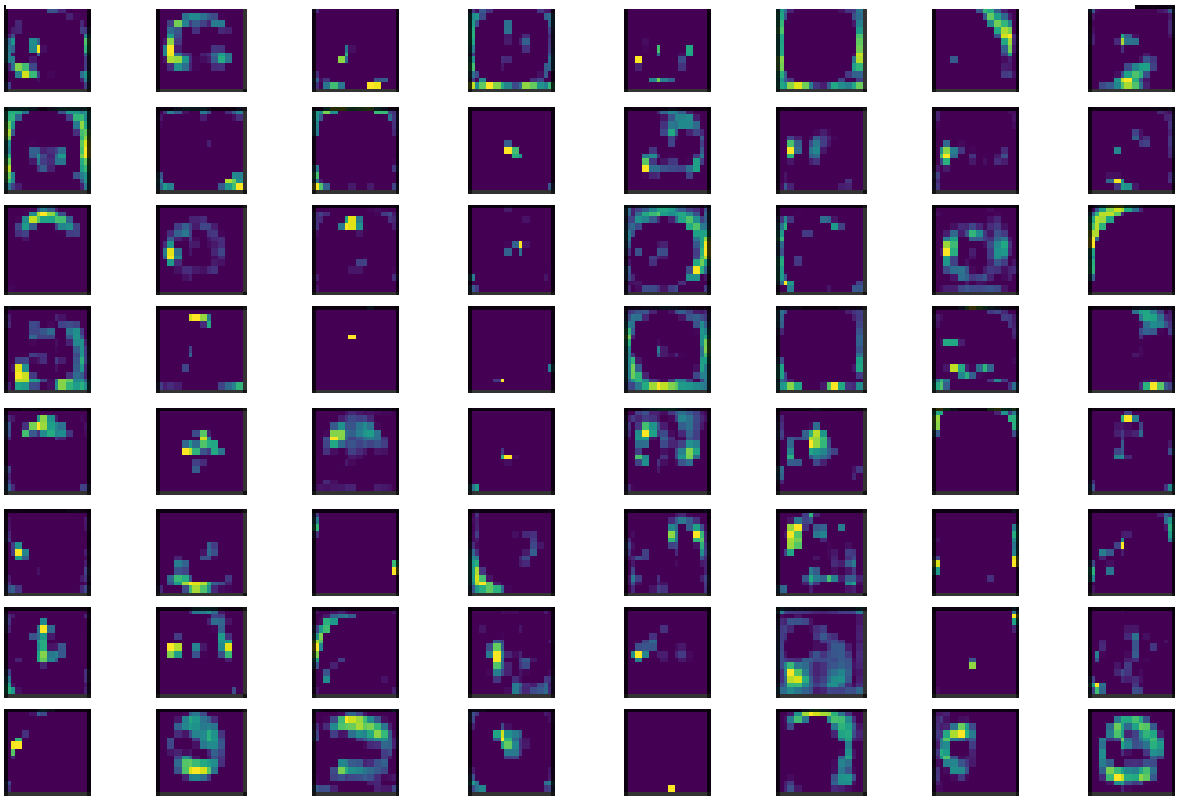
ref. (2020/4/3): Wikimedia commons: File:Red Apple.jpg CC BY 2.0 https://commons.wikimedia.org/wiki/File:Red_Apple.jpg

入力画像



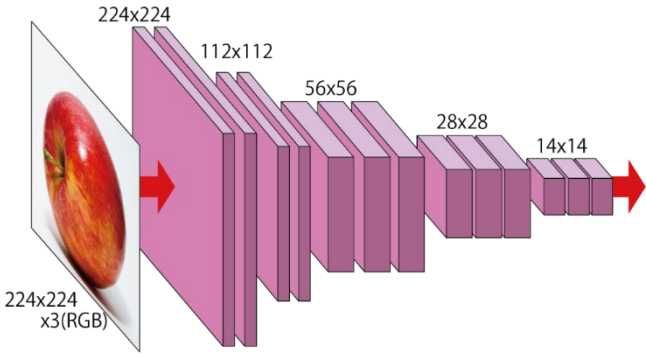
可視化している層

第5層の出力(形状レベルの特徴)



ref. (2020/4/3): Wikimedia commons: File:Red Apple.jpg CC BY 2.0 https://commons.wikimedia.org/wiki/File:Red_Apple.jpg

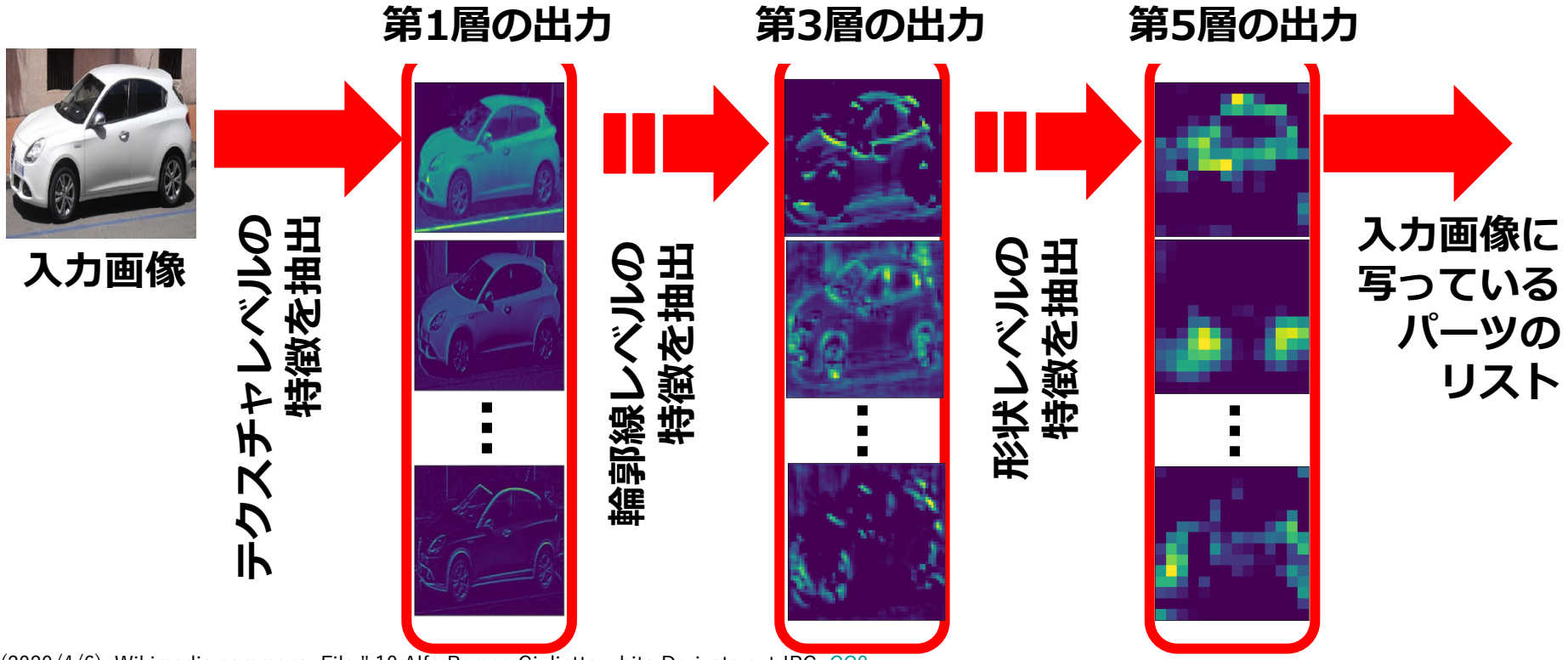
入力画像



可視化している層

CNNにおける画像のフィルタリング

- 層を経るごとにより具体的な形状の特徴を取り出していく
- 第5層になると、「入力画像にどんなパーツが写りこんでいるか（実際には尤度分布）」がわかってくる
- 「入力画像に写っているパーツのセット」が「他のクラスに比べ、車にありがちなパーツのセット」であるならば「車」と判別

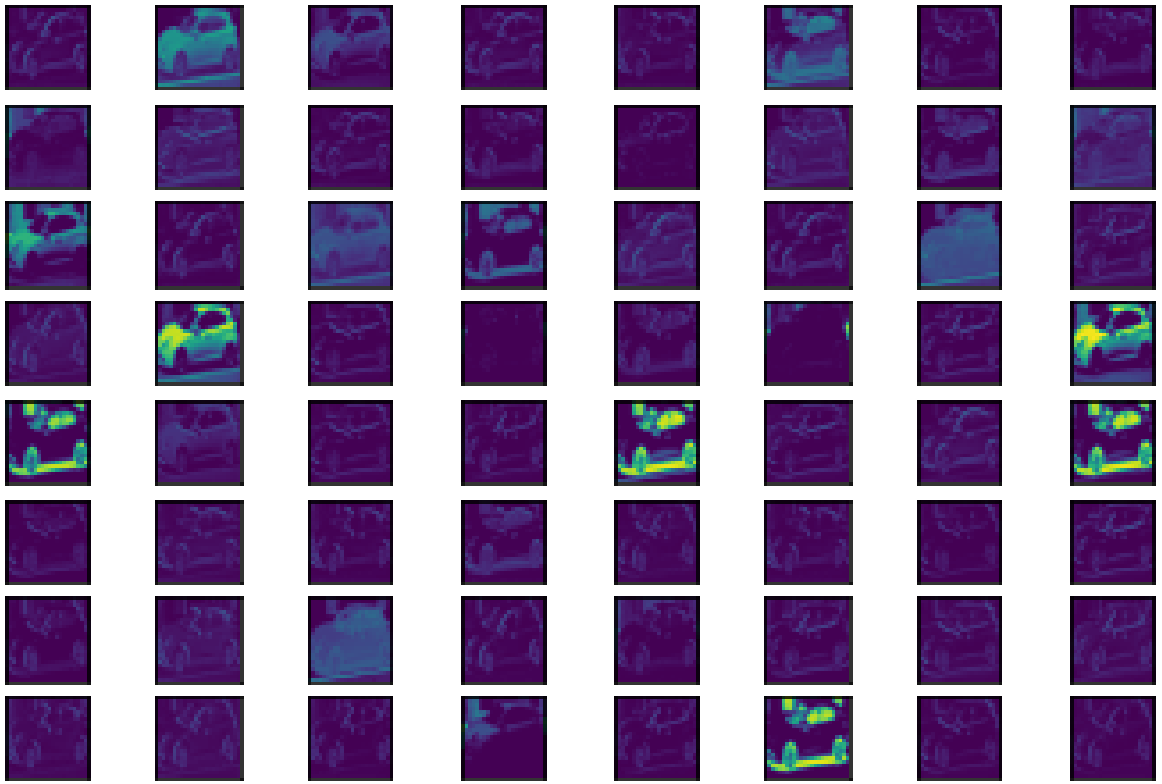


ref. (2020/4/6): Wikimedia commons: File:" 10 Alfa Romeo Giulietta white Derivate cut.JPG CC0 https://commons.wikimedia.org/wiki/Category:CC-Zero?uselang=ja#/media/File:%22_10_Alfa_Romeo_Giulietta_white_Derivate_cut.JPG

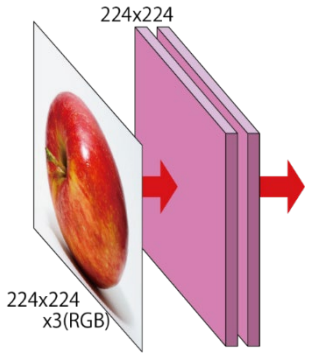
モデルはKeras VGG16 pretrainedを使用

ref. (2020/4/6): Wikimedia commons: File:" 10 Alfa Romeo Giulietta white Derivate cut.JPG CC0
https://commons.wikimedia.org/wiki/Category:CC-Zero?uselang=ja#/media/File:%22_10_Alfa_Romeo_Giulietta_white_Derivate_cut.JPG

第1層の出力（テクスチャレベルの特徴）



入力画像

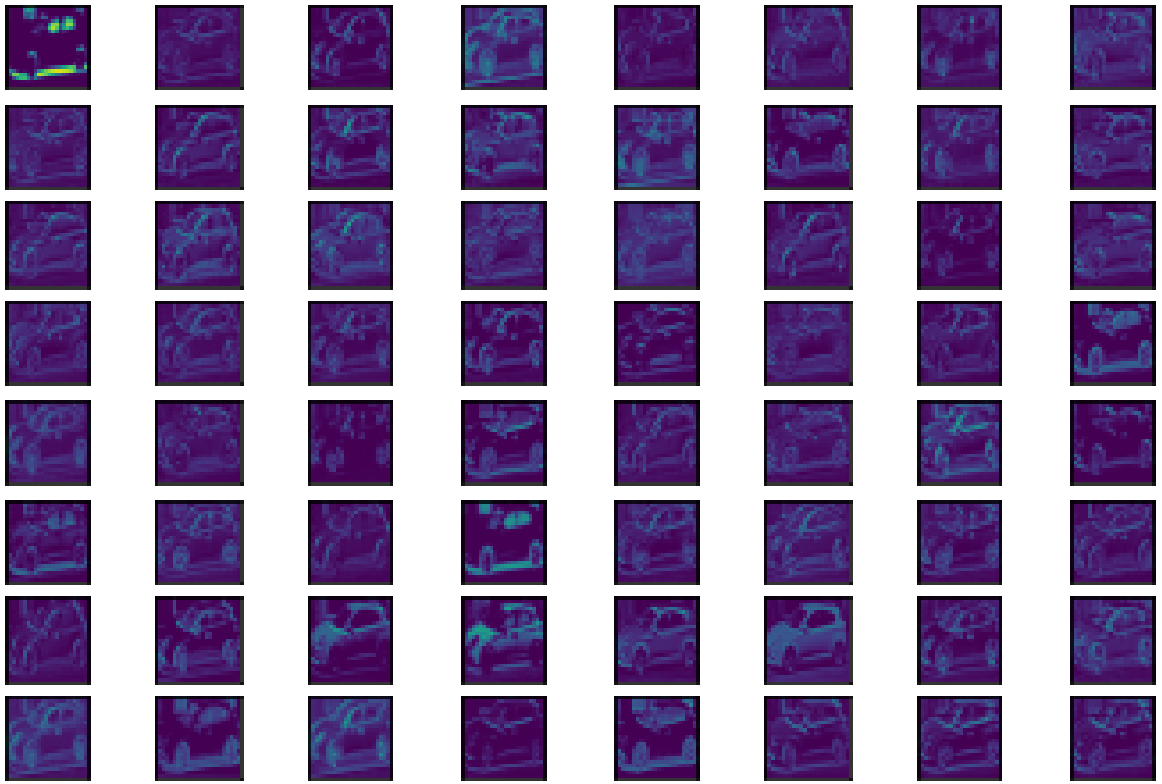


可視化している層

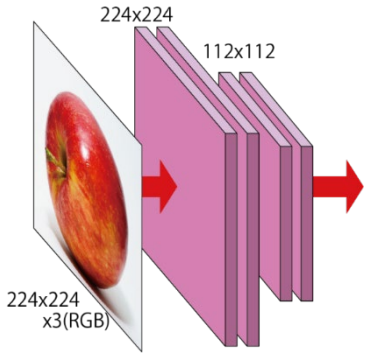
ref. (2020/4/3): Wikimedia commons: File:Red Apple.jpg CC BY 2.0
https://commons.wikimedia.org/wiki/File:Red_Apple.jpg

ref. (2020/4/6): Wikimedia commons: File:" 10 Alfa Romeo Giulietta white Derivate cut.JPG CC0
https://commons.wikimedia.org/wiki/Category:CC-Zero?uselang=ja#/media/File:%22_10_Alfa_Romeo_Giulietta_white_Derivate_cut.JPG

第2層の出力（テクスチャレベルの特徴）



入力画像



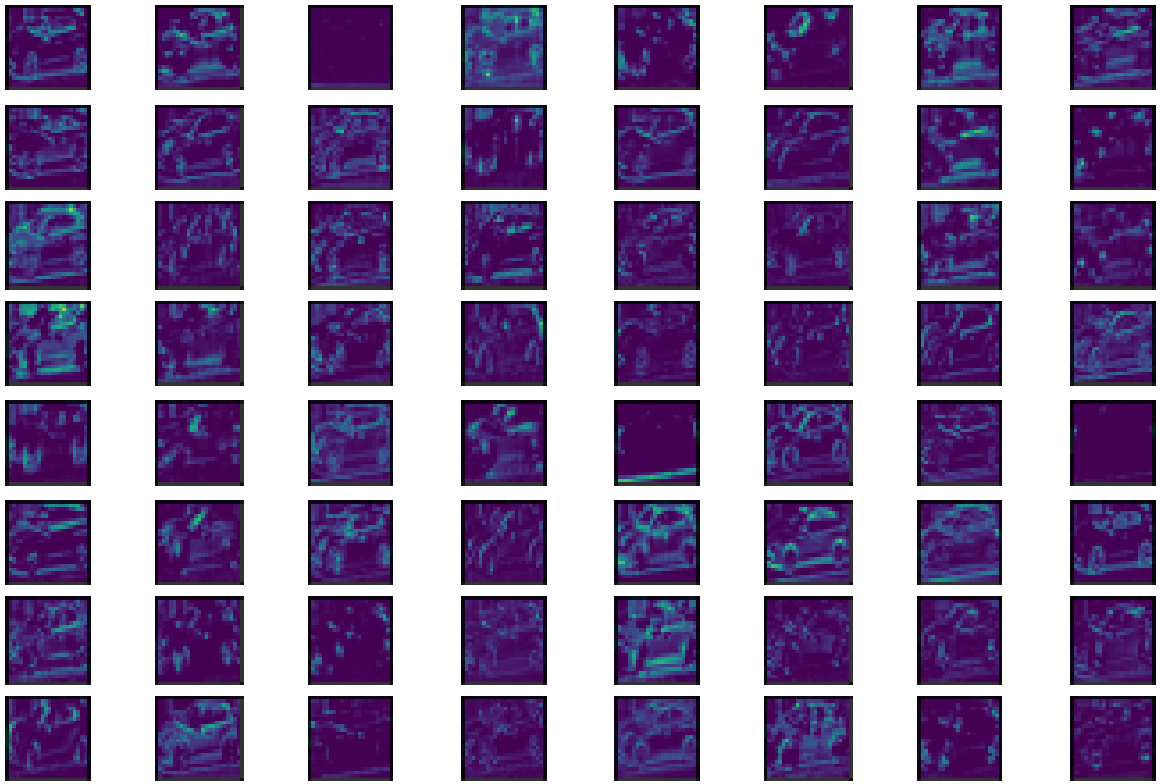
可視化している層

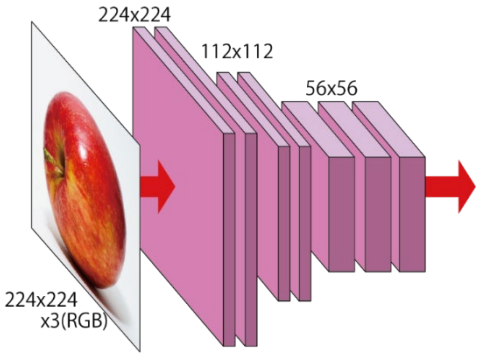
ref. (2020/4/3): Wikimedia commons: File:Red Apple.jpg CC BY 2.0
https://commons.wikimedia.org/wiki/File:Red_Apple.jpg

ref. (2020/4/6): Wikimedia commons: File:" 10 Alfa Romeo Giulietta white Derivate cut.JPG CC0
https://commons.wikimedia.org/wiki/Category:CC-Zero?uselang=ja#/media/File:%22_10_Alfa_Romeo_Giulietta_white_Derivate_cut.JPG

入力画像

第3層の出力（輪郭線レベルの特徴）

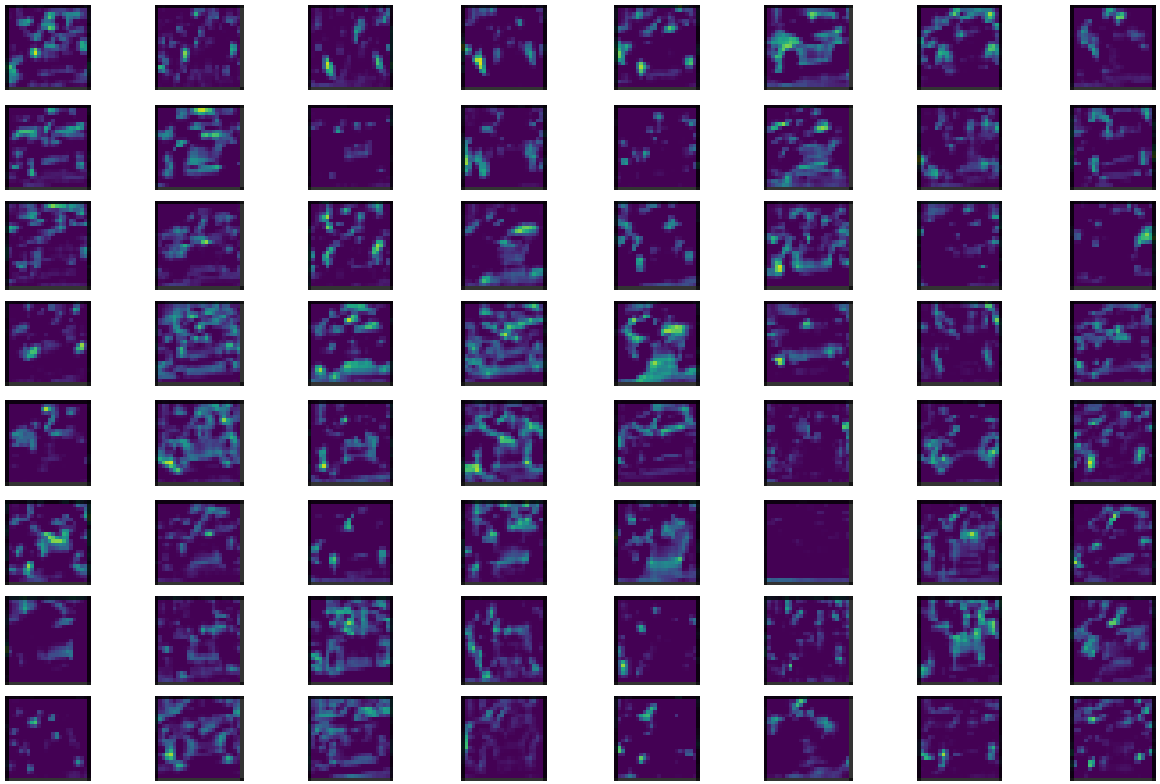




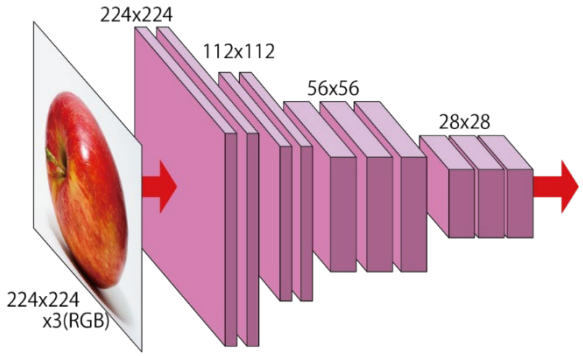
ref. (2020/4/3): Wikimedia commons: File:Red Apple.jpg CC BY 2.0
https://commons.wikimedia.org/wiki/File:Red_Apple.jpg

ref. (2020/4/6): Wikimedia commons: File:" 10 Alfa Romeo Giulietta white Derivate cut.JPG CC0
https://commons.wikimedia.org/wiki/Category:CC-Zero?uselang=ja#/media/File:%22_10_Alfa_Romeo_Giulietta_white_Derivate_cut.JPG

第4層の出力（輪郭線レベルの特徴）



入力画像

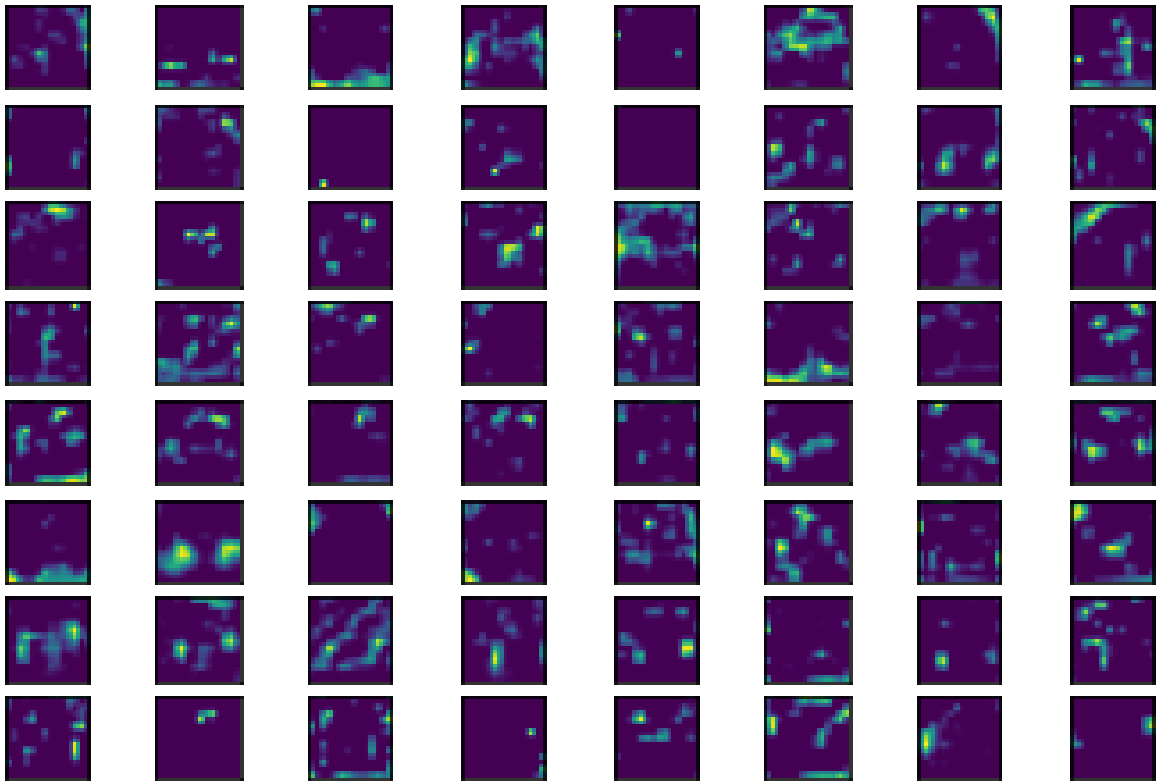


可視化している層

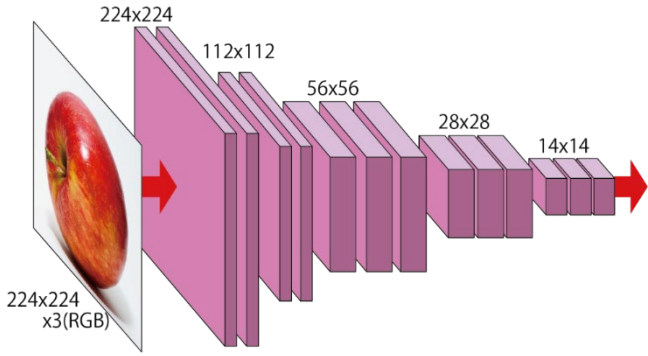
ref. (2020/4/3): Wikimedia commons: File:Red Apple.jpg CC BY 2.0
https://commons.wikimedia.org/wiki/File:Red_Apple.jpg

ref. (2020/4/6): Wikimedia commons: File:" 10 Alfa Romeo Giulietta white Derivate cut.JPG CC0
https://commons.wikimedia.org/wiki/Category:CC-Zero?uselang=ja#/media/File:%22_10_Alfa_Romeo_Giulietta_white_Derivate_cut.JPG

第5層の出力（形状レベルの特徴）



入力画像

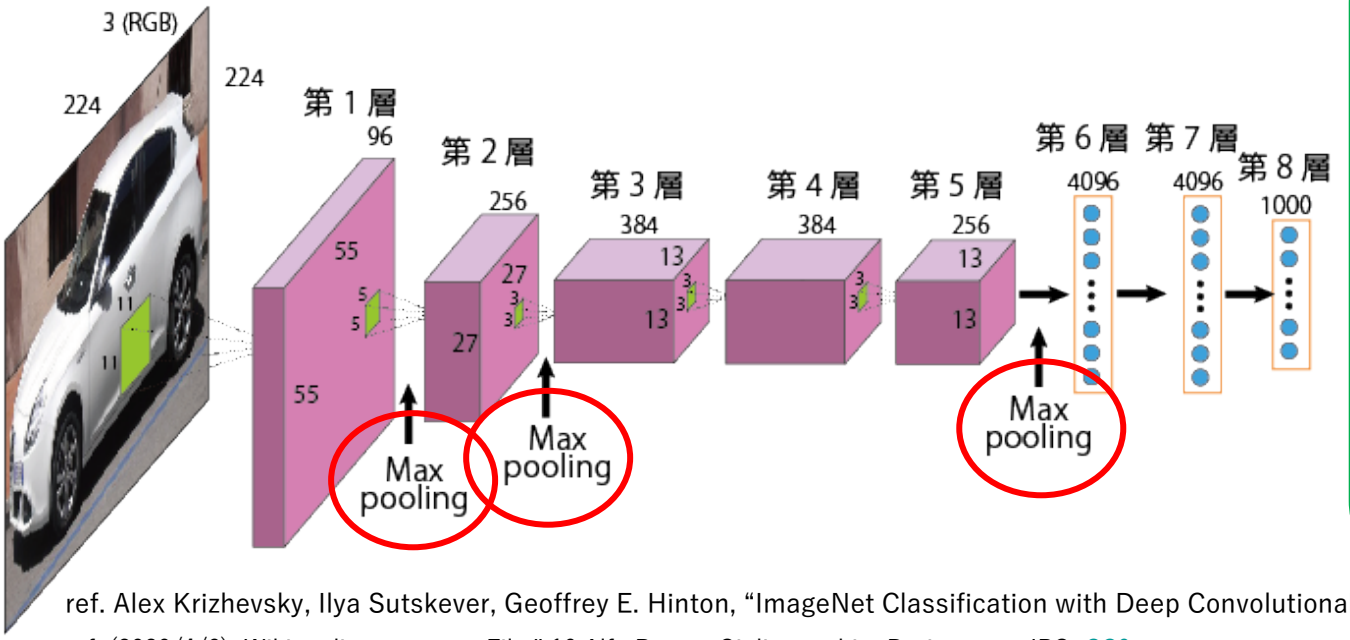


可視化している層

ref. (2020/4/3): Wikimedia commons: File:Red Apple.jpg CC BY 2.0
https://commons.wikimedia.org/wiki/File:Red_Apple.jpg

畳み込みニューラルネットワークにおけるプーリング (pooling)

- 画面を小さく区切り、各区間ごとに画素をまとめて1つの値にする
  - データのサイズを小さくする役割
  - 物体が写っている位置や角度の違いに頑健にする役割
- 最大値を取る場合 max pooling、平均値をとる場合 average pooling と呼ぶ



0.00 goldfish
0.00 great white shark
0.00 tiger shark
0.00 hammerhead
0.01 electric ray
...
0.89 car
...
0.00 stinkhorn
0.00 earthstar
0.00 hen-of-the-woods
0.02 bolete
0.00 ear, spike
0.00 toilet tissue

ref. Alex Krizhevsky, Ilya Sutskever, Geoffrey E. Hinton, “ImageNet Classification with Deep Convolutional Neural Networks”, NIPS2012

ref. (2020/4/6): Wikimedia commons: File:" 10 Alfa Romeo Giulietta white Derivate cut.JPG CC0
https://commons.wikimedia.org/wiki/Category:CC-Zero?uselang=ja#/media/File:%22_10_Alfa_Romeo_Giulietta_white_Derivate_cut.JPG

max pooling / average pooling

1	0	3	1
2	4	1	2
1	0	2	3
4	6	6	8

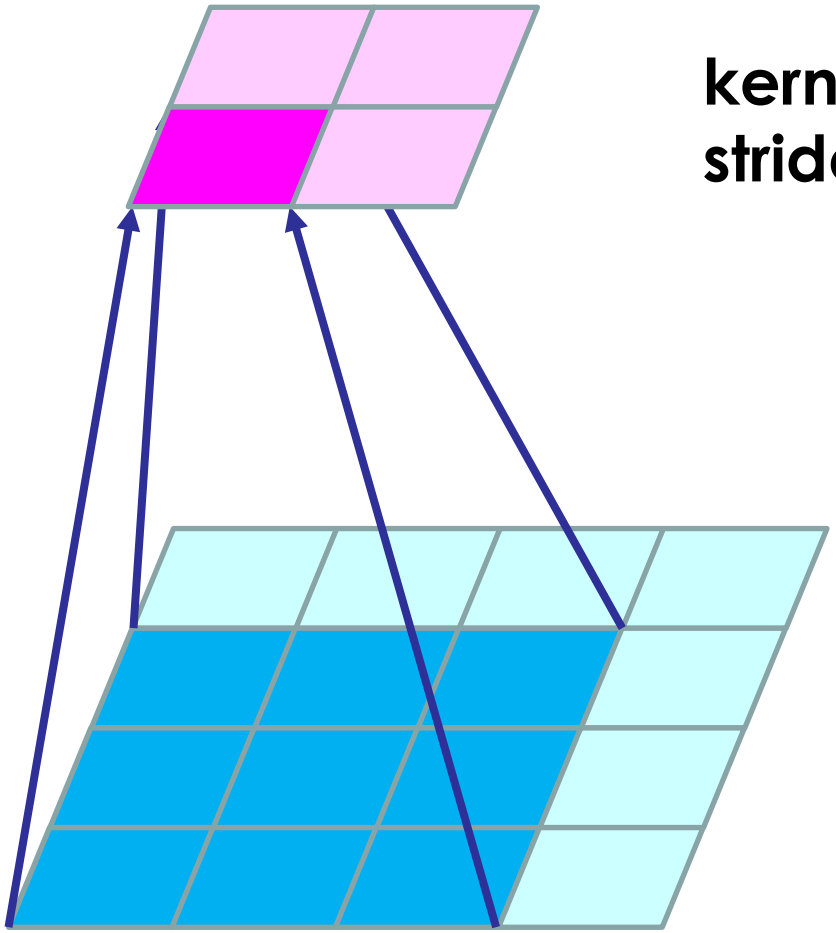
2x2 Max pooling

4	3
6	8

2x2 Average pooling

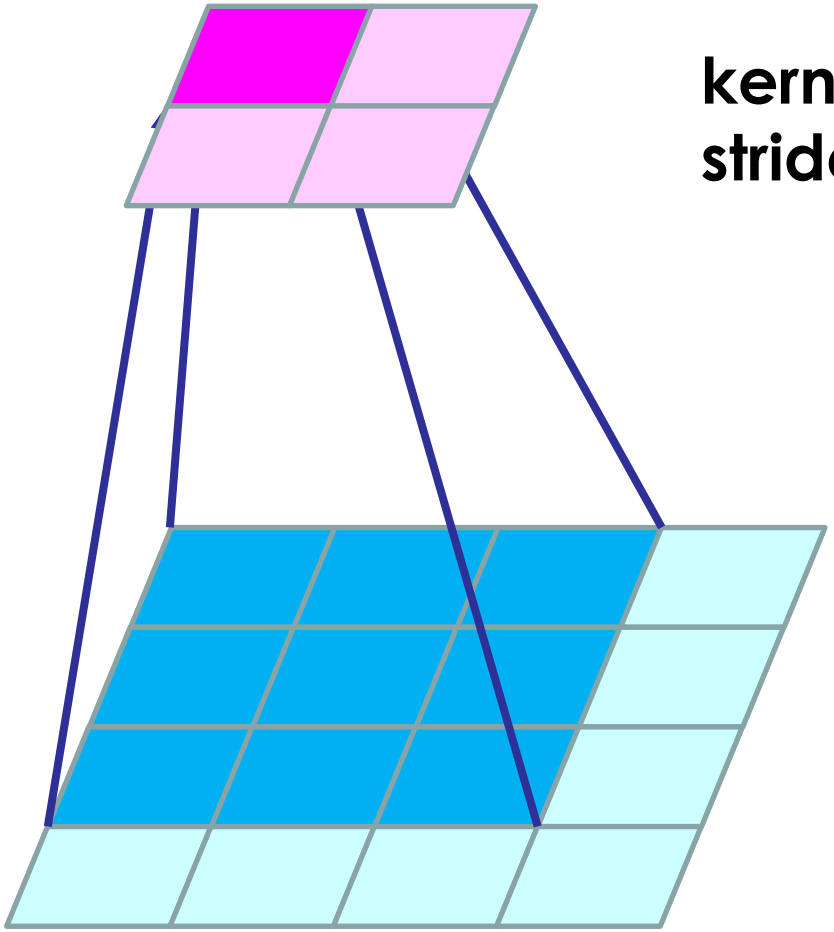
1.75	1.75
2.75	4.75

pooling size & stride



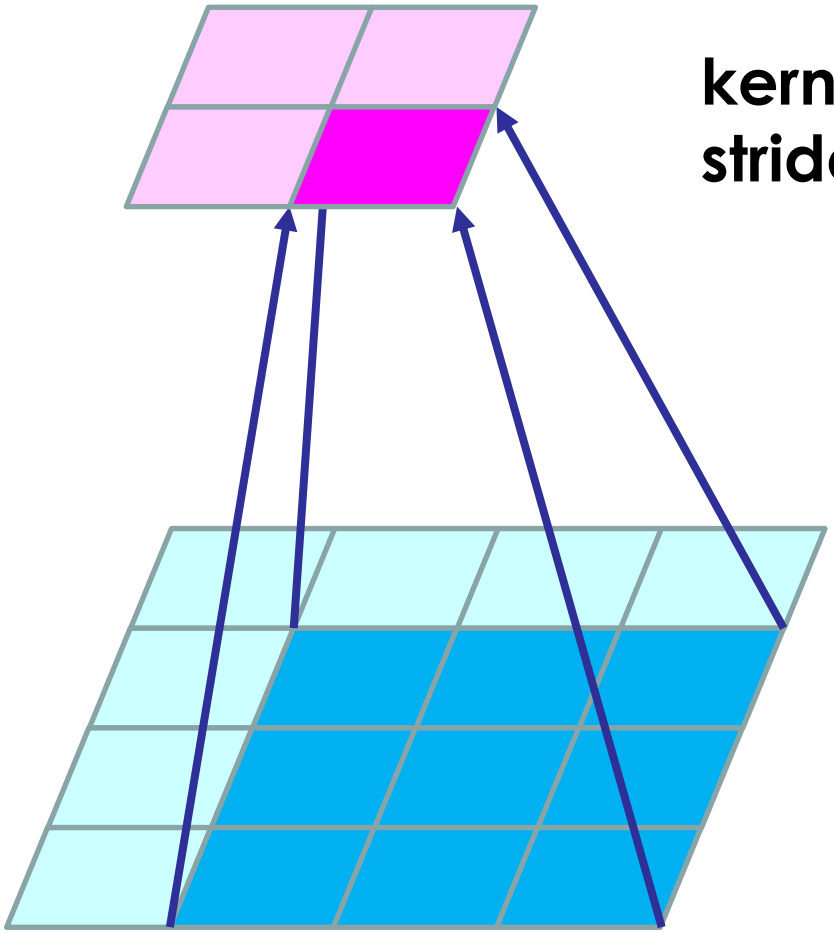
kernel size: 3x3
stride: 1

pooling size & stride



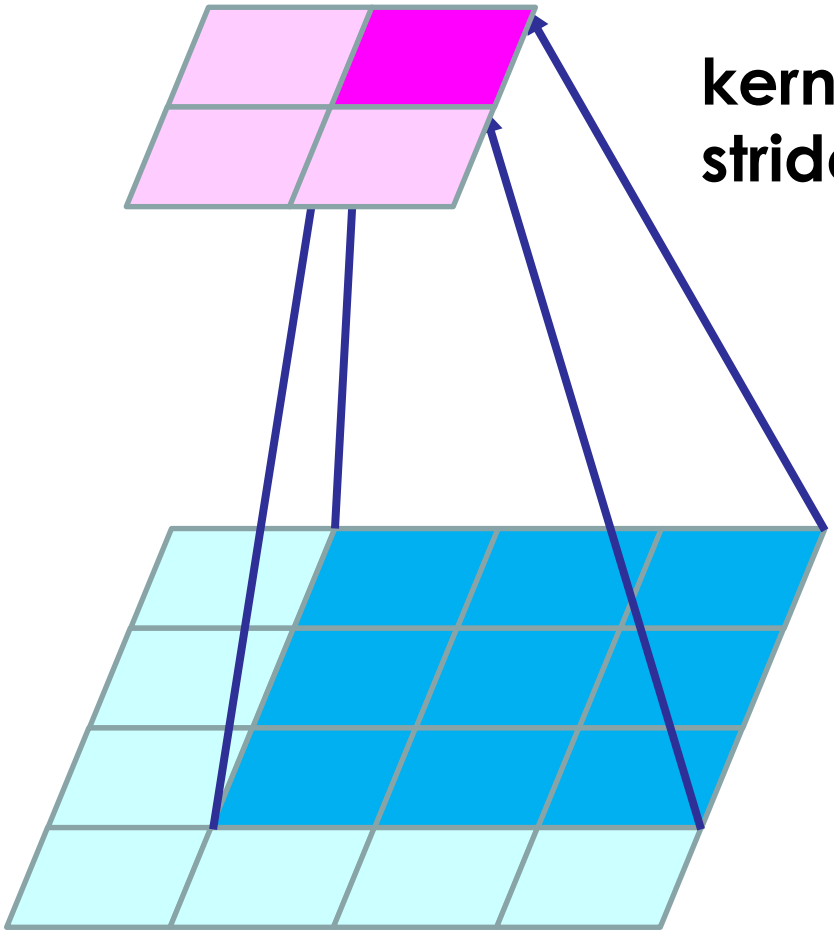
kernel size: 3x3
stride: 1

pooling size & stride



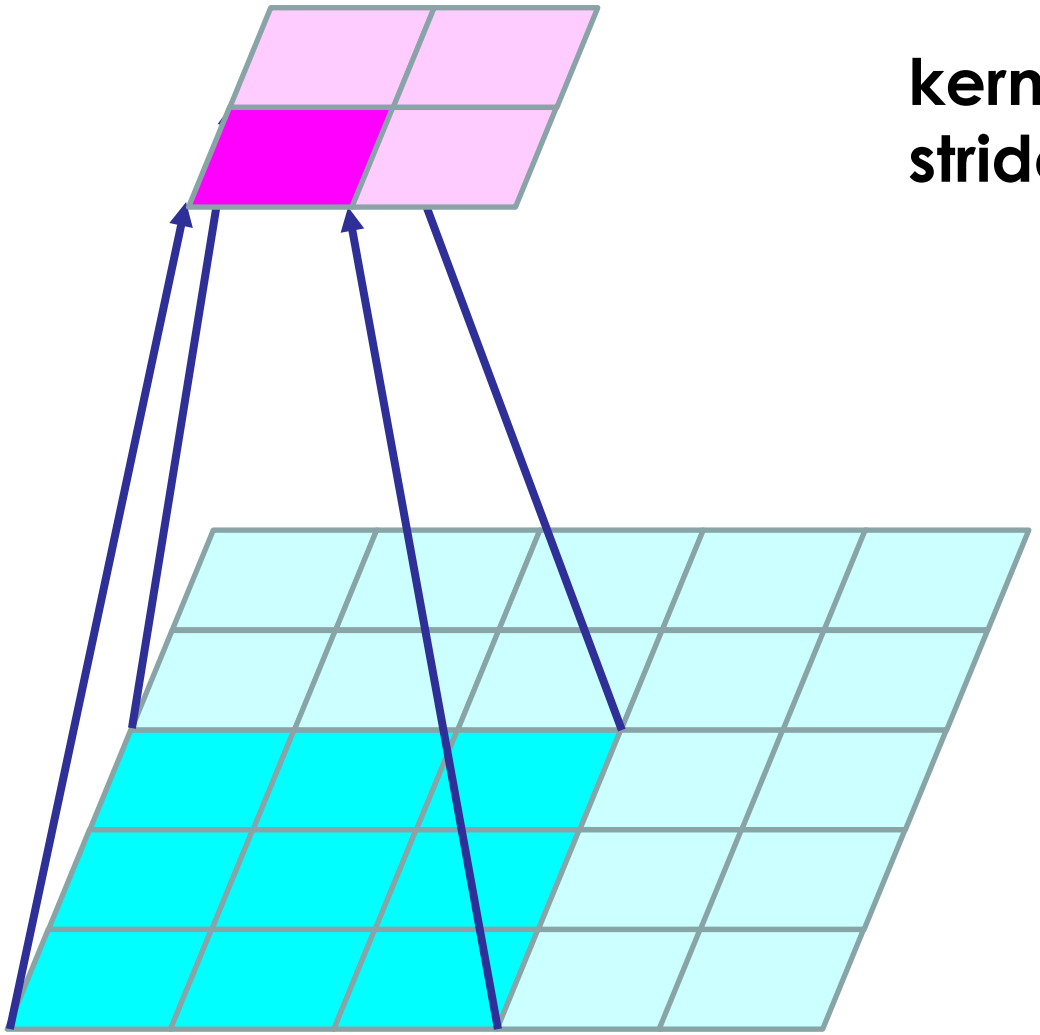
kernel size: 3x3
stride: 1

pooling size & stride



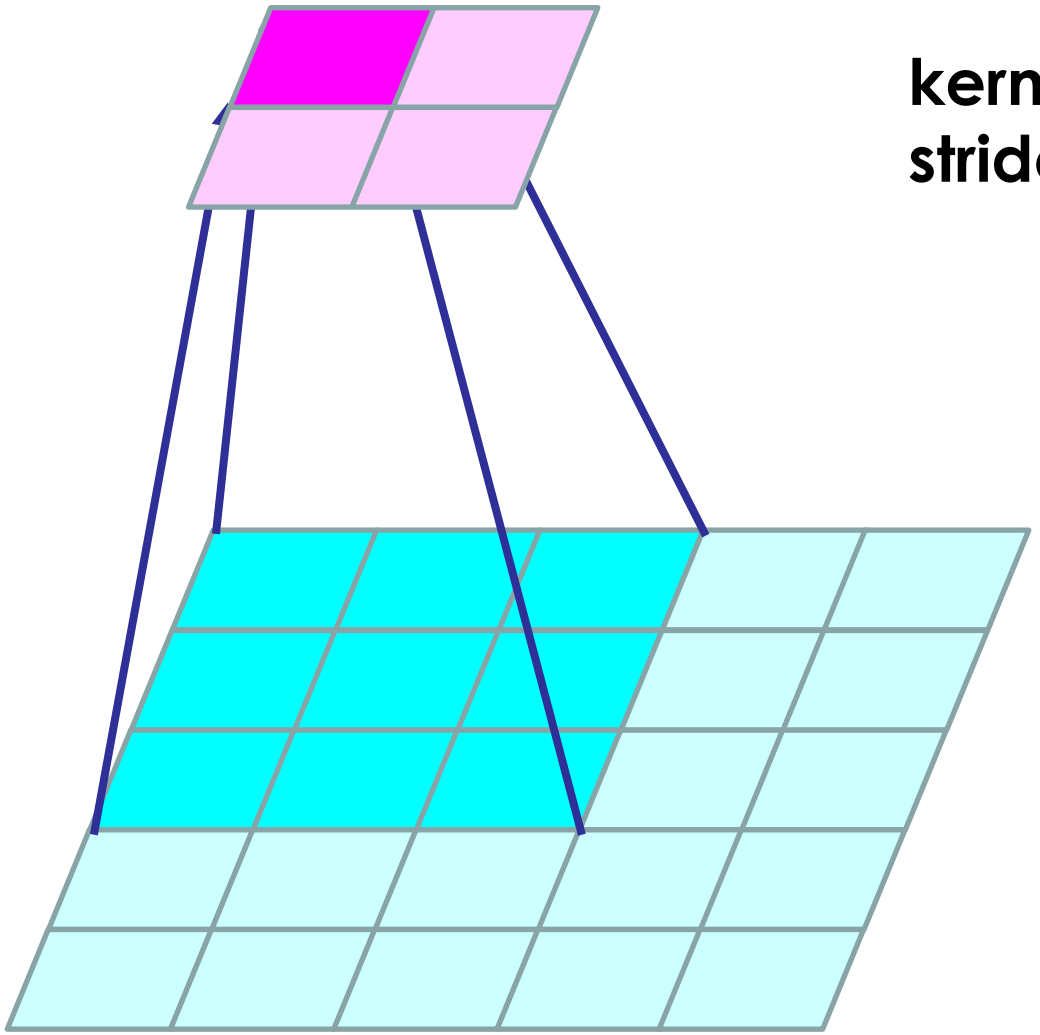
kernel size: 3x3
stride: 1

pooling size & stride



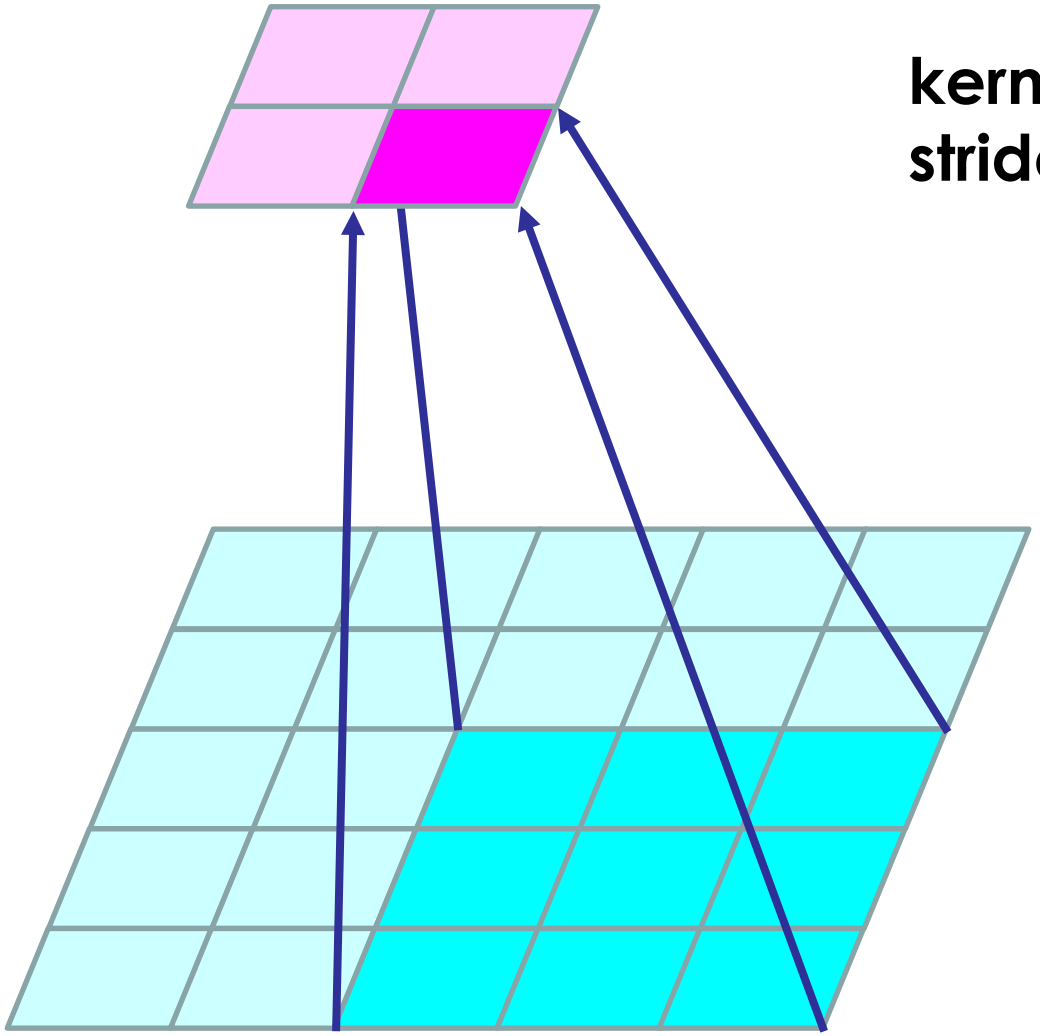
kernel size: 3x3
stride: 2

pooling size & stride



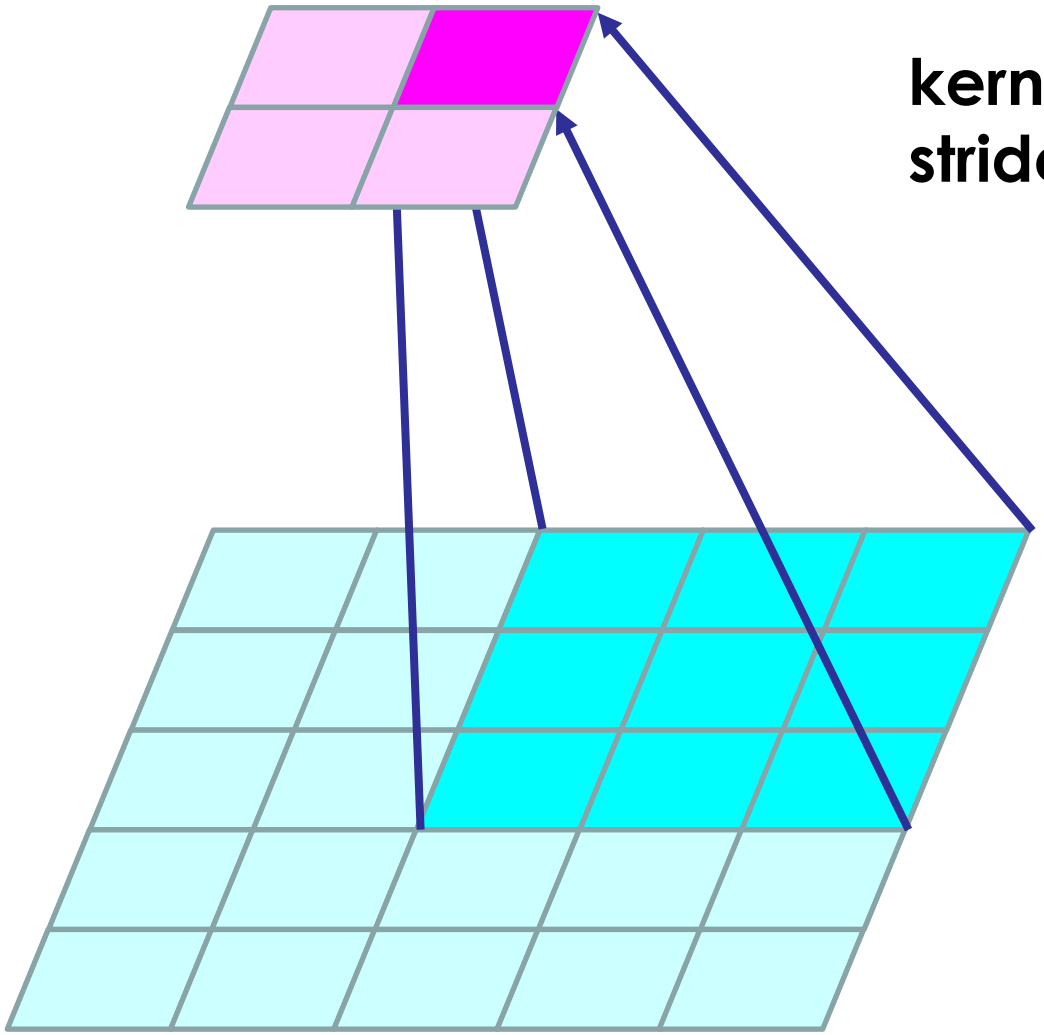
kernel size: 3x3
stride: 2

pooling size & stride



kernel size: 3x3
stride: 2

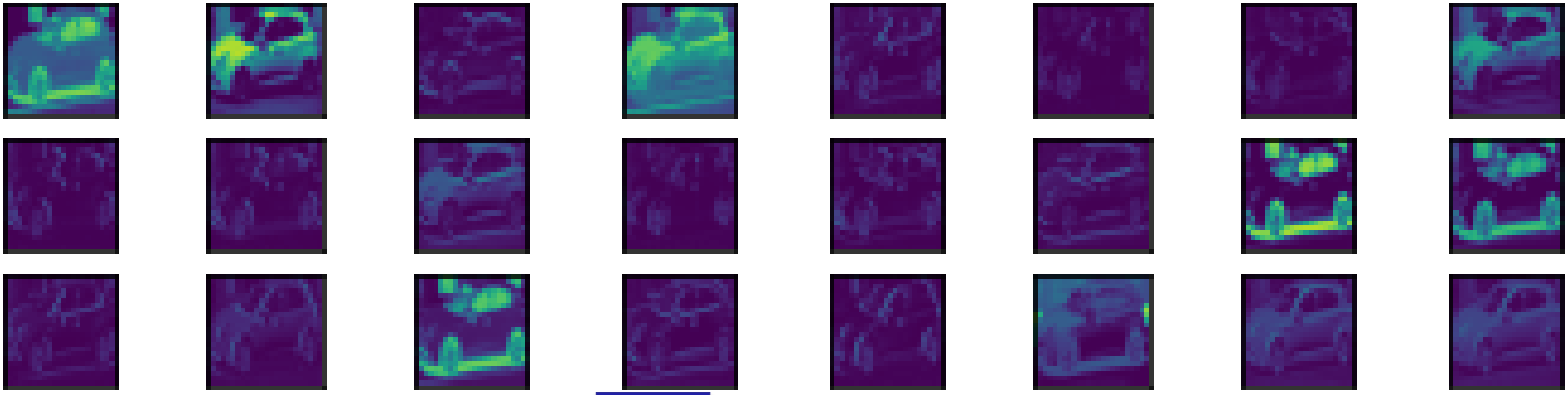
pooling size & stride



kernel size: 3x3
stride: 2

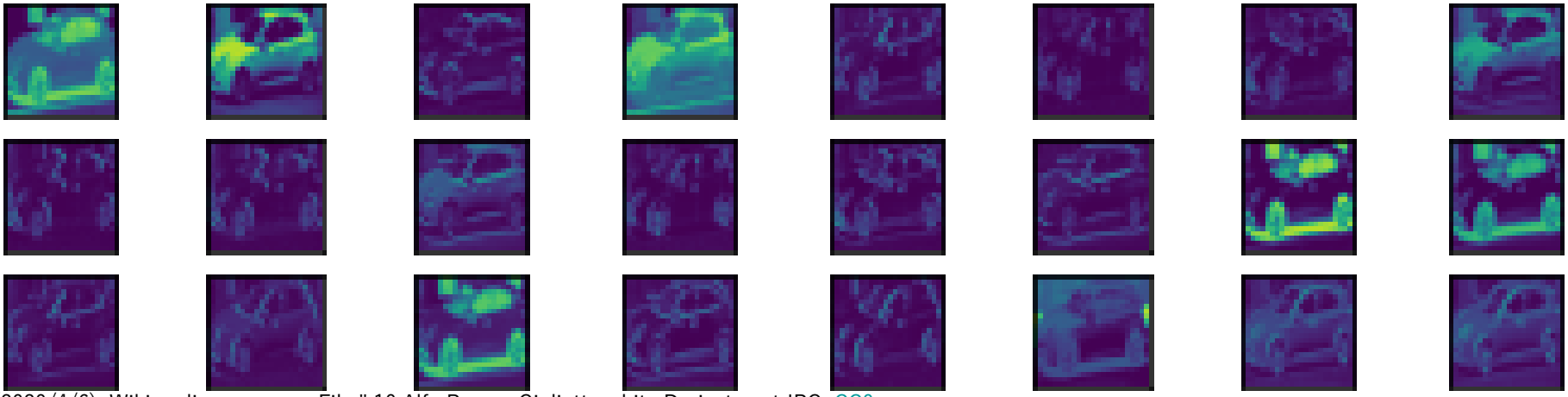
Pooling層の前後（第1層）

(224x224)x64枚



2x2 Max pooling

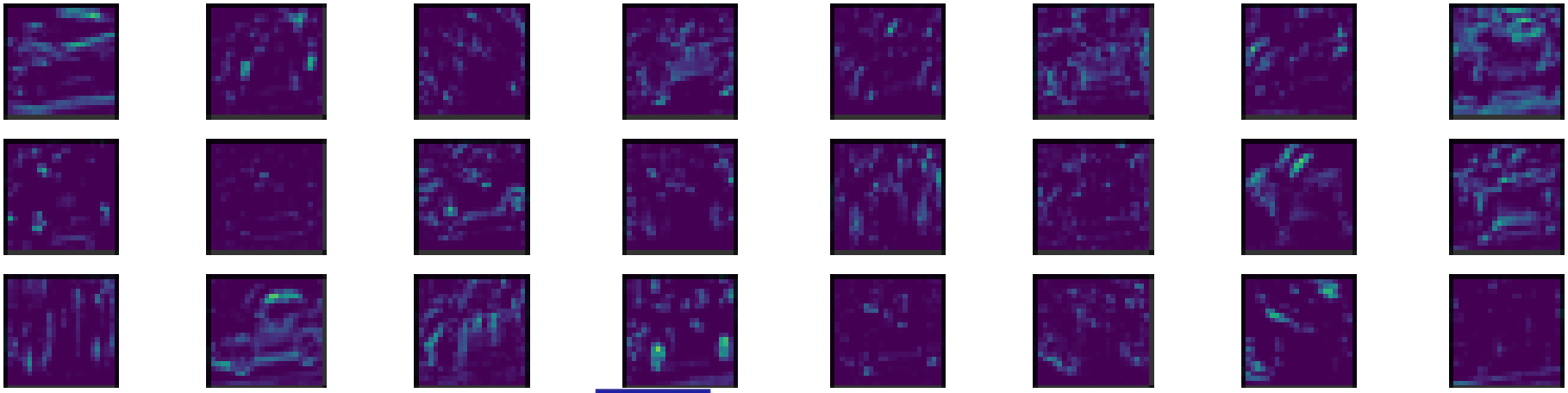
(112x112)x64枚



ref. (2020/4/6): Wikimedia commons: File:" 10 Alfa Romeo Giulietta white Derivate cut.JPG CC0
https://commons.wikimedia.org/wiki/Category:CC-Zero?uselang=ja#/media/File:%22_10_Alfa_Romeo_Giulietta_white_Derivate_cut.JPG

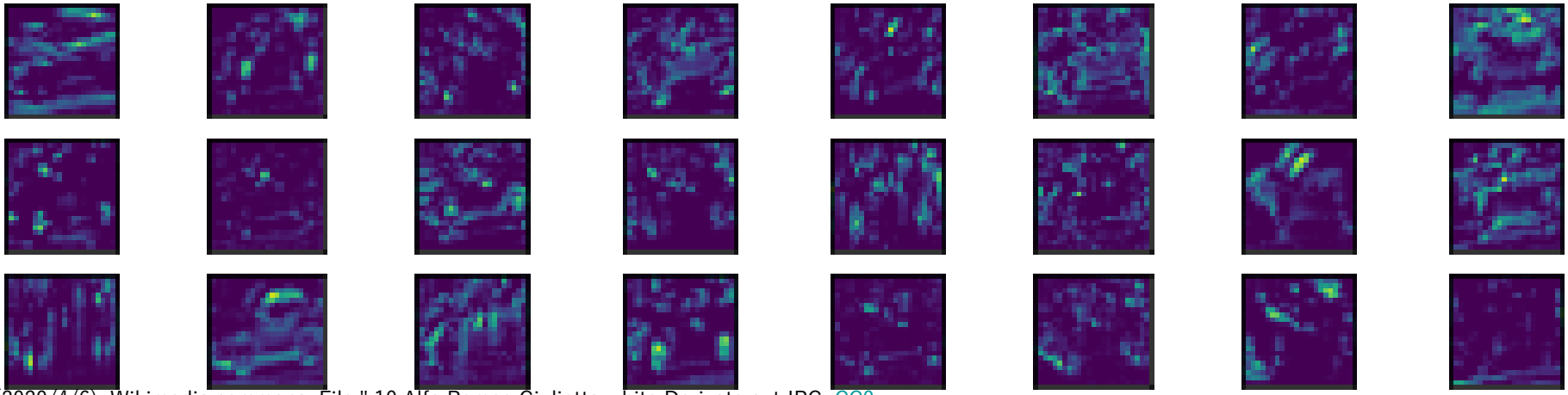
Pooling層の前後（第3層）

(56x56)x256枚



2x2 Max pooling

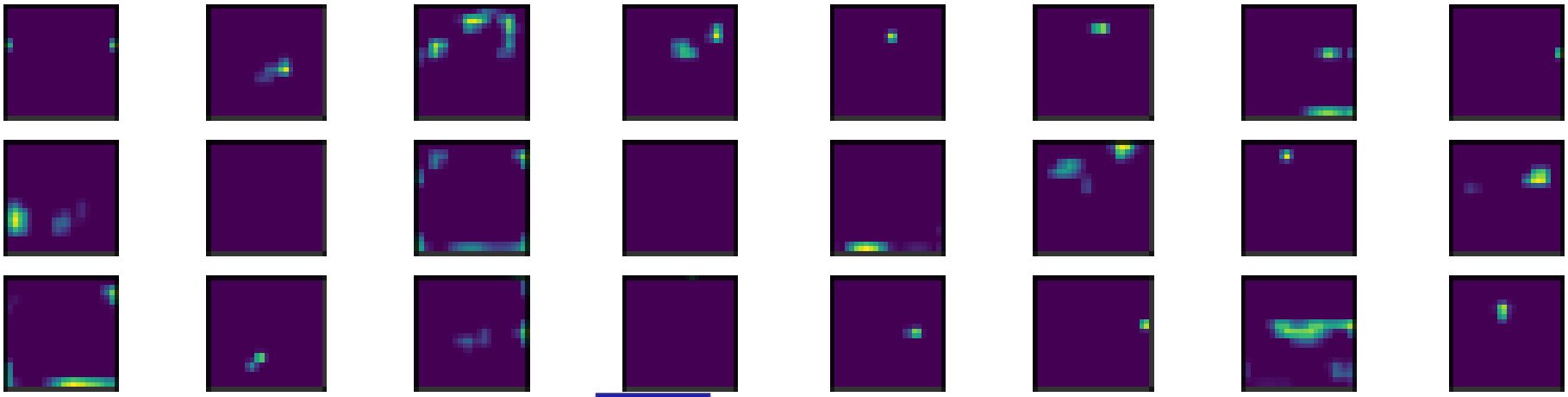
(28x28)x256枚



ref. (2020/4/6): Wikimedia commons: File:" 10 Alfa Romeo Giulietta white Derivate cut.JPG CC0
https://commons.wikimedia.org/wiki/Category:CC-Zero?uselang=ja#/media/File:%22_10_Alfa_Romeo_Giulietta_white_Derivate_cut.JPG

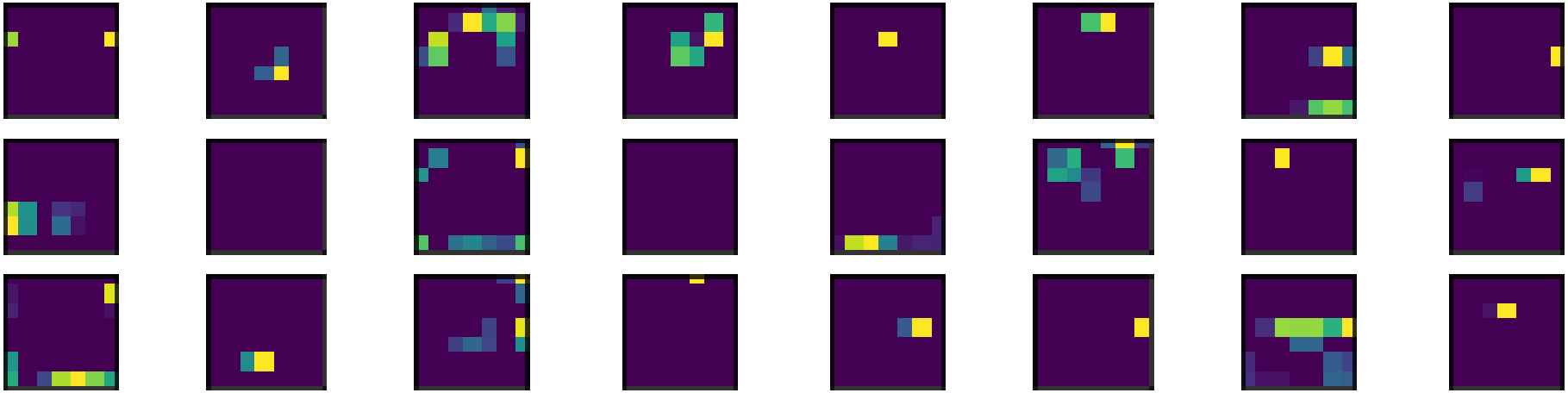
Pooling層の前後（第5層）

(14x14)x512枚



(7x7)x512枚

2x2 Max pooling

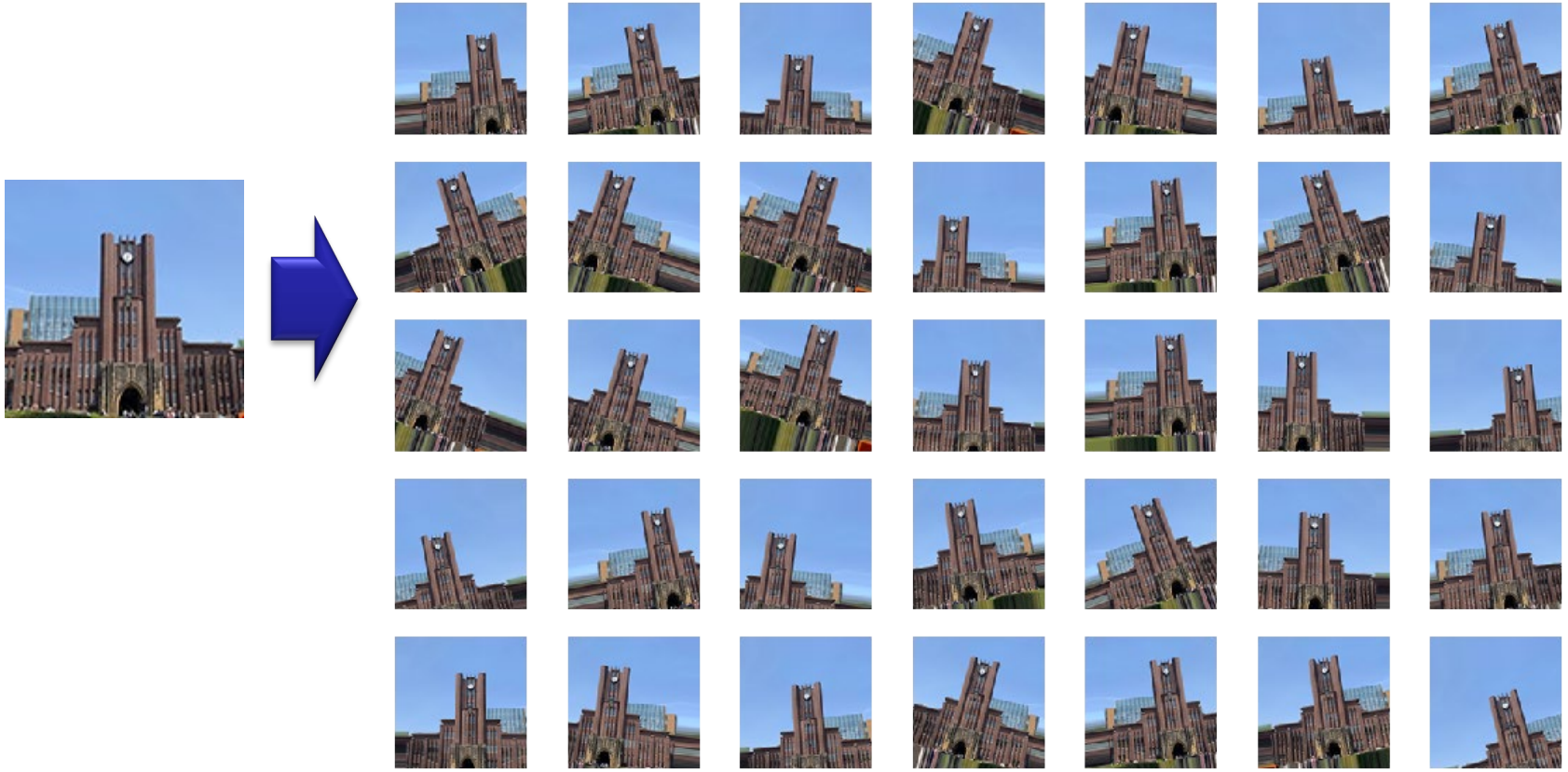


ref. (2020/4/6): Wikimedia commons: File:" 10 Alfa Romeo Giulietta white Derivate cut.JPG CC0
https://commons.wikimedia.org/wiki/Category:CC-Zero?uselang=ja#/media/File:%22_10_Alfa_Romeo_Giulietta_white_Derivate_cut.JPG

Data Augmentation

学習データを疑似的に増やす仕組み

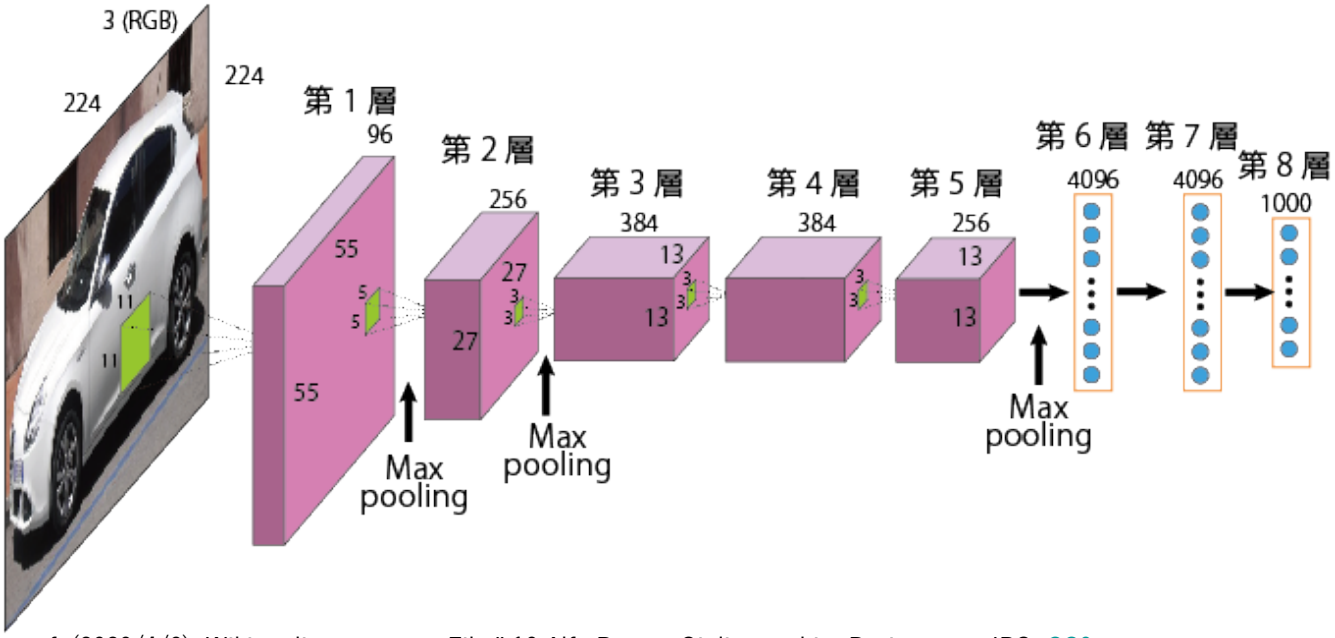
- ランダムクロップ、左右反転、水平・垂直シフト、ランダム回転、ランダム拡張・縮小…
- やっていいかどうか考えること（ex. 文字の左右反転）



モデルのドメイン適用

学習データの準備はとても大変！

- ImageNetは1000クラスの認識
  →それ以外の物体を認識したい場合は？
- 学習には時間とお金がかかる！
  →学習済みモデルを活用したい！



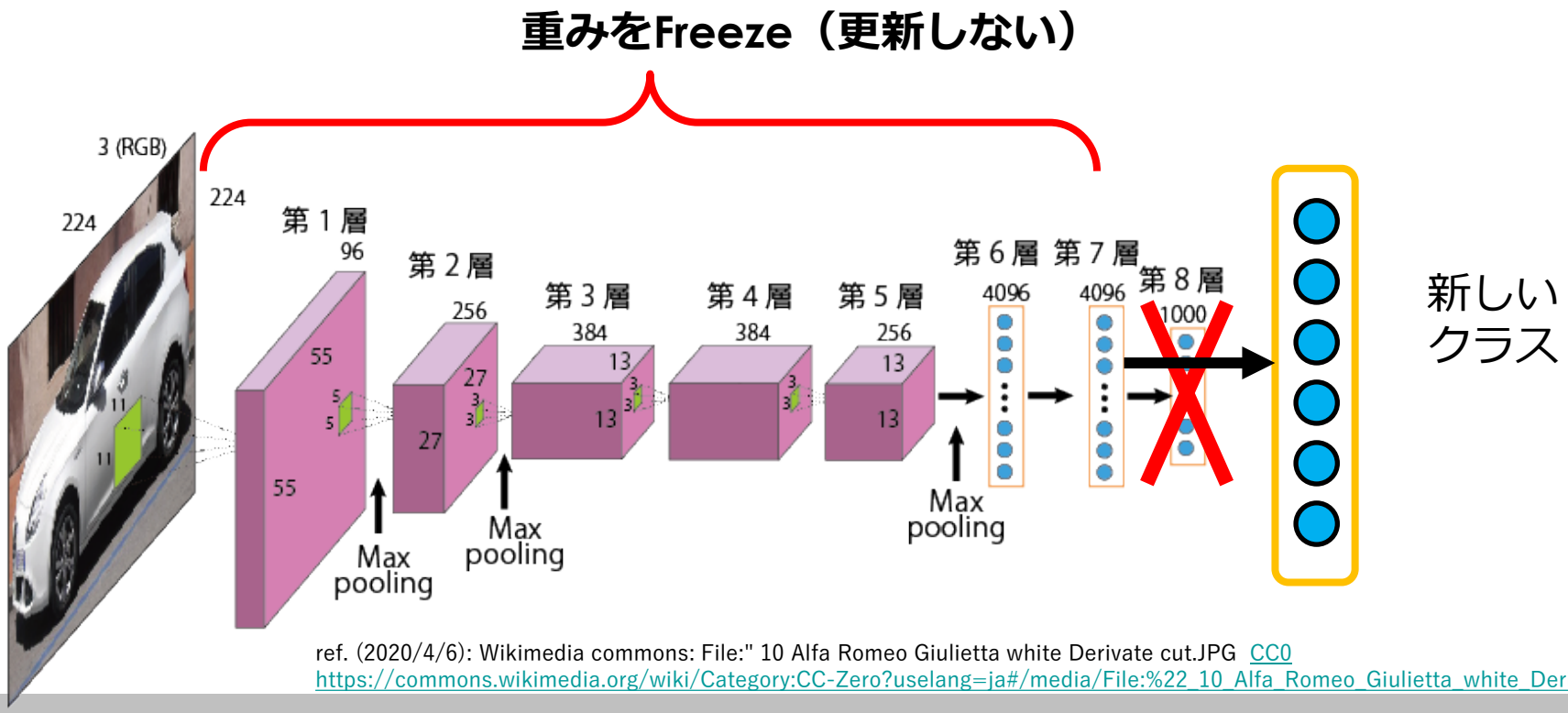
0.00 goldfish
0.00 great white shark
0.00 tiger shark
0.00 hammerhead
0.01 electric ray
...
0.89 car
...
0.00 stinkhorn
0.00 earthstar
0.00 hen-of-the-woods
0.02 bolete
0.00 ear, spike
0.00 toilet tissue

ref. (2020/4/6): Wikimedia commons: File:" 10 Alfa Romeo Giulietta white Derivate cut.JPG CC0
https://commons.wikimedia.org/wiki/Category:CC-Zero?uselang=ja#/media/File:%22_10_Alfa_Romeo_Giulietta_white_Derivate_cut.JPG

Fine-tuning/Continuous learning

学習済みモデルの重みを初期値として学習

- 重みの一部（通常は入力層に近い方からいくつかの層）の重みをFreeze
- 出力層に近いいくつかの層の重みを、新たに用意した学習データで学習
- 新たに層を加えてもいい
- **比較的少ない学習データでも高い精度が期待できる**

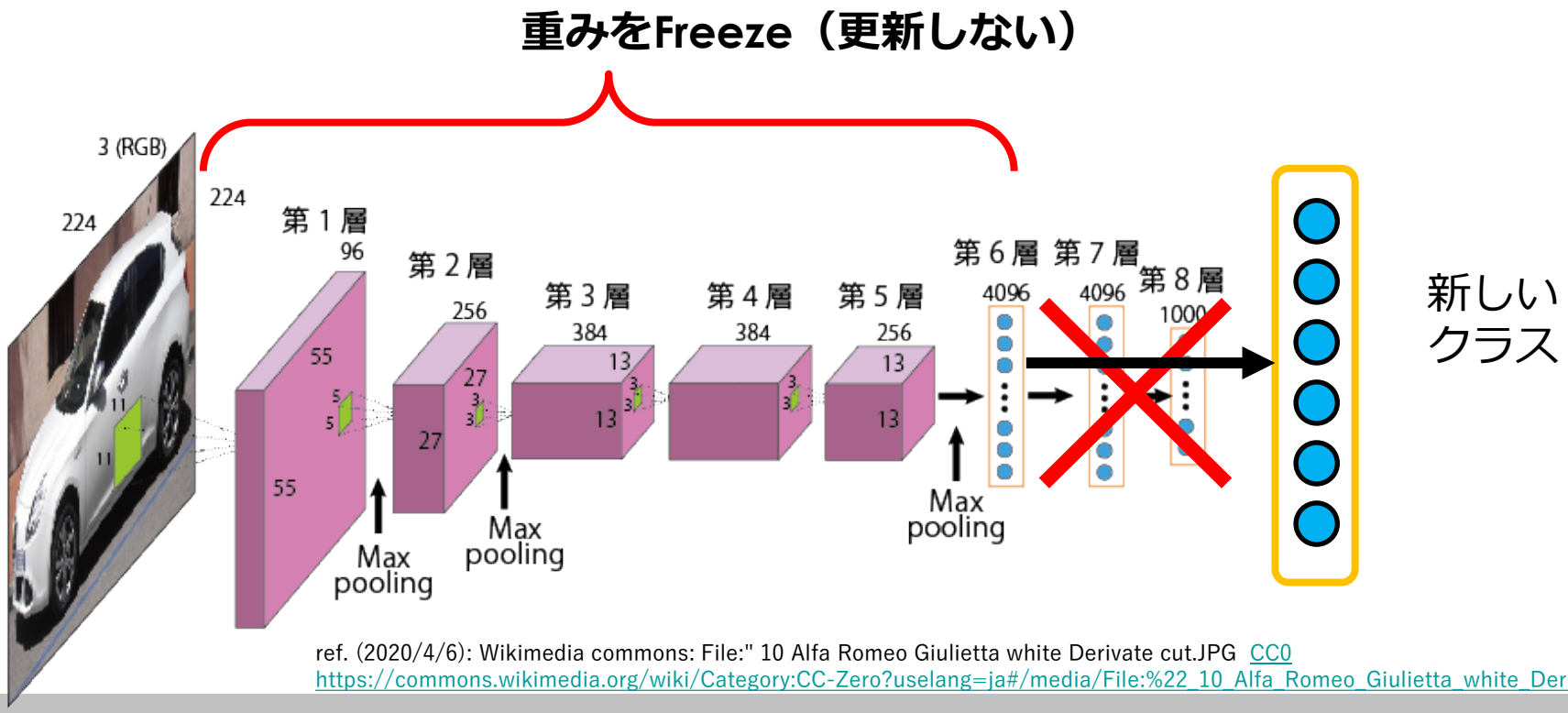


ref. (2020/4/6): Wikimedia commons: File:" 10 Alfa Romeo Giulietta white Derivate cut.JPG CC0
https://commons.wikimedia.org/wiki/Category:CC-Zero?uselang=ja#/media/File:%22_10_Alfa_Romeo_Giulietta_white_Derivate_cut.JPG

Fine-tuning/Continuous learning

学習済みモデルの重みを初期値として学習

- 重みの一部（通常は入力層に近い方からいくつかの層）の重みをFreeze
- 出力層に近いいくつかの層の重みを、新たに用意した学習データで学習
- 新たに層を加えてもいい
- **比較的少ない学習データでも高い精度が期待できる**

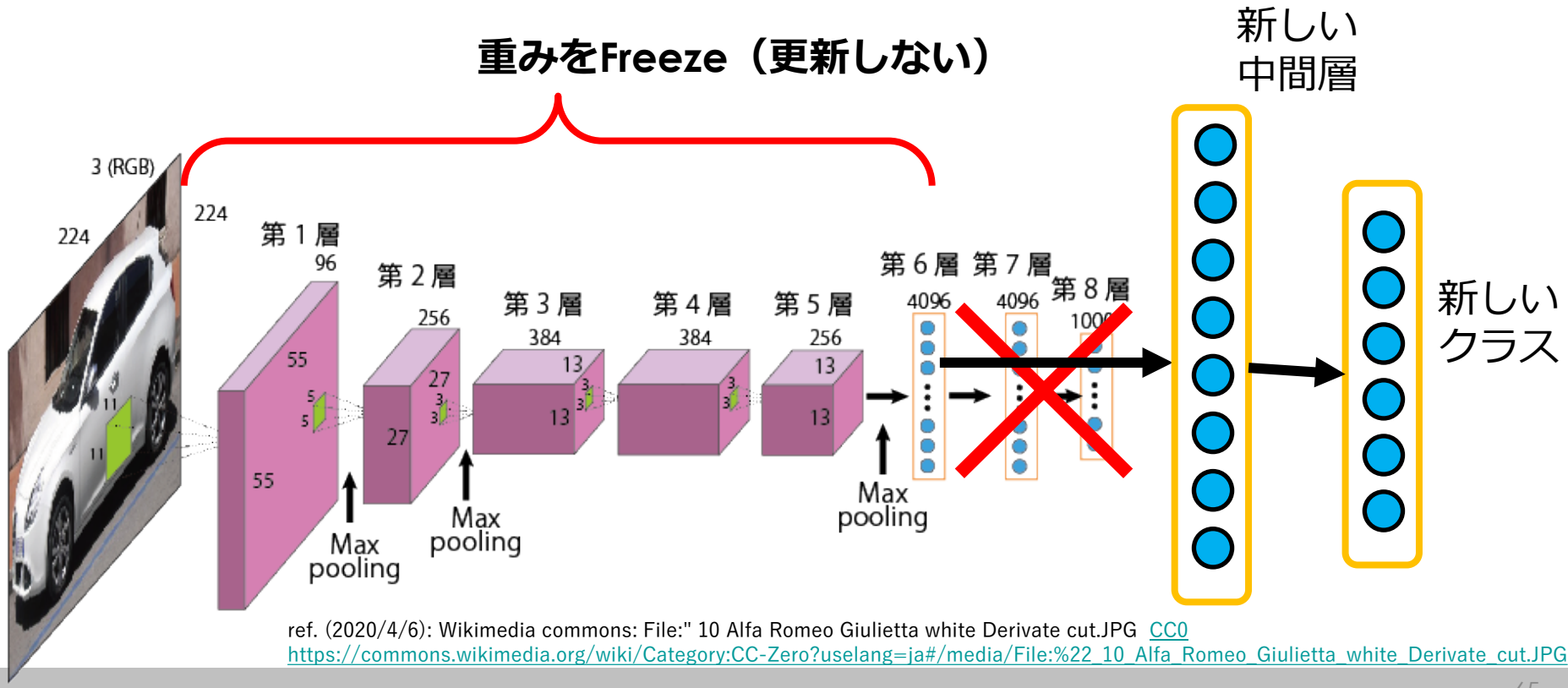


ref. (2020/4/6): Wikimedia commons: File:" 10 Alfa Romeo Giulietta white Derivate cut.JPG CC0
https://commons.wikimedia.org/wiki/Category:CC-Zero?uselang=ja#/media/File:%22_10_Alfa_Romeo_Giulietta_white_Derivate_cut.JPG

Fine-tuning/Continuous learning

学習済みモデルの重みを初期値として学習

- 重みの一部（通常は入力層に近い方からいくつかの層）の重みをFreeze
- 出力層に近いいくつかの層の重みを、新たに用意した学習データで学習
- 新たに層を加えてもいい
- **比較的少ない学習データでも高い精度が期待できる**

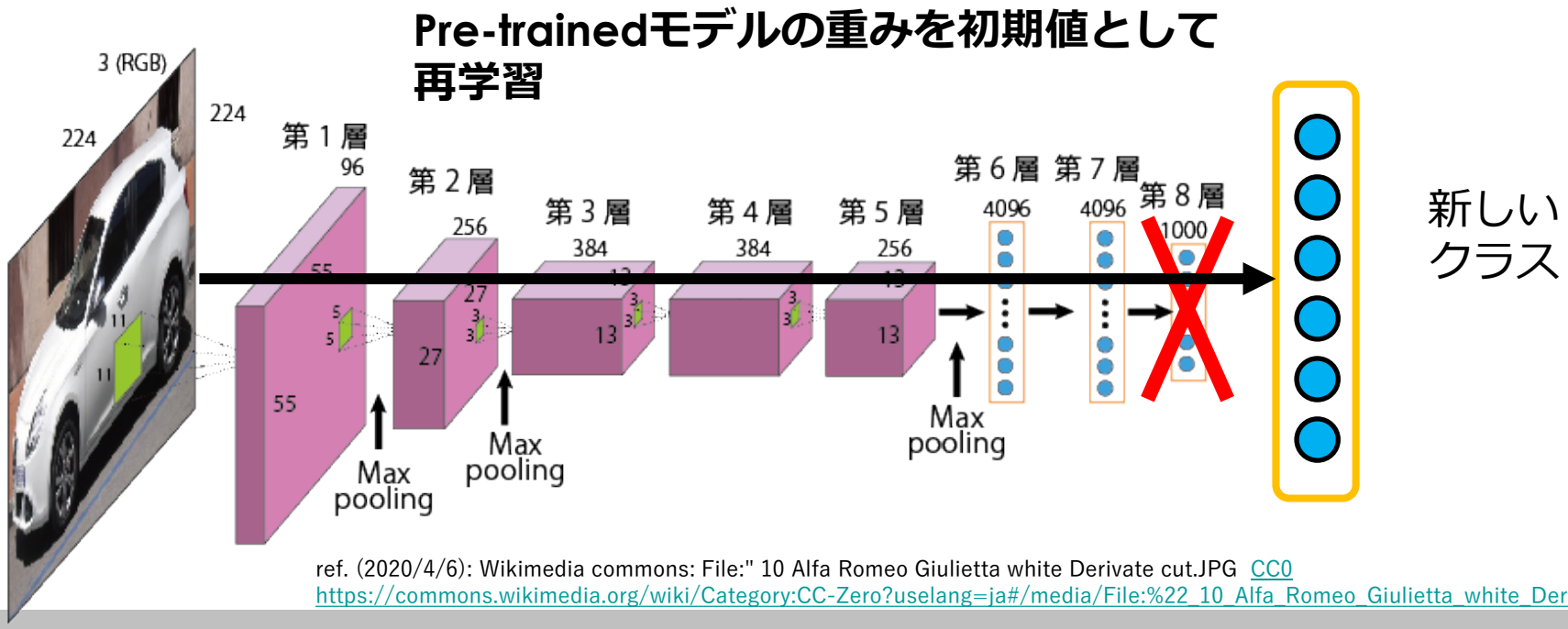


ref. (2020/4/6): Wikimedia commons: File:" 10 Alfa Romeo Giulietta white Derivate cut.JPG CC0
https://commons.wikimedia.org/wiki/Category:CC-Zero?uselang=ja#/media/File:%22_10_Alfa_Romeo_Giulietta_white_Derivate_cut.JPG

Fine-tuning/Continuous learning

学習済みモデルの重みを初期値として学習

- 重みの一部（通常は入力層に近い方からいくつかの層）の重みをFreeze
- 出力層に近いいくつかの層の重みを、新たに用意した学習データで学習
- 新たに層を加えてもいい
- **比較的少ない学習データでも高い精度が期待できる**

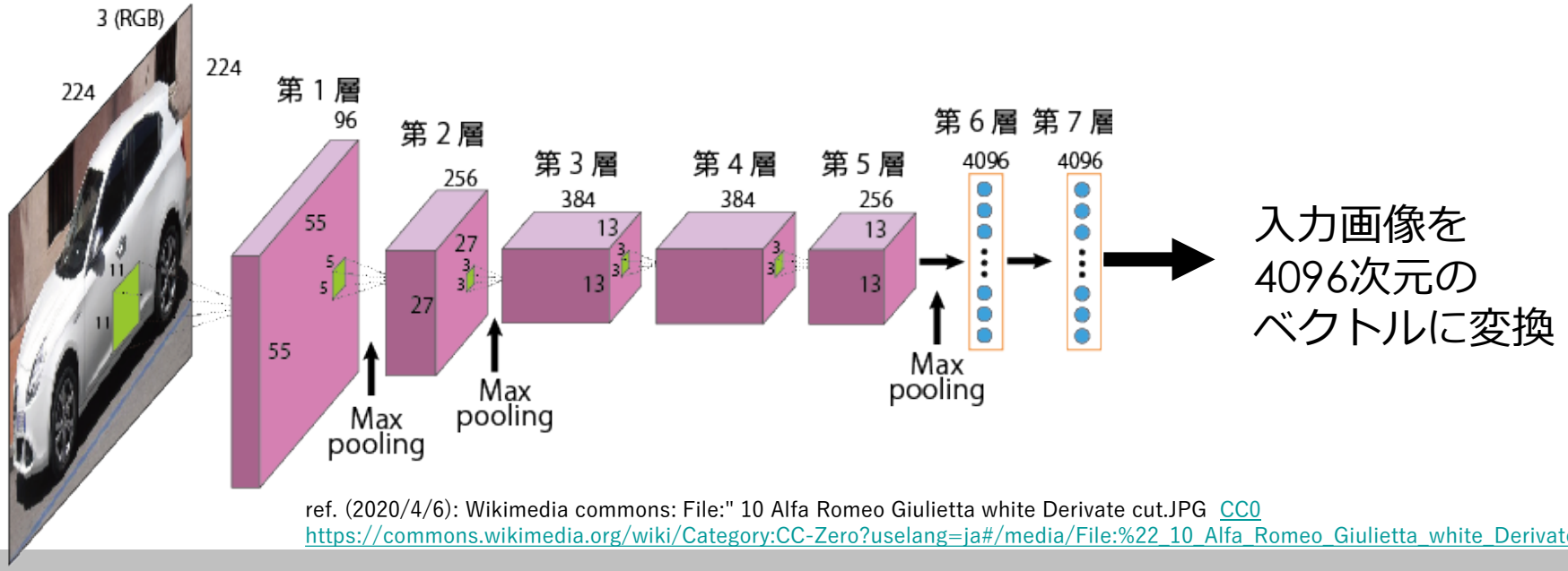


ref. (2020/4/6): Wikimedia commons: File:" 10 Alfa Romeo Giulietta white Derivate cut.JPG CC0
https://commons.wikimedia.org/wiki/Category:CC-Zero?uselang=ja#/media/File:%22_10_Alfa_Romeo_Giulietta_white_Derivate_cut.JPG

画像埋め込み（Image embedding）

中間層の出力を画像の特徴ベクトルとする

- 画像を機械学習に渡すため、とにかく特徴ベクトルに変換したい
- 入力画像が1000クラスに関するものかどうかはそれほど重要でない
  - 近い場合は後半の層の出力を使う
  - あまり関係がない場合は入力に近い層の出力を使う
  - 例）ResNet-50のFC2層の出力



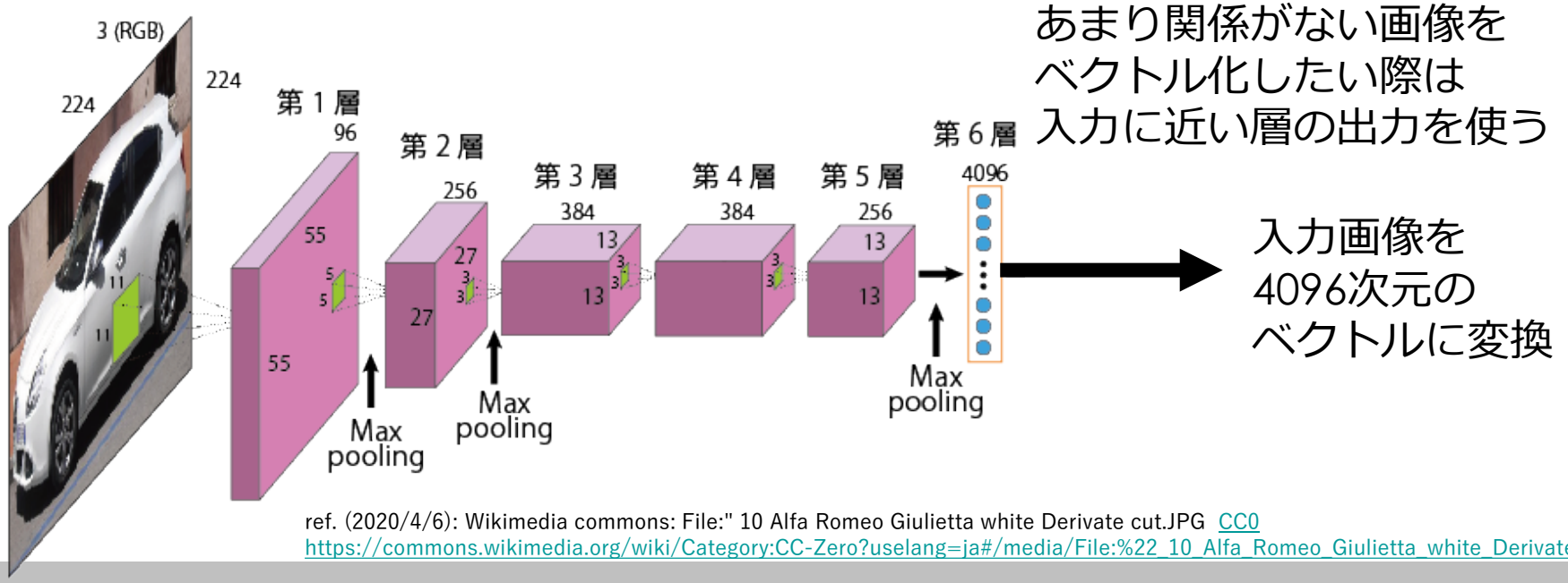
ref. (2020/4/6): Wikimedia commons: File:" 10 Alfa Romeo Giulietta white Derivate cut.JPG CC0
https://commons.wikimedia.org/wiki/Category:CC-Zero?uselang=ja#/media/File:%22_10_Alfa_Romeo_Giulietta_white_Derivate_cut.JPG

画像埋め込み（Image embedding）

中間層の出力を画像の特徴ベクトルとする

- 画像を機械学習に渡すため、とにかく特徴ベクトルに変換したい
- 入力画像が1000クラスに関するものかどうかはそれほど重要でない
  - 近い場合は後半の層の出力を使う
  - あまり関係がない場合は入力に近い層の出力を使う
  - 例）ResNet-50のFC2層の出力

学習したときのクラスと
あまり関係がない画像を
ベクトル化したい際は
入力に近い層の出力を使う



ref. (2020/4/6): Wikimedia commons: File:" 10 Alfa Romeo Giulietta white Derivate cut.JPG CC0
https://commons.wikimedia.org/wiki/Category:CC-Zero?uselang=ja#/media/File:%22_10_Alfa_Romeo_Giulietta_white_Derivate_cut.JPG

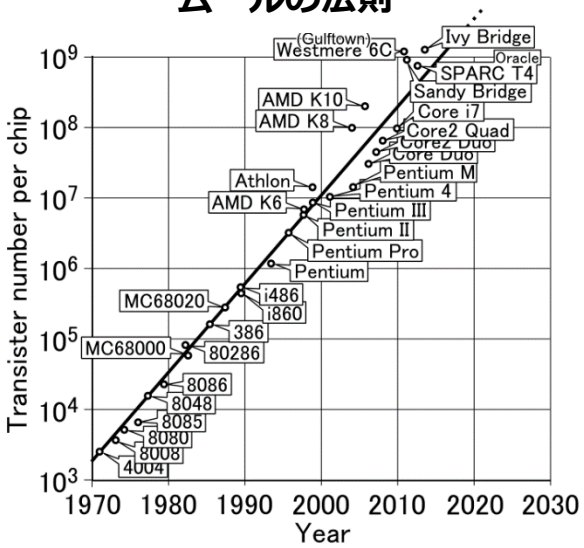
深層学習はそれまでの画像処理と何が違ったのか？

- AlexNet (2012)登場より前の画像認識では、CNNにおける第1~2層の出力に相当する情報を使って認識していた
  → Deep Networkと対比してShallow networkと呼ばれる
- 2000年前後にいくつかの技術革新
  - **数学的解法**：勾配消失問題（層を深くすると学習が進まなくなる現象）に対する効率的な解決法の提案（1990年代後半）
  - **GPGPUの発展**：コンピュータグラフィックスの描画に用いられていたGPUをベクトル計算機とみなして気象や地震シミュレーション等、数値計算に利用（2006年NVIDIAがCUDA提供開始）
  - **Big Data時代の到来**：
    Webで画像やテキストなどが大量に収集できるようになり、
    モデルの学習に使えるデータが爆発的に増加
- 様々な画像処理タスクに対する学習データセットが公開

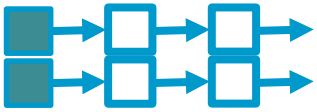
General Purpose Graphics Processing Unit (GPGPU)

- 深層学習は四則演算を同時に大量に行うためCPUでは実用不可能な技術
- CPUの性能向上は限界が近づいていた→単体で限界なら複数で並列処理
- ゲーム等のコンピュータグラフィック用に開発されたGPUを機械学習に転用

ムールの法則



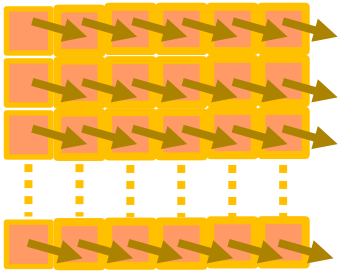
CPU：デュアルコアの場合



1度に2つ
同時計算

- ☺ 1個1個の計算はGPUよりも高速
- ☺ 複雑な計算も可能
- 😟 1回につき2つしか計算できない
- ☺ 前の処理結果を受けて処理するような手法ならこちらで問題ない

GPU：Tesla V100（5,120コア）の場合



1度に
5,120個
同時計算

- ☺ 一度に大量の計算を実行
- ☺ まだまだ高速化の余地あり
- 😟 並列計算するようにプログラムを組まなければならない（自動的に並列計算になるわけではない）
- ☺ 深層学習にはピッタリ

ref. (2020/06/30) File:Moores law (1970-2011).PNG, CC 表示-継承 3.0, https://commons.wikimedia.org/wiki/File:Moores_law_(1970-2011).PNG

画像認識の例：手書き文字認識

- 画像認識初期のタスク
- よく使われるデータはMNIST (Mixed National Institute of Standards and Technology database)
  - 「0~9」の10種類の数字の認識
    = 10クラス分類タスク
  - 各画像に数字が1つ記入
  - 解像度は20x20 pixel
  - 訓練データ：60,000枚
  - 評価データ：10,000枚
- 間違いやすいサンプルも含む

著作権等の都合上、ここに挿入されていた画像を削除しました。

手書き文字認識結果と正解

http://yann.lecun.com/exdb/publis/pdf/lecun-98.pdf
Fig.8の左上2行4列

著作権等の都合上、ここに挿入されていた画像を削除しました。

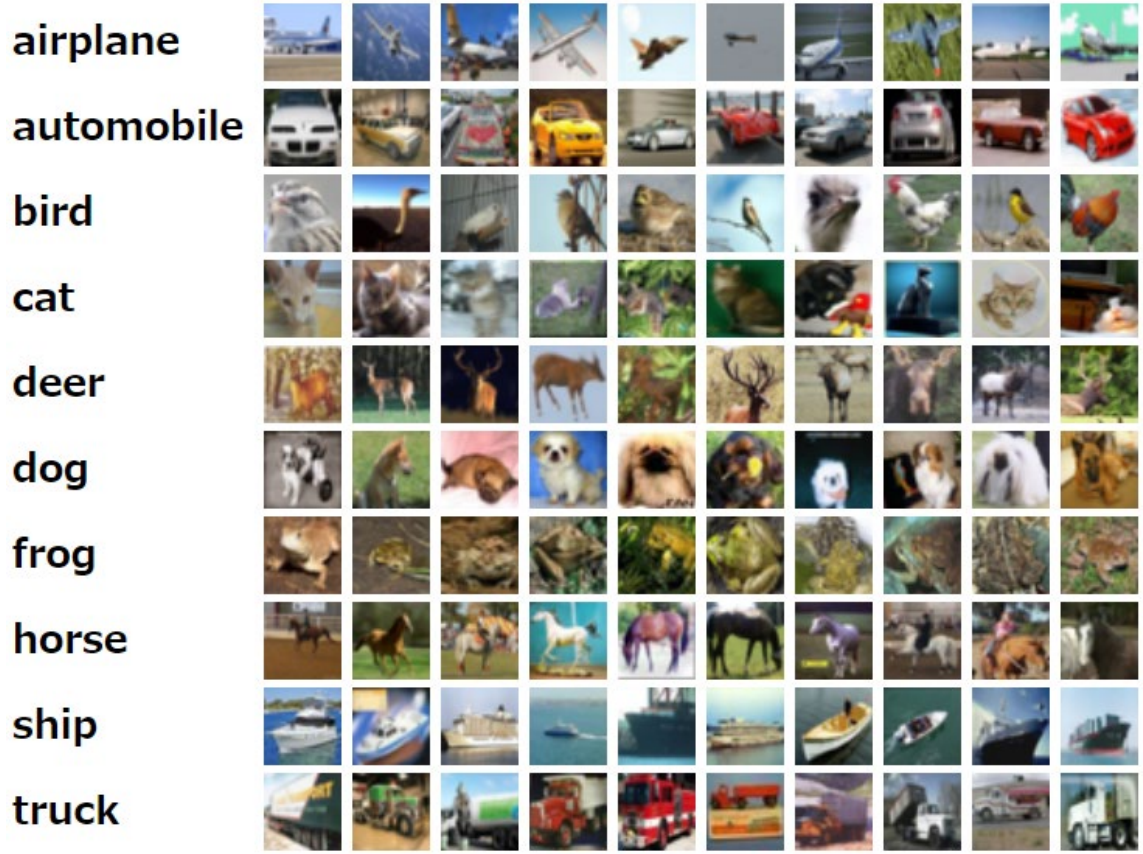
手書き文字の例

http://yann.lecun.com/exdb/publis/pdf/lecun-98.pdf
Fig.4

ref. (2020/4/6): THE MNIST DATABASE of handwritten digits http://yann.lecun.com/exdb/mnist/
Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. "Gradient-based learning applied to document recognition." Proceedings of the IEEE, 86(11):2278-2324, November 1998.

画像認識の例： 一般物体認識

- 写真に写っている物体が何かを当てるタスク
- 基本のデータセット：CIFAR-10
  - 10クラス分類、各クラス6000枚、32x32 pixelのカラー画像

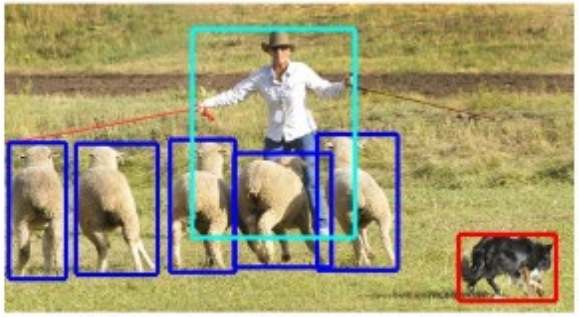


ref. (2020/04/06): The CIFAR-10 dataset, http://www.cs.toronto.edu/~kriz/cifar.html

様々な画像処理タスク

- 画像に対し、クラスを1つ特定する物体認識タスク以外にも、様々な画像処理タスクがある
- 多くの研究グループがデータセットを提供（下図はMicrosoft COCO)

(a) Image classification


(b) Object localization


(c) Semantic segmentation


(d) This work

(a)複数物体認識
(b)矩形領域検出
(c)領域
**　　セグメンテーション**
(d)個体別領域
**　　セグメンテーション**

個体ごとに
別の領域として抽出

Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick, Piotr Dollár, “Microsoft COCO: Common Objects in Context”, CVPR2015

画像の動作認識 (Action recognition)

- 物体ではなく動作を認識するタスク
- 下図はPASCAL VOCのAction Classification Competitionの例
  - 10クラス(ジャンプする、電話する、楽器を演奏する、読む、自転車やバイクに乗る、馬に乗る、走る、写真を撮る、パソコンを使う、歩く)
  - 上の10クラスに属さない動作を行う画像（"その他")も含まれている

ref. (2020/4/6) The PASCAL Visual Object Classes Challenge 2012 (VOC2012) http://host.robots.ox.ac.uk/pascal/VOC/

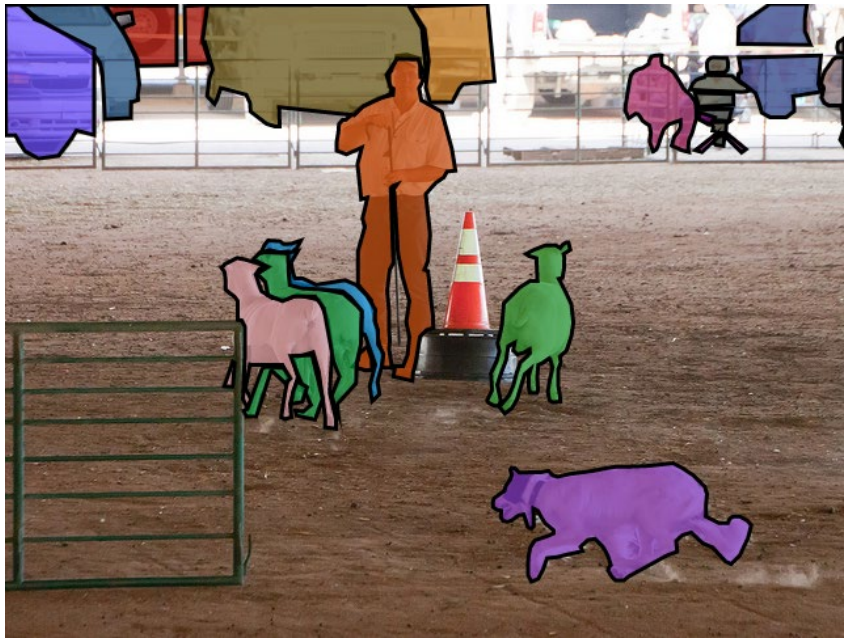
動画における動作認識 (Action recognition)

- 画像ではなく動画を対象とした動作認識
- ショートクリップに写っている人物の動作を認識
- 下図はUCF101: University of Central Floridaの研究グループが作った、101種類の動作認識を行うタスクのためのデータセットの例
  - 合計13320クリップ
  - 各動画の平均長は7.21 s（最短1.06 s、最長71.04 sec)
  - 解像度320x240
  - うち51種類については音付

UCF101: A Dataset of 101 Human Action Classes From Videos in The Wild
https://www.crcv.ucf.edu/wp-content/uploads/2019/03/UCF101_CRCV-TR-12-01.pdf

画像に対するキャプション生成 (Image captioning)

- 画像に対し、その内容を説明する短い文（caption)を生成するタスク
- データセットの一例：MS-COCO
  - 123,287枚の画像に886,284個の物体領域
  - 個々の物体ごとの領域とその物体ラベル
  - **５文程度のキャプション**



物体ラベル：
"person", "car", "dog", "sheep", "chair"

キャプション：
- the dog is hurdling the sheep towards the man.
  (その犬は羊を男のほうに追いやっている)
- a man and a dog herding some sheep in a fenced off area.
  (男と犬が柵で囲まれた場所で羊を群れにしている)
- a dog chases after some goats while a man watches.
  (男が見守る中、犬がヤギを追いかける)
- a man guides a dog to herd sheep.
  (男が犬を誘導して羊の群れを作る)
- a dog herds sheep into a pen as a shepherd looks on
  (犬は羊飼いが見ている間に、羊を小さな群れにする)

ref. (2020/4/6) Microsoft COCO, http://cocodataset.org/

画像に対するキャプション生成 (Image captioning)

深層学習により自動生成されたキャプションの例
（ありがちな情景と誤認して生成された誤ったキャプションがあることに注意）

ref. (2020.4.6) Andrej Karpathy, Li Fei-Fei, “Deep Visual-Semantic Alignments for Generating Image Descriptions”, CVPR2015.
https://cs.stanford.edu/people/karpathy/deepimagesent/

人体の姿勢推定

- OpenPose: 米国カーネギーメロン大学が開発
  - 人体の19か所の関節（左右の区別あり）を高精度で検出
  - 商用にも広く使われている

複数名いても正しく対応関係を獲得

右の肘と右の手首の連結の尤度マップ 位置と方向を検出

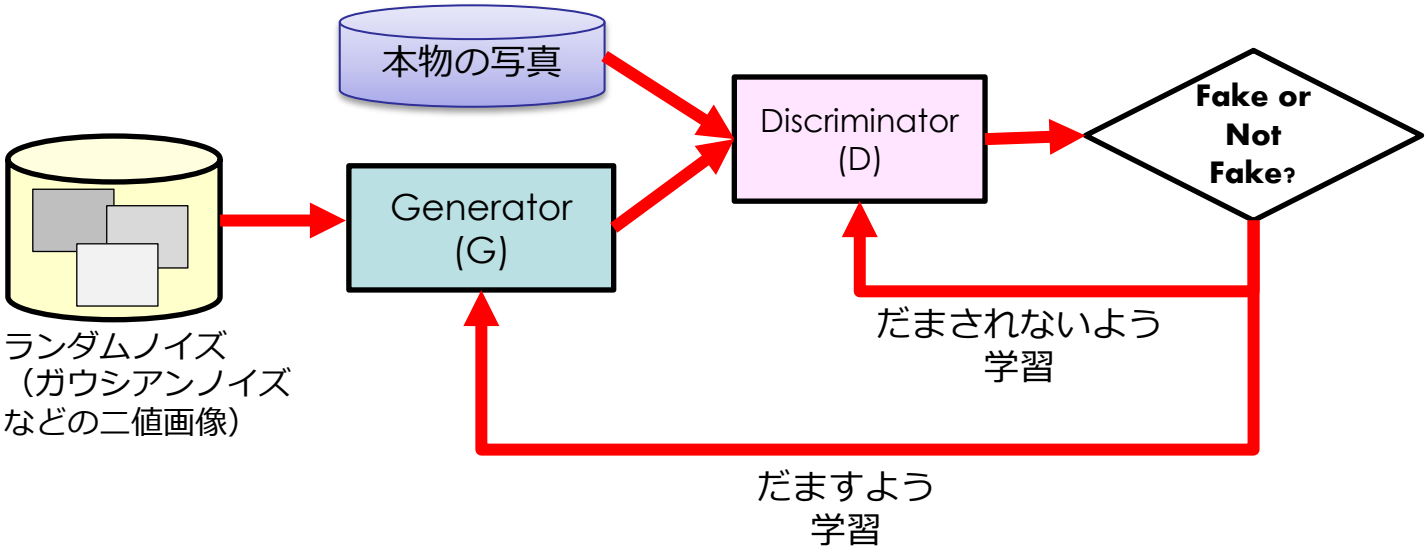
ref. (2020/4/6) Zhe Cao, Tomas Simon, Shih-En Wei, Yaser Sheikh, “Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields”, CVPR2017, https://github.com/CMU-Perceptual-Computing-Lab/openpose

敵対的生成ネットワーク: Generative Adversarial Network (GAN) [2014]

ref. Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville Yoshua Bengio, "Generative Adversarial Nets", NIPS2017. https://papers.nips.cc/paper/5423-generative-adversarial-nets

敵対的生成ネットワーク: Generative Adversarial Network (GAN) [2014]

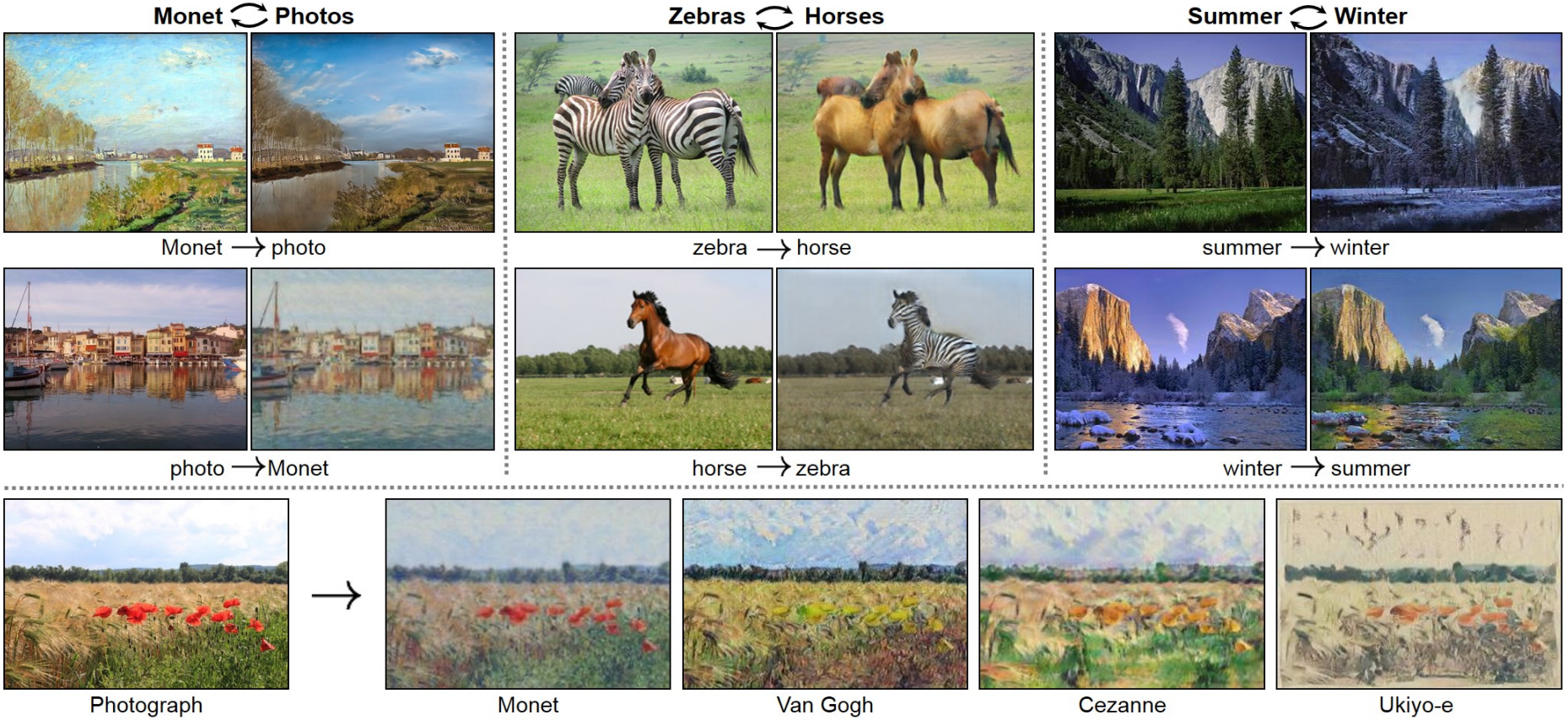
- 偽の画像を生成するネットワーク(Generator)と、本物か偽物かを見分けるネットワーク(Discriminator)を用意
- Discriminatorは正しく見分けるよう学習、Generatorは見分けられないよう学習



ref. Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville Yoshua Bengio, "Generative Adversarial Nets", NIPS2017. https://papers.nips.cc/paper/5423-generative-adversarial-nets

GANの応用：スタイル変換（StyleTransfer)[2017]

- 大局的な特徴はそのままで、ローカルな特徴をだますよう学習
- 画像を任意のスタイルに変換



Jun-Yan Zhu, Taesung Park, Phillip Isola, Alexei A. Efros, "Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks", ICCV2017, https://junyanz.github.io/CycleGAN/

Artistic style transfer for videos [GCPR 2016]

動画と絵を入力すると、動画をその絵のスタイルに変換

著作権等の都合上、ここに挿入されていた動画を
削除しました。

Artistic style transfer for videos

0:00　の画像

https://www.youtube.com/watch?v=Khuj4ASldmU

ref. Style transfer for videos, as described in the paper "Artistic style transfer for videos" by Manuel Ruder, Alexey Dosovitskiy and Thomas Brox http://arxiv.org/abs/1604.08610, ref. https://www.youtube.com/watch?v=Khuj4ASldmU

styleGAN [ICLR2018 by NVIDIA]

存在しない人の顔画像を生成

著作権等の都合上、ここに挿入されていた動画を削除しました。

A Style-Based Generator Architecture for Generative Adversarial Networks

1:41頃の画像

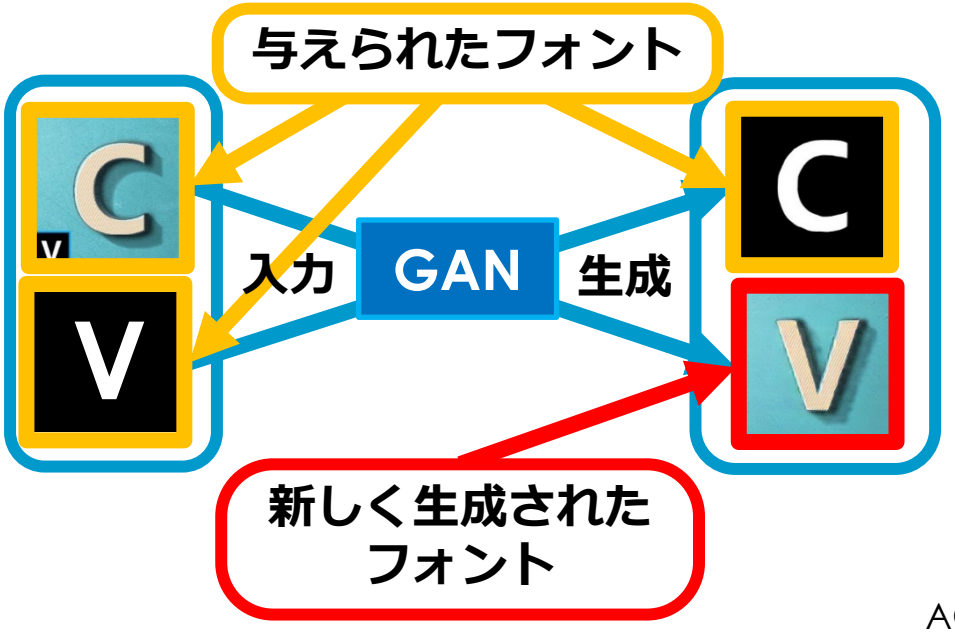
https://youtu.be/kSLJriaOumA

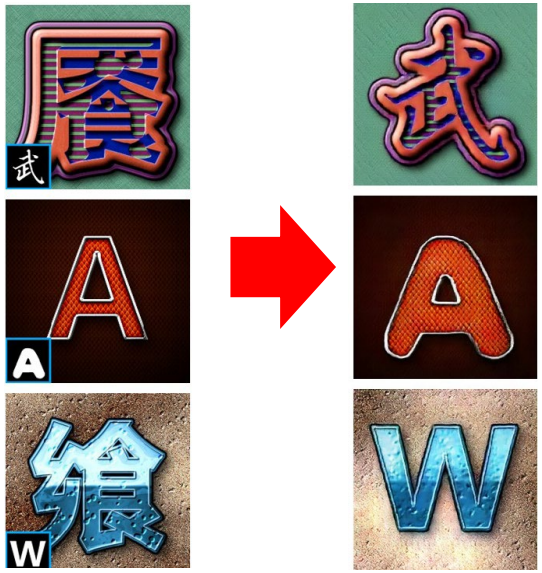
ref. Tero Karras, Samuli Laine, Timo Aila, " A Style-Based Generator Architecture for Generative Adversarial Networks", CVPR2019, https://arxiv.org/abs/1812.04948

ディープラーニングによるフォントの生成

- 多用な用途に適したデザインのフォントを使いたい
- 日本語文字は常用文字だけで2300文字
  → 新しいデザインのフォントを生成するのは大変

→ 数文字だけデザインすれば、残りのフォントを自動生成！

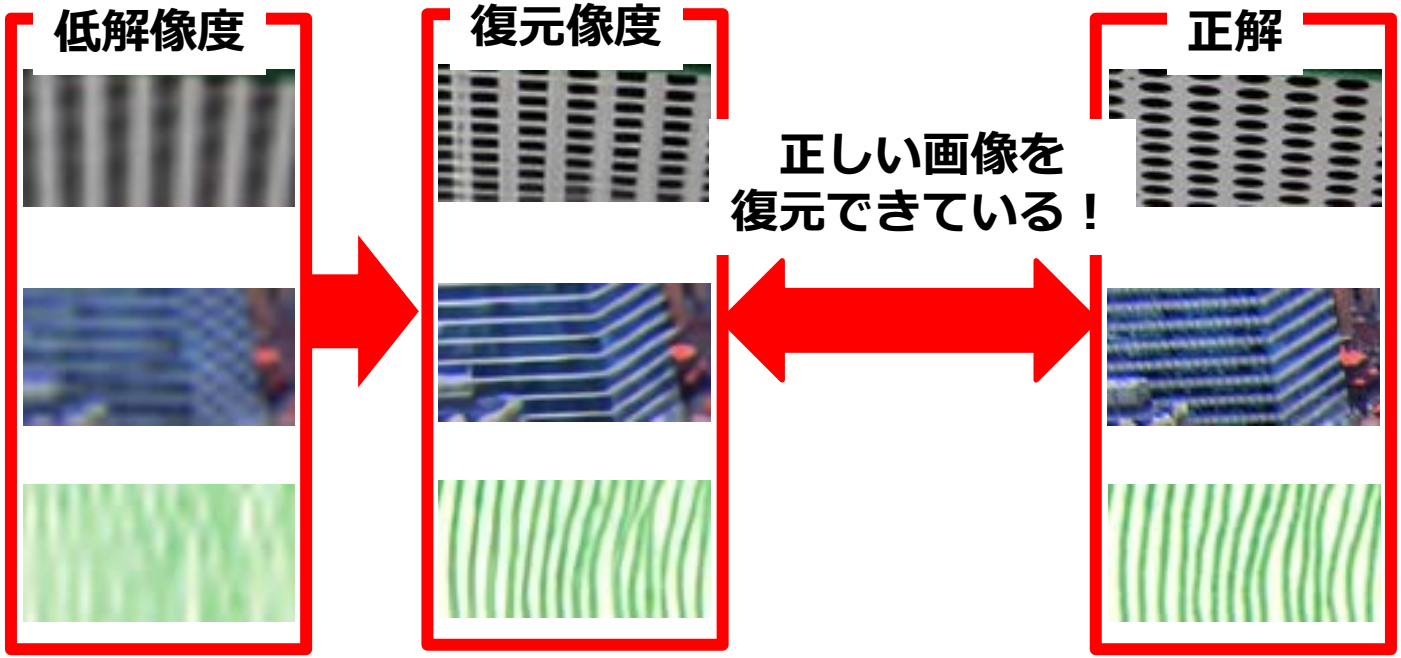




AGIS-NET, https://hologerry.github.io/AGIS-Net/

超解像深層学習（Single Image Super-Resolution）

- 低解像度画像や一部が欠けている画像を高解像度画像に変換する
- 大量の画像から学習したモデルを使って復元
  （実際には復元というよりは、それらしい画像を生成するに近い）



Yulun Zhang, Kunpeng Li, Kai Li, Lichen Wang, Bineng Zhong, Yun Fu,
"Image Super-Resolution Using Very Deep Residual Channel Attention Networks", ECCV2018

Image Inpainting

- Demo: https://www.nvidia.com/research/inpainting/selection
- 様々な画像に対し、原画像とランダムに切り取った画像のペアを作成→学習データとする
- モデルの学習にはNVIDIA V100 GPUを使ってざっと2週間
  - ImageNet：一般物体認識のデータセット
  - Places2：風景が増のデータセット
  - CelebA：セレブの顔画像データセット

ref. Guilin Liu, Fitsum A. Reda, Kevin J. Shih, Ting-Chun Wang, Andrew Tao, Bryan Catanzaro, "Image Inpainting for Irregular Holes Using Partial Convolutions", ECCV2018, https://arxiv.org/abs/1804.07723

出展

- P3
  - 顔画像検出：　ref. (2020/4/6): Wikimedia commons: File:Kasahara Saitama Kasahara Jinjo Elementary School 1920 1.jpg パブリックドメイン https://ja.wikipedia.org/wiki/%E3%83%95%E3%82%A1%E3%82%A4%E3%83%AB:Kasahara_Saitama_Kasahara_Jinjo_Elementary_School_1920_1.jpg
  - 一般物体認識：　ref. (2020/4/6): Photo by suzukii xingfu from Pexels https://www.pexels.com/photo/crowded-street-with-cars-passing-by-708764/
  - 画像認識：　ref. (2020/4/6): Wikimedia commons: File:“ 10 Alfa Romeo Giulietta white Derivate cut.JPG CC0 https://commons.wikimedia.org/wiki/Category:CC-Zero?uselang=ja#/media/File:%22_10_Alfa_Romeo_Giulietta_white_Derivate_cut.JPG
  - 人物姿勢推定：　ref. (2020/4/6): Photo by Yogendra Singh from Pexels https://www.pexels.com/ja-jp/photo/1701194/