画像認識分野における
deep learning の発展と最新動向

東京大学 大学院情報理工学系研究科
創造情報学専攻 講師
中山 英樹

本日の内容

- １．画像認識分野におけるdeep learningの歴史
- ２．一般画像認識：Deep learning 以前と以後で何が変わったか
  - Bag-of-visual-words (VLAD, Fisher Vector)
  - Convolutional neural network (ConvNets)
- ３．最新の動向・今後の展望
  - ILSVRC 2014
  - さらに高度な知能へ
- ４．実践するにあたって
  - 適切に利用するために必要な知識
  - 汎用ソフトウェア：Caffe

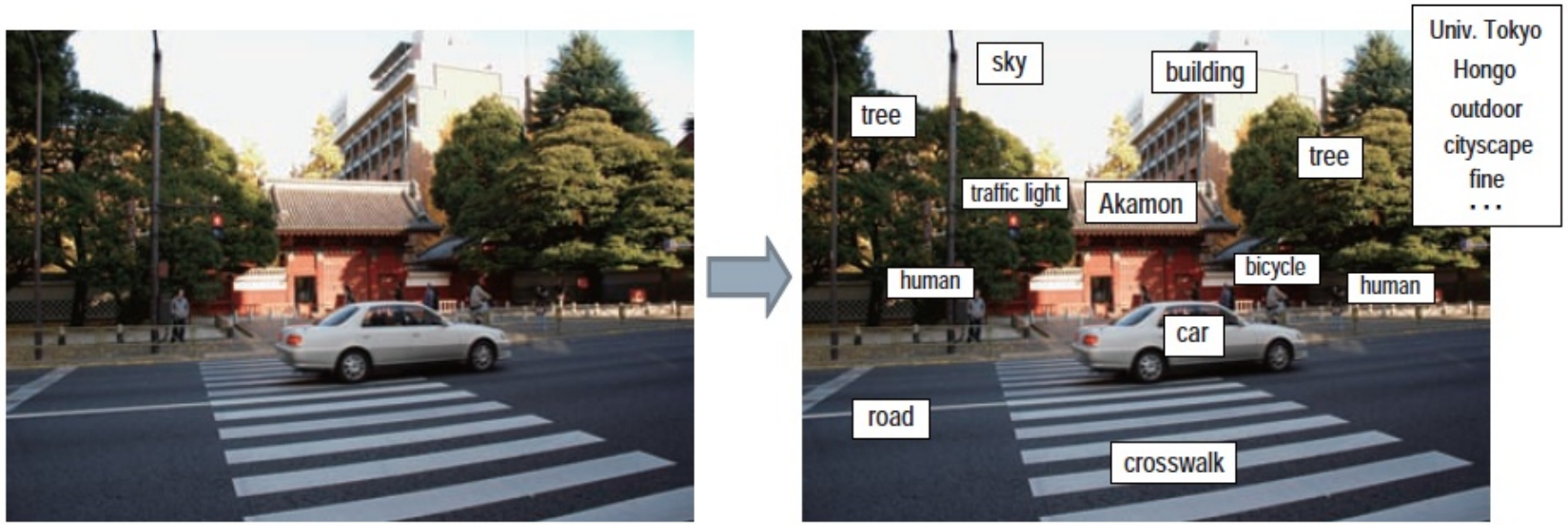
本日の内容

- １．画像認識分野におけるdeep learningの歴史
- ２．一般画像認識：Deep learning 以前と以後で何が変わったか
  - Bag-of-visual-words（VLAD, Fisher Vector)
  - Convolutional neural network (ConvNets)
- ３．最新の動向・今後の展望
  - ILSVRC 2014
  - さらに高度な知能へ
- ４．実践するにあたって
  - 適切に利用するために必要な知識
  - 汎用ソフトウェア：Caffe

一般画像認識（一般物体認識）

- 制約をおかない実世界環境の画像を単語で記述
  - 一般的な物体やシーン、形容詞（印象語）
  - 2000年代以降急速に発展（コンピュータビジョンの人気分野）
  - 幅広い応用先

    デジタルカメラ、ウェアラブルデバイス、画像検索、ロボット、…



一般画像認識の主要なタスク

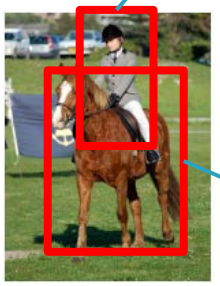
易

難

- **Categorization (カテゴリ識別)**
  - 映ってる物体の名前だけ
  - 物体の位置を答える必要はない

horse
human

- **Detection (物体検出)**
  - 矩形で物体の位置を切り出す



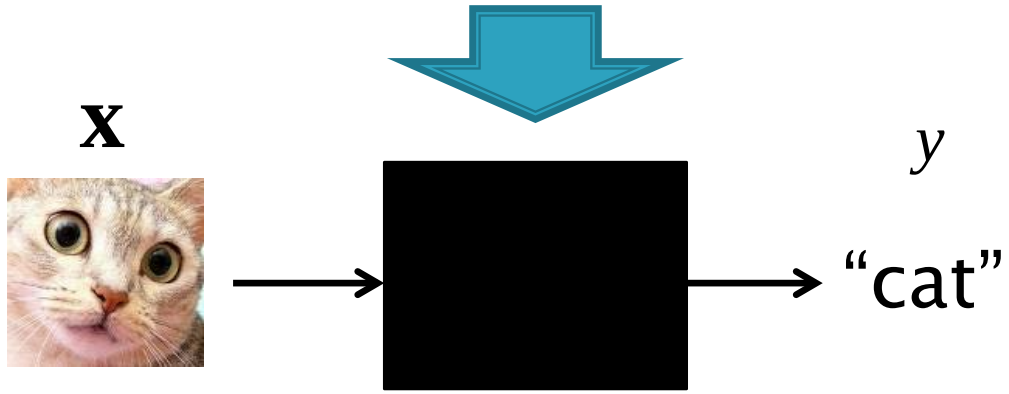
- **Semantic Segmentation**
  - ピクセルレベルで物体領域を認識

コンピュータに"学習"させる

- **機械学習（教師付）**

$$\{(\mathbf{x}_i, y_i), i = 1, \ldots, N\}$$

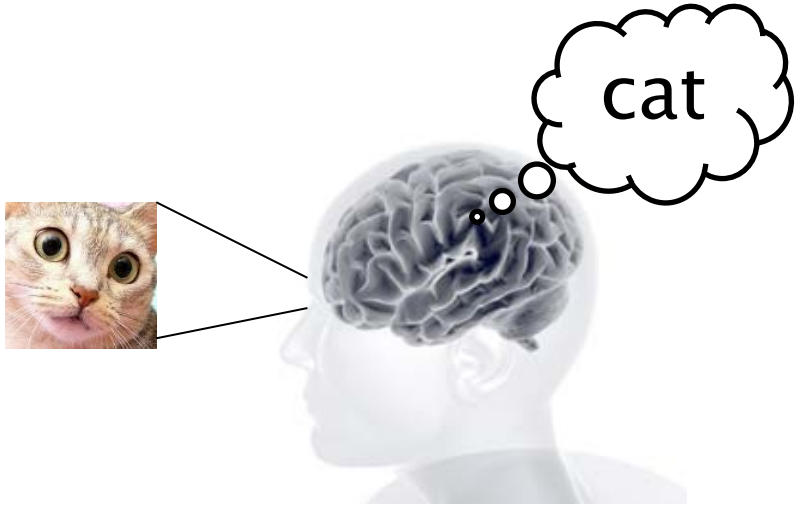
大量のラベル付き訓練データ
(x:画像，y:ラベル)

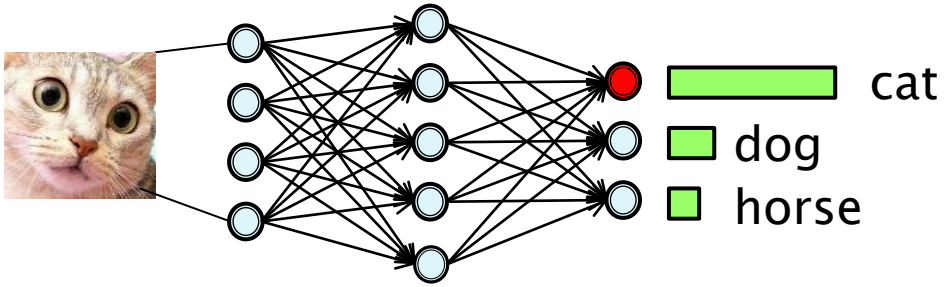


未知のデータ（学習データに含まれない）を正しく認識させることが目標

Deep learning（深層学習）

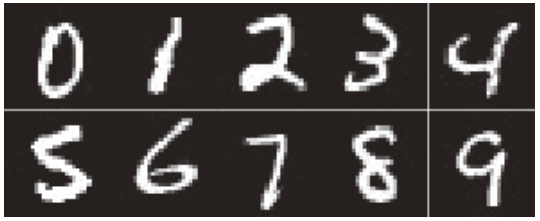
- ニューラルネットワークを用いた人工知能の構築技術の総称
  - 脳（神経細胞）の働きを模した学習アルゴリズム
- 特に、深く大規模な構造を備えていることが特徴





ニューラルネットと画像認識：2010年頃まで

- 小さな画像を用いた基礎研究が主流
  - MNISTデータセット [LeCun]
    - 文字認識、28 x 28ピクセル、6万枚



  - CIFAR-10/100 データセット [Krizhevsky]
    - 物体認識、32 x 32ピクセル、5万枚

- 機械学習のコミュニティで地道に発達
  - ビジョン系ではあまり受け入れられず…

CVPR 2012- One Author's Withdrawal Statement

"We are withdrawing it for three reasons: 1) the scores are so low, and the reviews so ridiculous, that I don't know how to begin writing a rebuttal without insulting the reviewers; 2) we prefer to submit the paper to ICML where it might be better received. （中略）

Getting papers about feature learning accepted at vision conference has always been a struggle, and I 've had more than my share of bad reviews over the years. Thankfully, quite a few of my papers were rescued by area chairs. （中略）

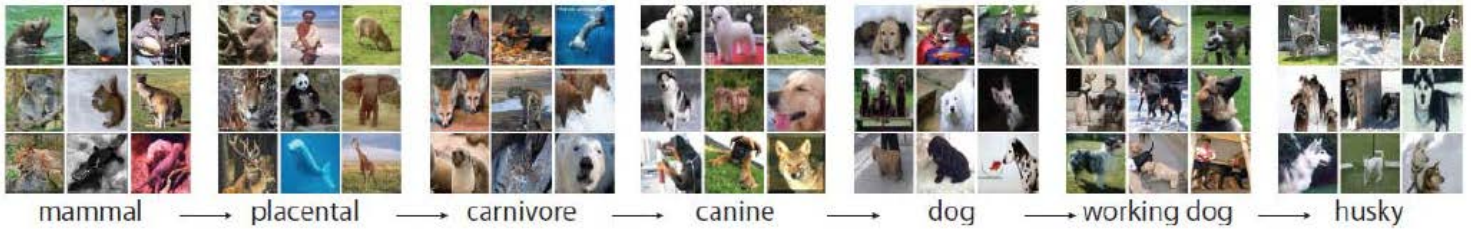
This time though, the reviewers were particularly clueless, or negatively biased, or both. （中略）

So, I 'm giving up on submitting to computer vision conferences altogether. CV reviewers are just too likely to be clueless or hostile towards our brand of methods. Submitting our papers is just a waste of everyone's time （中略）

Regardless, I actually have a keynote talk at [Machine Learning Conference], where I'll be talking about the results in this paper."

ImageNet Large-scale Visual Recognition Challenge (ILSVRC)

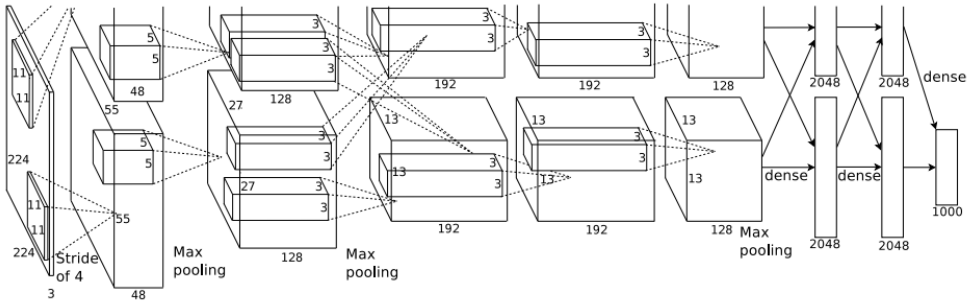
- **ImageNetのデータの一部を用いたフラッグシップコンペティション (2010年より開催)**
  - **ImageNet [Deng et al., 2009]**
    - クラウドソーシングにより構築中の大規模画像データセット
    - **1400万枚、2万2千カテゴリ**（WordNetに従って構築）



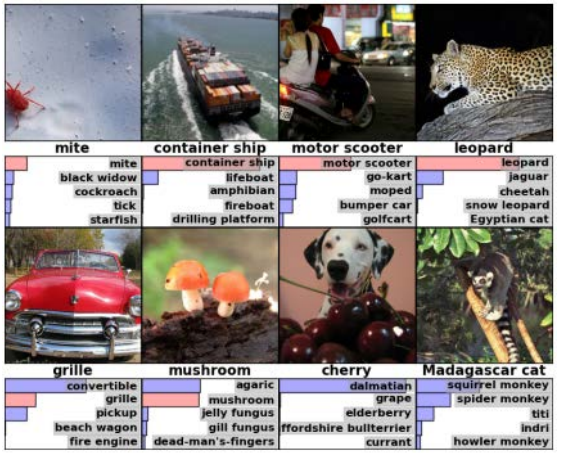
- **コンペでのタスク**
  - **1000クラスの物体カテゴリ分類**
    - 学習データ120万枚、検証用データ5万枚、テストデータ10万枚
  - **200クラスの物体検出**
    - 学習データ45万枚、検証用データ2万枚、テストデータ4万枚

ILSVRC 2012 での衝撃

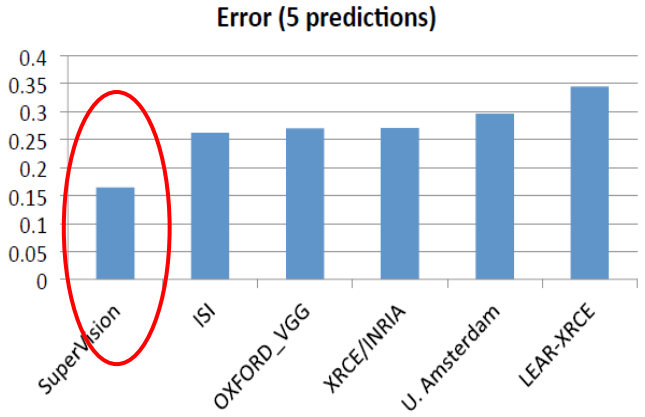
- 1000クラス識別タスクで、deep learning を用いたシステムが圧勝
  - トロント大学Hinton先生のチーム (AlexNet)



[A. Krizhevsky *et al.*, NIPS'12]

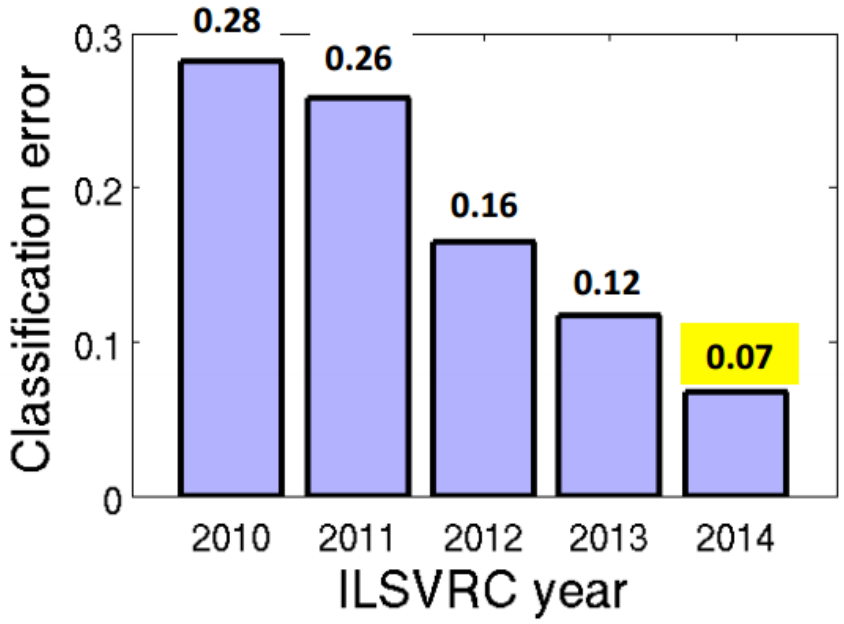


エラー率が一気に10%以上減少！
(※過去数年間での向上は1~2%)



その後も急激な性能向上が続く

- エラー率が 0.16 (2012) → 0.07 (2014)



http://www.image-net.org/challenges/LSVRC/2014/slides/ILSVRC2014_09_12_14_det.pdf

Russakovsky et al., “ImageNet Large Scale Visual Recognition Challenge”, 2014.

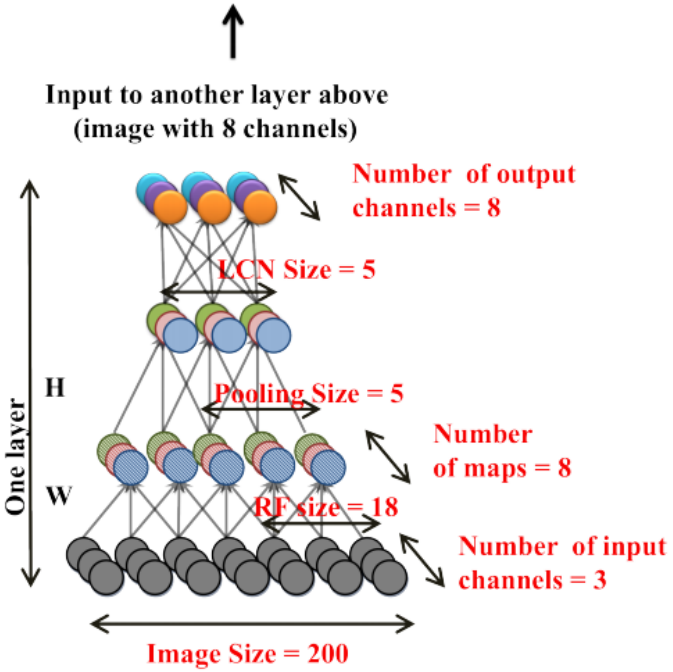
デモサイトなど

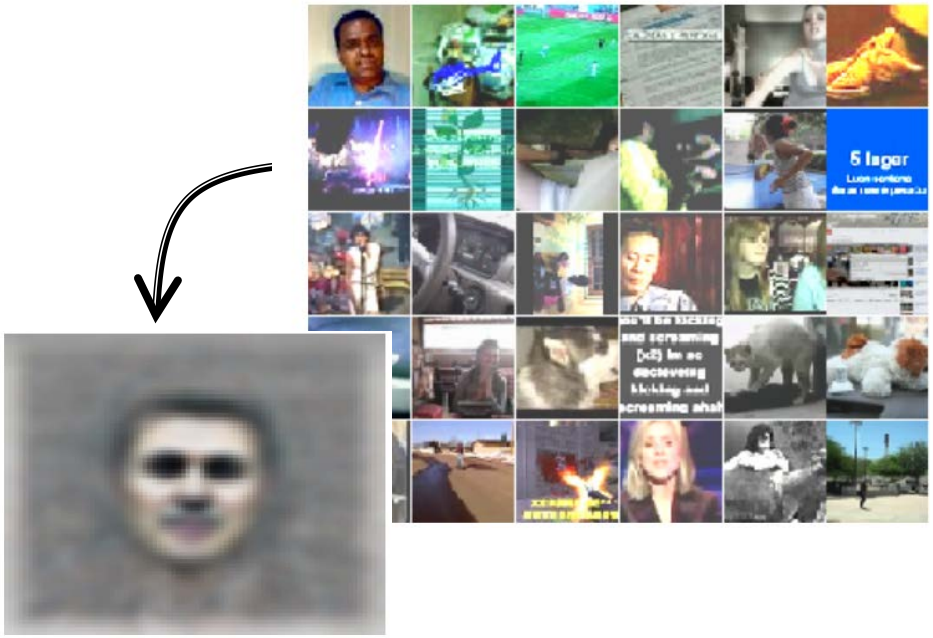
- Clarifai
  - ILSVRC 2013優勝者 (NY大)が作ったベンチャー
  - http://www.clarifai.com/
- Euvision (Qualcommが買収)
  - Impala : スマートフォンアプリ
- トロント大学DLグループ
  - 画像識別に加え、説明文生成もサポート
  - iPhone、Andoroidアプリも
  - http://deeplearning.cs.toronto.edu/

画像特徴の教師なし学習

Le et al., “Building High-level Features Using Large Scale Unsupervised Learning”, ICML’12.

- 9層のautoencoder
- 1000万枚のYouTube画像から教師なし事前学習
- これを初期状態として識別的学習を始めることで識別精度向上



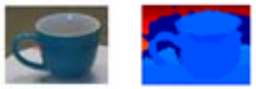


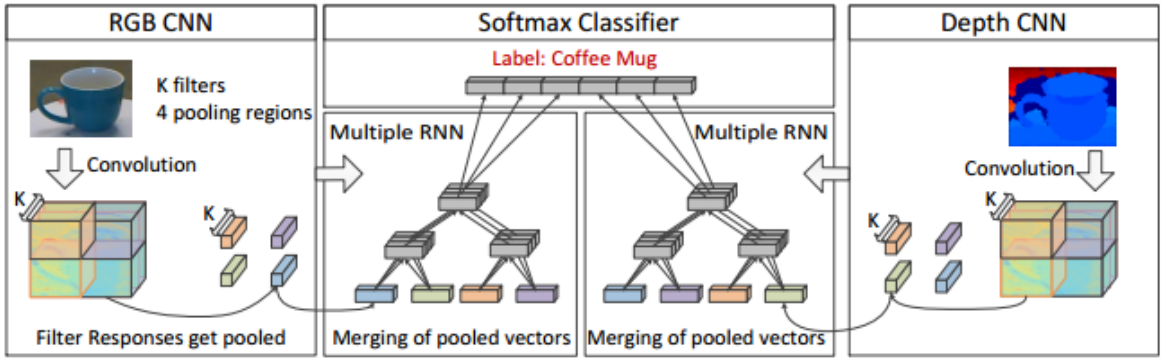
人の顔に特異的に反応するニューロンが自動的に獲得された（他、猫なども）
≒“おばあちゃんニューロン”？

他の画像認識タスク

- 映像 認識
  - 487クラスの
    スポーツカテゴリ認識
    [Karpathy., CVPR'14]

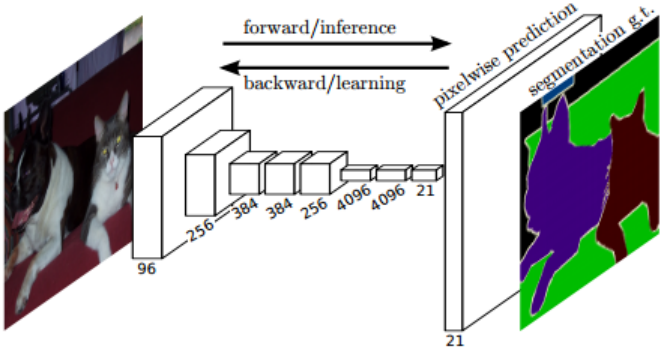
- RGB-D物体認識
  - [Socher et la., NIPS'13]



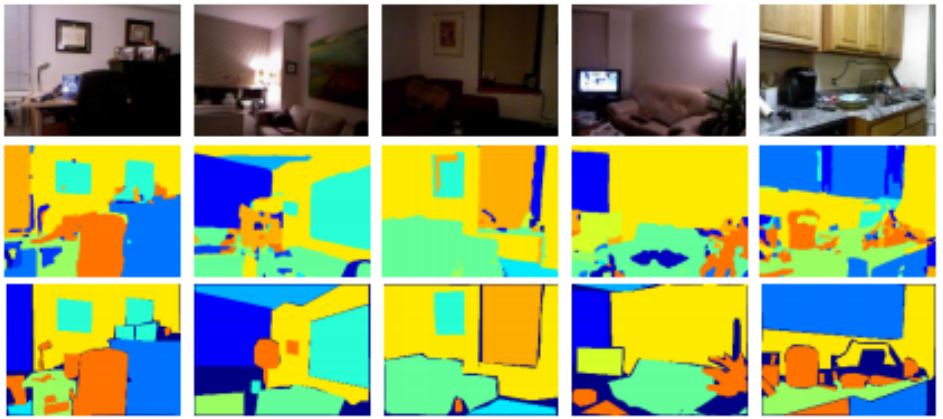


他の画像認識タスク

- 領域分割 （シーンラベリング）
  - ピクセルレベルで物体領域を認識
  - [Long et al., 2014]

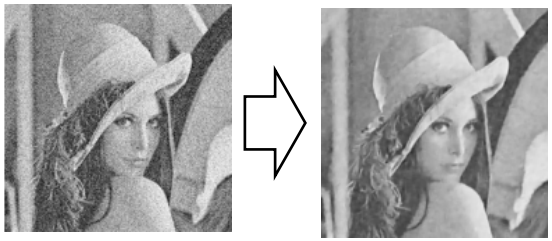


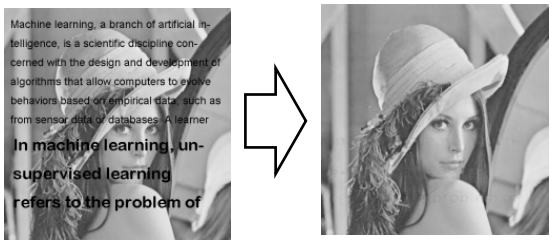
- RGB-Dシーンラベリング
  - [Wang et al., ECCV'14]



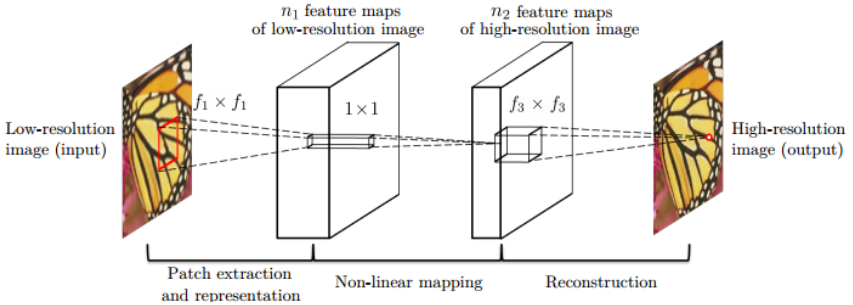
画像処理の諸分野でも広い応用

- デノイジング・インペインティング [Xie et al., NIPS'12]
  - 画像のノイズ除去
  - Stacked denoising auto-encoder





- 超解像 [Dong et al., ECCV'14]
  - 低解像度画像から
    高解像度画像を復元（推定）



- ボケ補正 [Xu et al., NIPS'14]

本日の内容

- １．画像認識分野におけるdeep learningの歴史
- ２．一般画像認識：Deep learning 以前と以後で何が変わったか
  - Bag-of-visual-words（VLAD, Fisher Vector)
  - Convolutional neural network (ConvNets)
- ３．最新の動向・今後の展望
  - ILSVRC 2014
  - さらに高度な知能へ
- ４．実践するにあたって
  - 適切に利用するために必要な知識
  - 汎用ソフトウェア：Caffe

画像認識パイプラインの変化

人手で設計

人手で設計/教師なし学習

伝統的方法論
("Shallow" learning)

Low-level image feature

Mid-level image feature

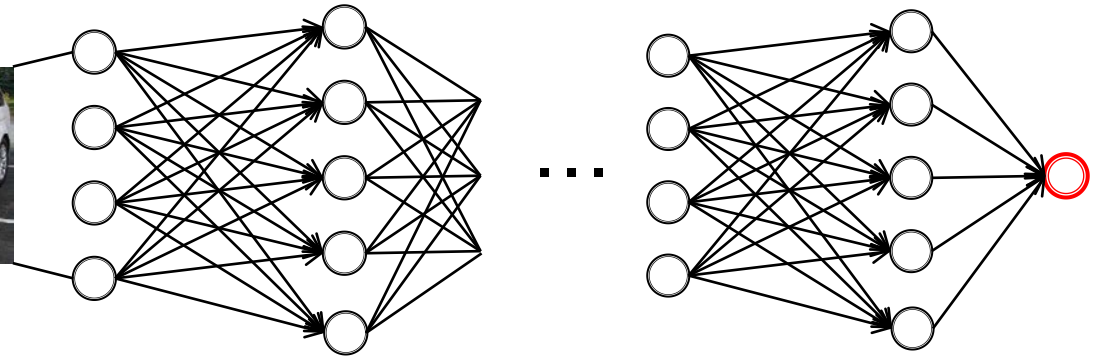
Supervised Classifier: SVM, Logistic Regression, etc.

"Car"

SIFT, HOG, SURF, etc.

BoVW, VLAD, Fisher Vector, etc.

Deep learning



生の画素値から、識別に至る階層構造を直接的に学習

画像認識パイプラインの変化

人手で設計　　人手で設計/教師なし学習

伝統的方法論
("Shallow" learning)

Low-level image feature

SIFT, HOG, SURF, etc.

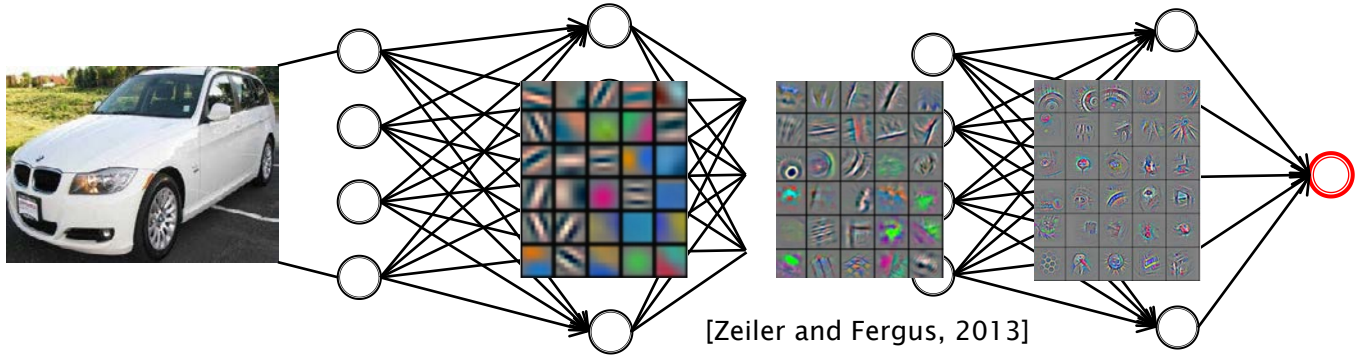
Mid-level image feature

BoVW, VLAD, Fisher Vector, etc.

Supervised Classifier: SVM, Logistic Regression, etc.

"Car"

Deep learning



[Zeiler and Fergus, 2013]

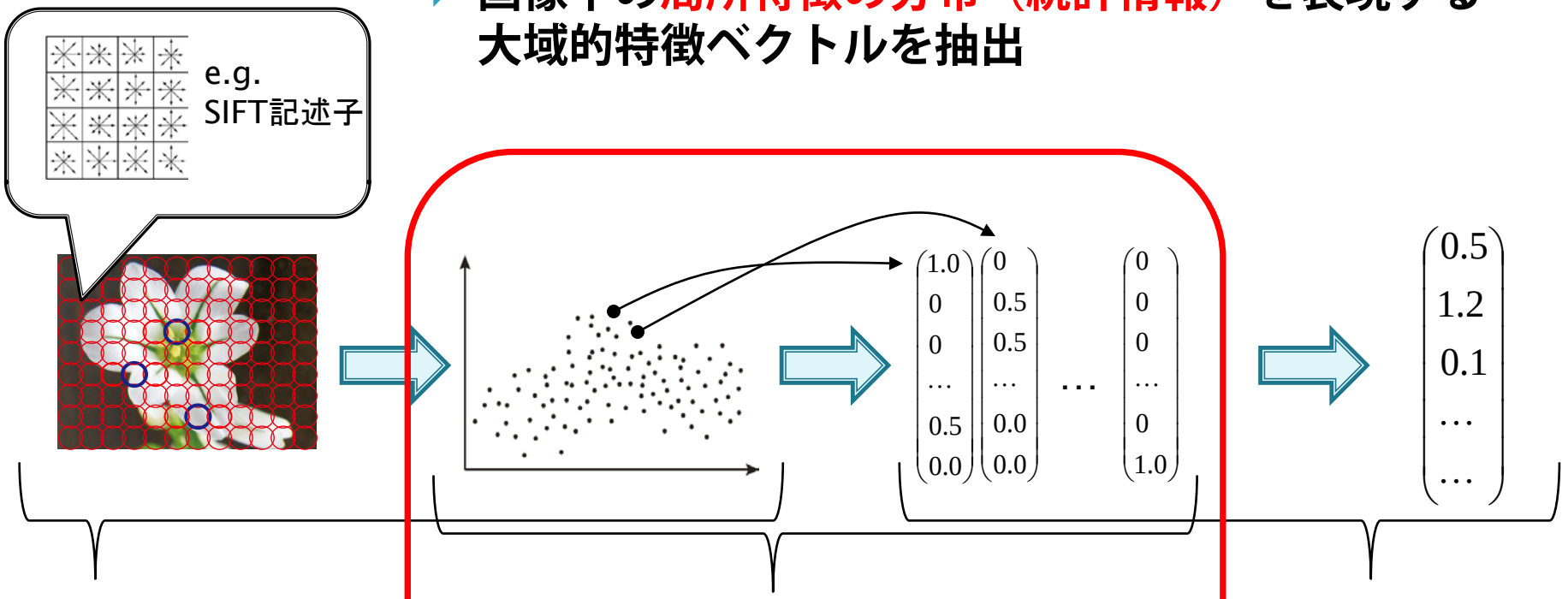
"Car"

生の画素値から、識別に至る階層構造を直接的に学習

従来の特徴量に相当する構造が中間層に自然に出現

深層学習以前の画像特徴抽出の枠組

- 画像中の局所特徴の分布（統計情報）を表現する大域的特徴ベクトルを抽出



1. 局所特徴抽出

- SIFT, SURF, HOG, etc.
- Dense sampling
  （回転、スケールの正規化なし）

2. エンコーディング

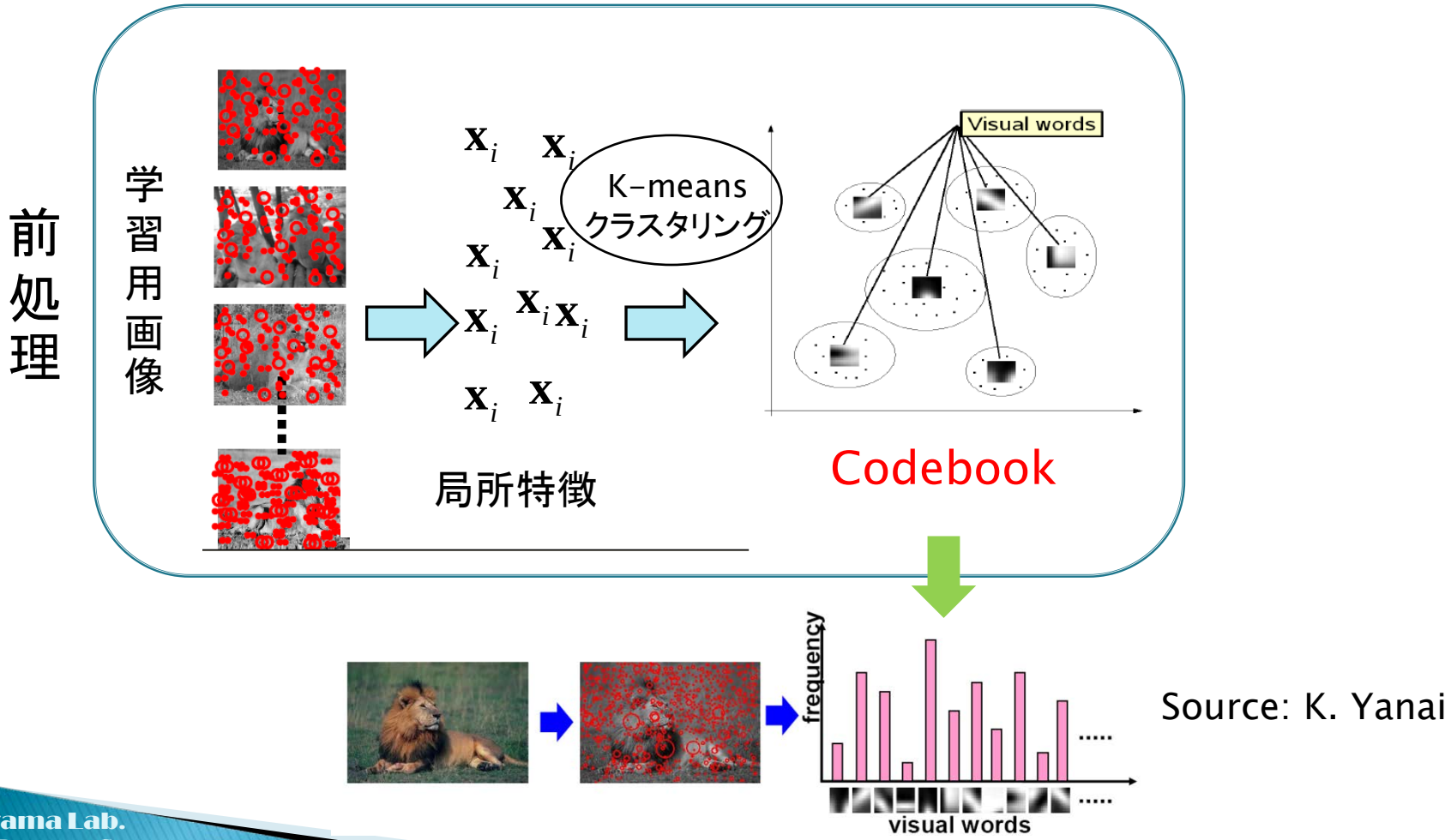
- ベクトル量子化
- 多項式特徴（要素積）

3. プーリング

- 最大値プーリング
- 平均値プーリング

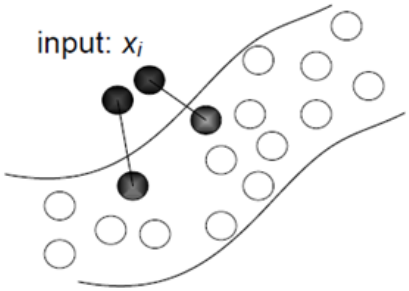
Bag-of-Visual-Words (BoVW) [Csurka et al. 2004]

- ベクトル量子化により局所特徴のヒストグラムを作成



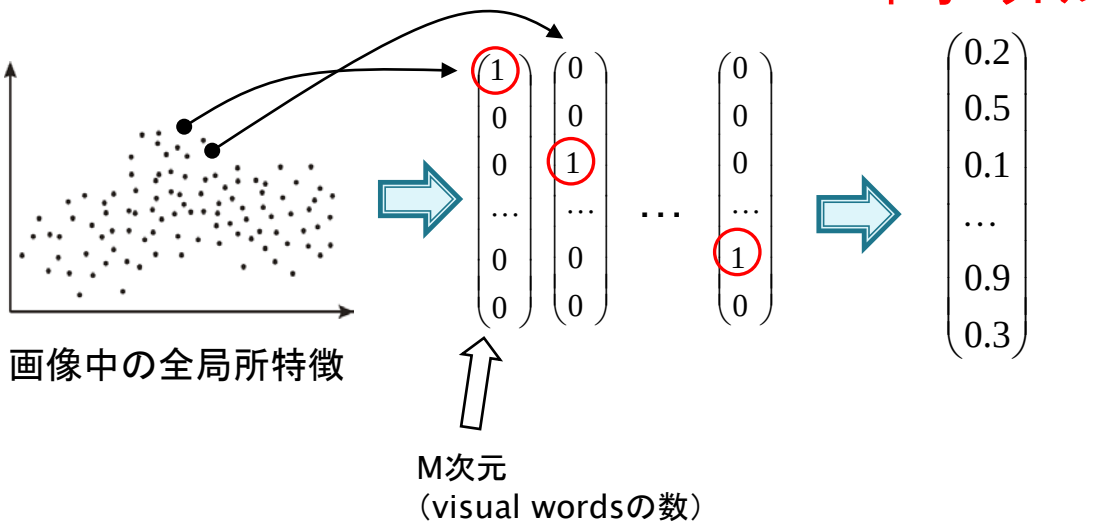
BoVW (contd.)

- つまり…
  - 最近傍のvisual wordに対応するコードに対してのみ1、
    それ以外に0を埋める最も単純な局所特徴エンコーディング
  - 平均値プーリング



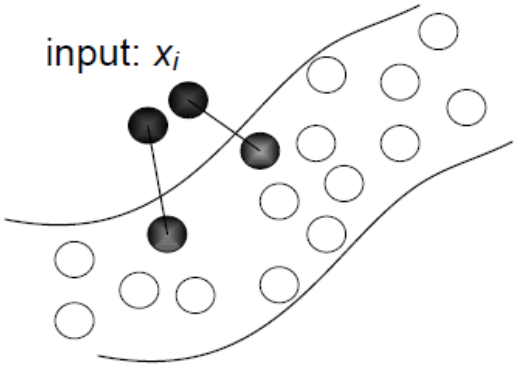
$$\arg\min_{\mathbf{C}} \sum_{i=1}^{N} ||\mathbf{x}_i - \mathbf{B}\mathbf{c}_i||^2$$
$$s.t. \|\mathbf{c}_i\|_{\ell^0} = 1, \|\mathbf{c}_i\|_{\ell^1} = 1, \mathbf{c}_i \succeq 0, \forall i$$

[Wang et al., CVPR'10]



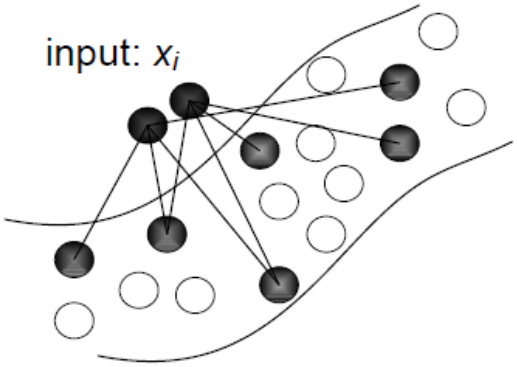
BoVWの発展①　スパースコーディング

- ベクトル量子化のreconstruction error を低減させる
  - 局所特徴の空間はサンプル数の割に高次元
  - 複数の基底(visual words)を用いてエンコーディング
  - 最大値プーリングと合わせて用いる場合が多い



VQ

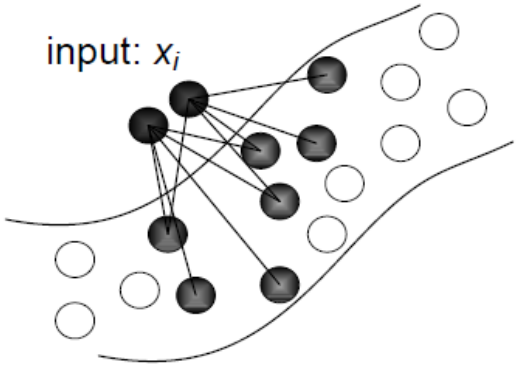
$$\arg\min_{\mathbf{C}} \sum_{i=1}^{N} ||\mathbf{x}_i - \mathbf{B}\mathbf{c}_i||^2$$
$$s.t. \|\mathbf{c}_i\|_{\ell^0} = 1, \|\mathbf{c}_i\|_{\ell^1} = 1, \mathbf{c}_i \succeq 0, \forall$$



SC

$$\arg\min_{\mathbf{C}} \sum_{i=1}^{N} \|\mathbf{x}_i - \mathbf{B}\mathbf{c}_i\|^2 + \lambda\|\mathbf{c}_i\|_{\ell^1}$$

[Yang et al., CVPR'09]

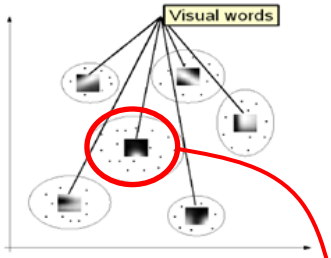


LLC

$$\min_{\mathbf{C}} \sum_{i=1}^{N} \|\mathbf{x}_i - \mathbf{B}\mathbf{c}_i\|^2 + \lambda||\mathbf{d}_i \odot \mathbf{c}_i||^2$$
$$s.t.\ \mathbf{1}^\top \mathbf{c}_i = 1,\ \forall i$$

[Wang et al., CVPR'10]

BoVWの発展②　高次統計量の密な利用



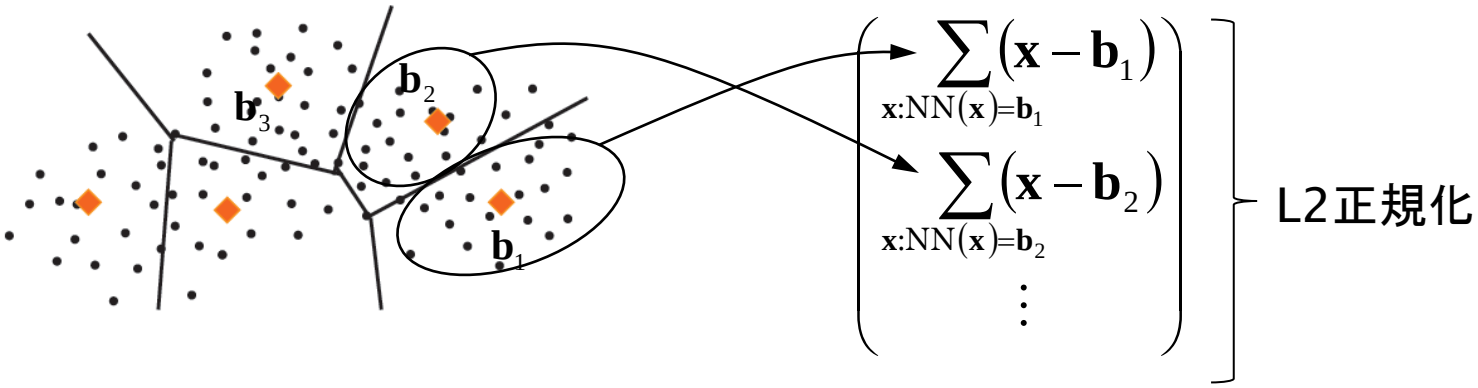
M:　visual wordの数
d:　局所特徴量の次元数

手法	統計量	特徴ベクトルの次元数
BoVW	個数（割合）	M
VLAD [Jegou+,CVPR'10]	平均	Md
Super vector [Zhou+, ECCV'10]	割合＋平均	M(d+1)
Fisher vector [Perronnin+, ECCV'10]	平均＋分散	2Md
Global Gaussian [Nakayama+, CVPR'10]	平均＋分散共分散	d(d+1)/2 (M=1)
VLAT [Picard+ ICIP'11]	平均＋分散共分散	Md(d+1)/2

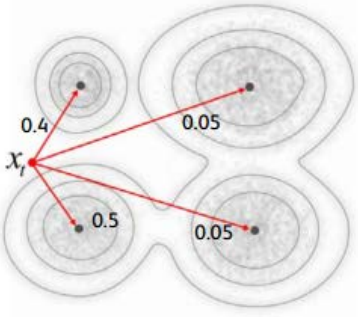
基本的には、**局所特徴分布のさまざまな統計量**を素性として特徴ベクトル化していると解釈できる

BoVWの発展②

- **VLAD** [Jegou et al., CVPR'10]
  - 各visual wordに属する局所特徴の平均ベクトルを列挙
  - 1次の統計量



- **Fisher vector** [Perronnin et al., ECCV'10]
  - 平均に加え、分散も利用
  - 混合正規分布と情報幾何を用いたエンコーディング
  - 1次, 2次の統計量



- gradient wrt to μ and σ

$$\mathcal{G}^X_{\mu,i} = \frac{1}{T\sqrt{w_i}}\sum_{t=1}^{T}\gamma_t(i)\left(\frac{x_t-\mu_i}{\sigma_i}\right)$$

$$\mathcal{G}^X_{\sigma,i} = \frac{1}{T\sqrt{2w_i}}\sum_{t=1}^{T}\gamma_t(i)\left[\frac{(x_t-\mu_i)^2}{\sigma_i^2}-1\right]$$

http://people.rennes.inria.fr/Herve.Jegou/courses/2012_cpvr_tutorial/4-new-patch-agggregation.pptx.pdf

BoVWの発展②

- いずれも、**多項式特徴を用いたエンコーディング＋平均値プーリング**であると解釈できる

- VLAD
  - 各局所特徴 $\mathbf{x}^i \in R^d$ について

$$\mathbf{x}^i \Rightarrow \Big(\underbrace{0\,0\,0\ldots 0}_{\text{d個}}\ \underbrace{0\,0\,0\ldots 0}_{\text{d個}}\ \underbrace{x_1^i - b_1^c\ x_2^i - b_2^c \ldots x_d^i - b_d^c}_{\text{最近傍のvisual word（c番目とする）との差分}}\ \underbrace{0\,0\,0\ldots 0}_{\text{d個}}\ \cdots\Big)^T$$

- Fisher vector
  - 同様に、x_k^i に加え $\left(x_k^i\right)^2$ の項を対応するvisual wordの場所へ列挙
    （※厳密には、Fisher情報行列による変換が入る）

BoVWの各手法の比較

- Fisher vectorなど、高次統計量（多項式特徴によるエンコーディング）を利用した手法が強力

Fisher vector

Method		256	codebook size 600	1500	2000	4000	8000
(a) FK	Lin	77.78 ± 0.56	–	–	–	–	–
(b) LLC	Lin	–	73.10 ± 1.09	74.84 ± 0.67	75.75 ± 0.71	76.15 ± 0.59	76.95 ± 0.39
(c) LLC	Chi	–	72.30 ± 1.08	74.23 ± 0.62	75.24 ± 0.71	75.95 ± 0.57	76.62 ± 0.61
(d) VQ	Chi	–	72.65 ± 0.77	73.62 ± 0.51	73.93 ± 0.79	74.41 ± 1.04	74.23 ± 0.65
(e) KCB	Chi	–	73.38 ± 0.65	75.24 ± 0.63	75.50 ± 0.65	75.92 ± 0.63	75.93 ± 0.57

Table 2: Image classification results on Caltech-101 dataset (30 training images)

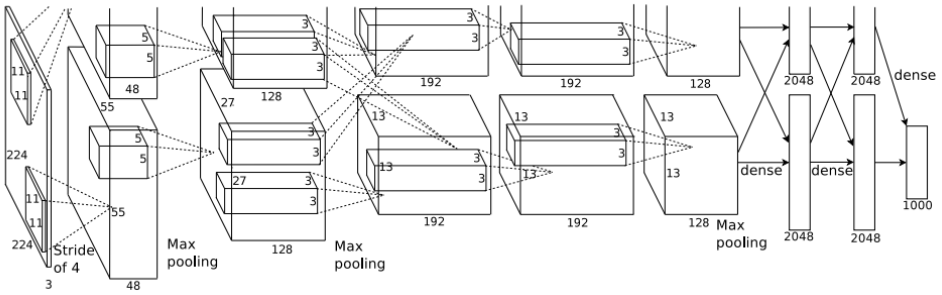
K. Chatfield, V. Lempitsky, A. Vedaldi, A. Zisserman, “The devil is in the details: an evaluation of recent feature encoding methods”, In Proc. BMVC, 2011.

- ただし、非常に高次元な特徴ベクトルとなる
  - 例えば、ILSVRC’11で用いられたシステムでは
    (64 + 64) * 256 * 8 = 262,144次元

平均　分散　Visual words数 (GMMの混合数)　領域数

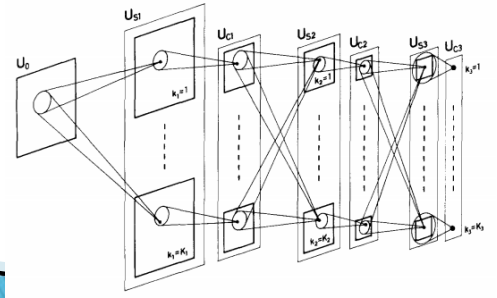
Deep learning ブレイク後の画像認識

- 畳み込みニューラルネットワーク
  - 脳の視覚野の構造を模倣した多層パーセプトロン
  - ニューロン間の結合を局所に限定（パラメータ数の大幅な削減）



[A. Krizhevsky et al., NIPS'12]

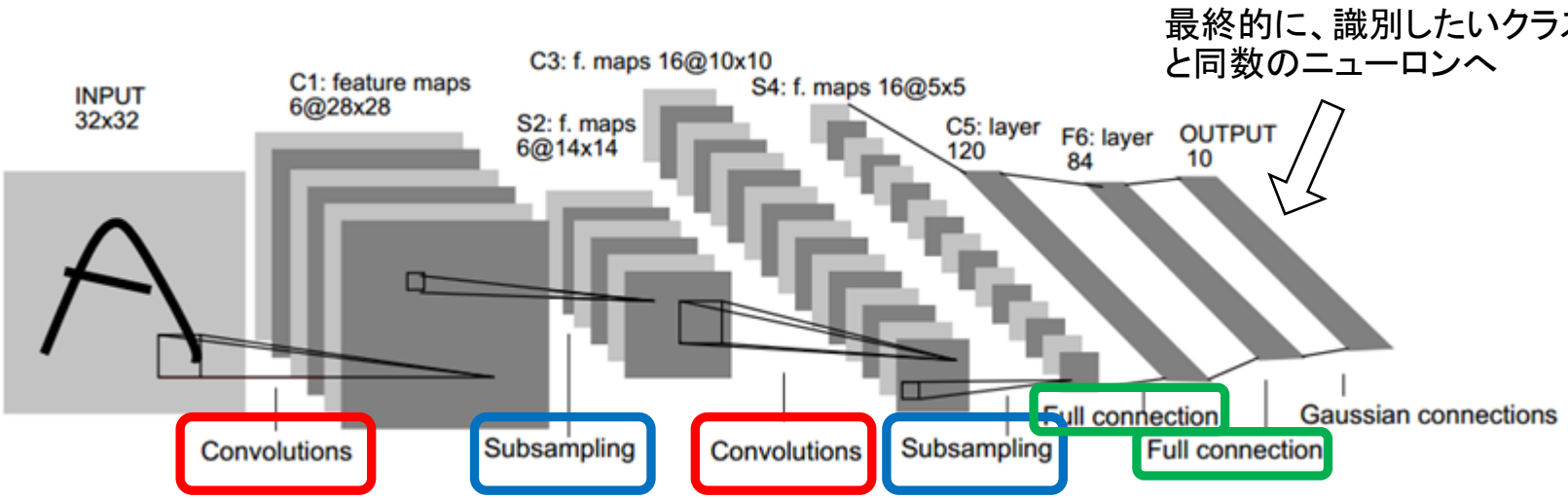
- 最初に基本構造が提案されたのは実はかなり昔
  - ネオコグニトロン （福島邦彦先生、1980年代前後)



Kunihiko Fukushima, "Neocognitron: A Self-organizing Neural Network Model for a Mechanism of Pattern Recognition Unaffected by Shift in Position", Biological Cybernetics, 36(4): 93-202, 1980.

Convolutional neural network (ConvNet)

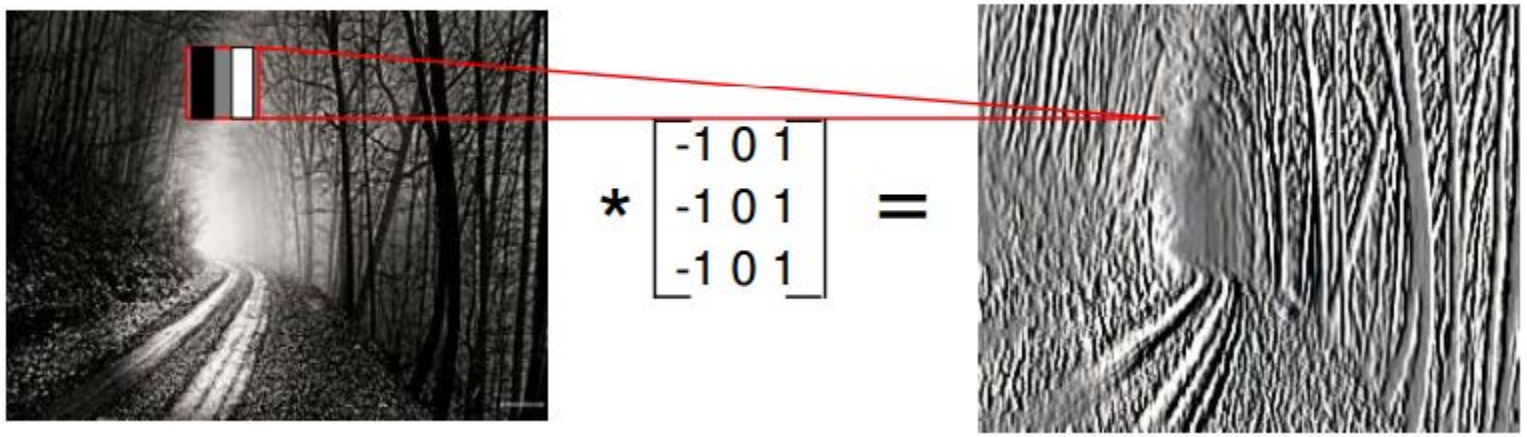
- 局所領域（受容野）の畳み込みとプーリングを繰り返す多層ネットワーク
  - 段階的にスケールを変えながら、局所的な相関パターンを抽出
  - プーリングにより、局所的な平行移動不変性を確保



Y. LeCun, L. Bottou, Y. Bengio and P. Haffner: Gradient-Based Learning Applied to Document Recognition, Proceedings of the IEEE, 86(11):2278-2324, 1998.

畳み込み

- 一般的なフィルタだと…
  - 例）エッジ抽出

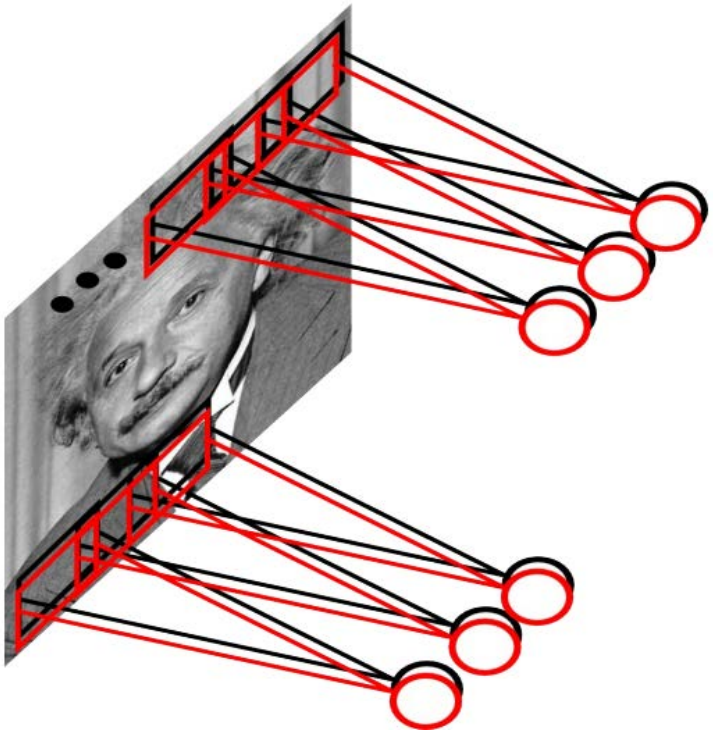


- 識別に有効なフィルタ（カーネル）をデータから学習
  - 係数をチューニング

Source: M. Ranzato, CVPR'14 tutorial slides

畳み込み層

- 色の違いは異なる畳み込みフィルタを示す
  - 各フィルタのパラメータは全ての場所で共有



※もちろん入力は生画像のみ
とは限らない（中間層など）

非線形活性化関数（とても重要）

$$r = \phi(\mathbf{w} * \mathbf{h} - \theta)$$

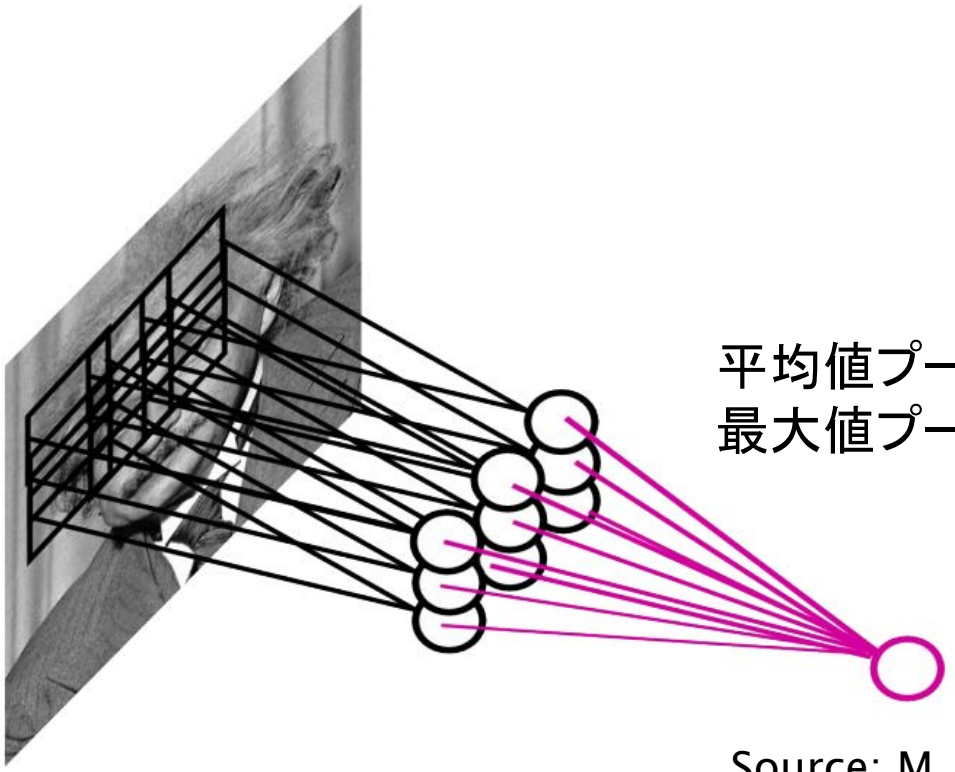
フィルタの係数　　入力　　バイアス

例えば、5x5の畳み込み、
10チャンネルの入力の場合、
5x5x10=250個

Source: M. Ranzato, CVPR'14 tutorial slides

プーリング層

- 一定領域内の畳み込みフィルタの反応をまとめる
  - 領域内での平行移動不変性を獲得



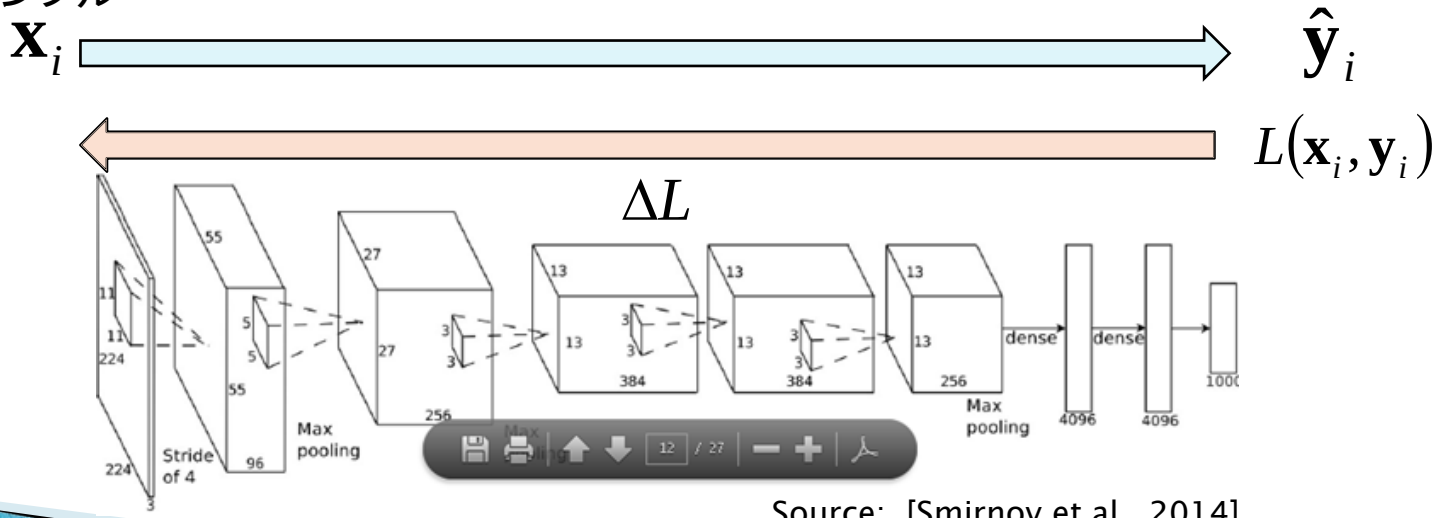
平均値プーリング、
最大値プーリングなど

Source: M. Ranzato, CVPR'14 tutorial slides

パラメータの学習

- フリーパラメータが存在するのは畳み込み層、全結合層
  - 大半は全結合層に集中
- 誤差逆伝播法で最適化
  - 実際には、ミニバッチ法で誤差をある程度まとめてパラメータを更新(100枚単位など)
- 初期値はランダムに与える場合が多い
  - ただし、大量の教師付データが必要

訓練サンプル

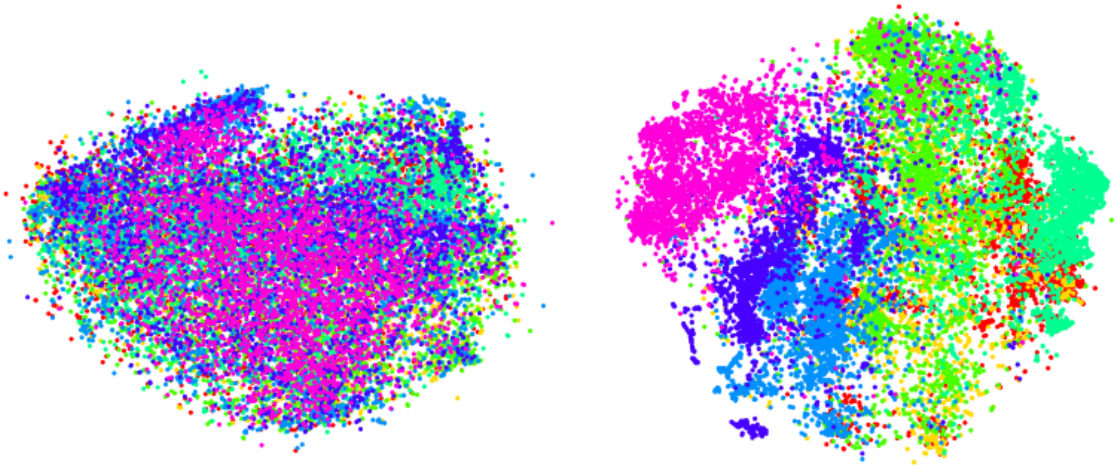


Source: [Smirnov et al., 2014]

各層のニューロンの出力

- 層を上るにつれ、クラスの分離性能が上がる

ILSVRC'12 の
validation data
（色は各クラスを示す）



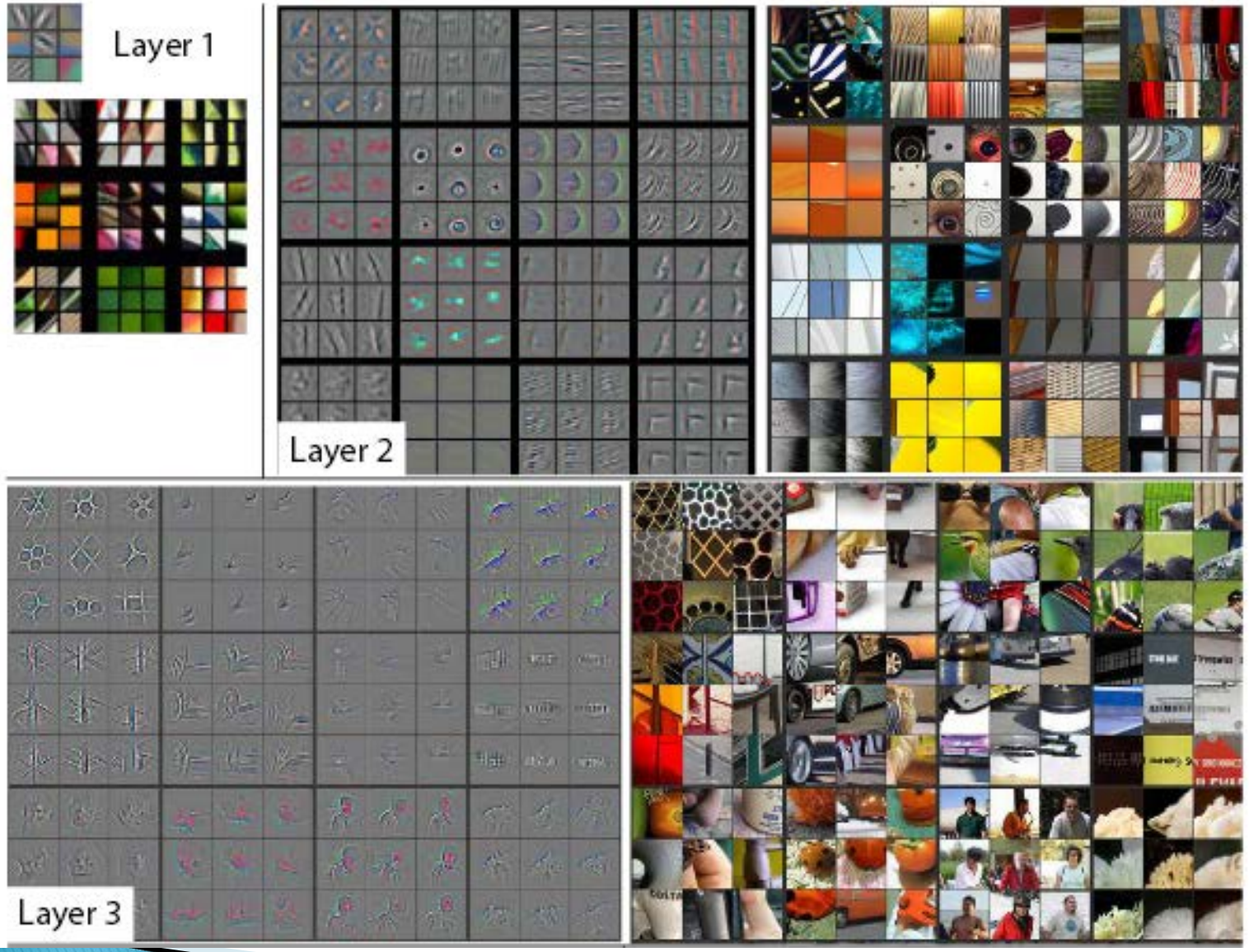
(c) DeCAF_1
第1層

(d) DeCAF_6
第6層

J. Donahue et al., "DeCAF: A Deep Convolutional Activation Feature for Generic Visual Recognition", In Proc. ICML, 2014.

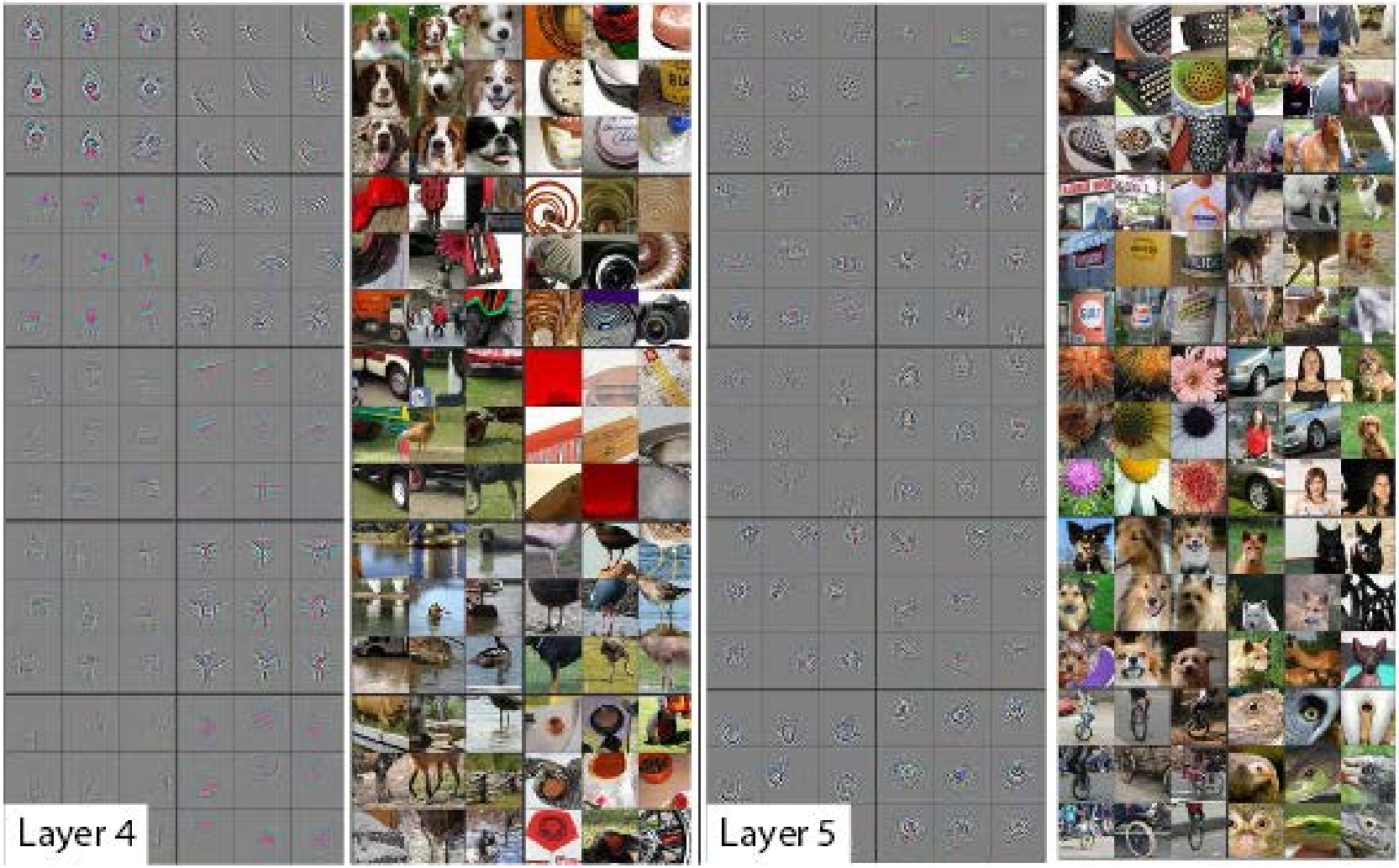
中間層の可視化

Matthew D. Zeiler and Rob Fergus, “Visualizing and Understanding Convolutional Networks”, In Proc. ECCV, 2014.



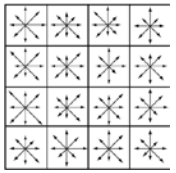
中間層の可視化

Matthew D. Zeiler and Rob Fergus, “Visualizing and Understanding Convolutional Networks”, In Proc. ECCV, 2014.



BoVW と ConvNet

- エンコーディング+プーリングの構造自体は同じ（活性化関数が違う）
  - 例えばSIFT-BoVWの場合、4x4の畳み込みと解釈できる
  - スパースコーディングに代表されるようなアサインメントの工夫は**活性化関数の工夫**と解釈できる



- BoVW

$$\underset{\mathcal{D},s}{\text{minimize}} \sum_i ||\mathcal{D}s^{(i)} - x^{(i)}||_2^2 + \lambda ||s^{(i)}||_1$$

$$\text{subject to } ||D^{(j)}||_2 = 1, \forall j.$$

A. Coates, A. Ng, “Learning Feature Representations with K-Means”, Neural Networks: Tricks of the Trade, pp.561-580, 2012.

⇐ 球面k-meansの場合

$$\text{エンコーディング}: \; s_j^{(i)} = \begin{cases} 1 & \text{if } j == \arg\max_l |\mathcal{D}^{(l)\top} x^{(i)}| \\ 0 & \text{otherwise.} \end{cases}$$

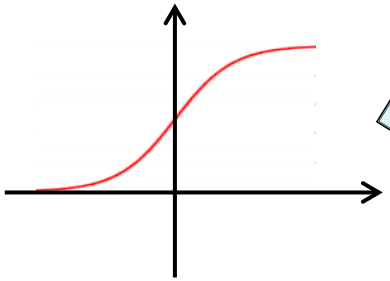
  - 線形射影＋非線形な活性をとる点で全く同じ
  - Coatesらのエンコーディング方法 (2012)

$$(f(x; \mathcal{D}, \alpha) = \max\{0, \mathcal{D}^\top x - \alpha\}, \text{ where } \alpha \text{ is a tunable constant})$$

これは，活性化関数にReLU(後述)を用いた場合の畳み込みに他ならない

ニューラルネットの活性化関数の変遷

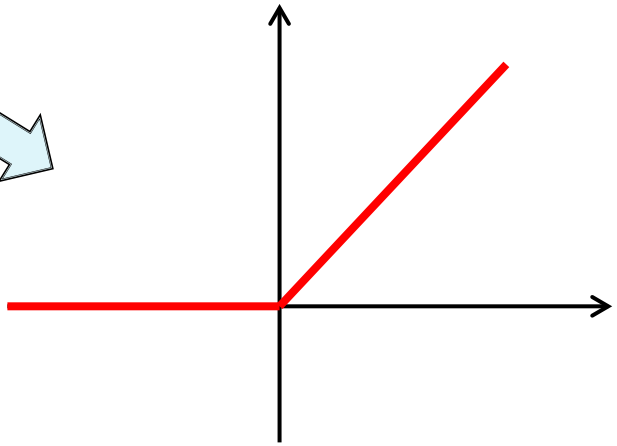
- **勾配が出やすいように関数の設計を工夫**
- **区分線形関数が良好な性能を示すことが分かってきた**



シグモイド関数

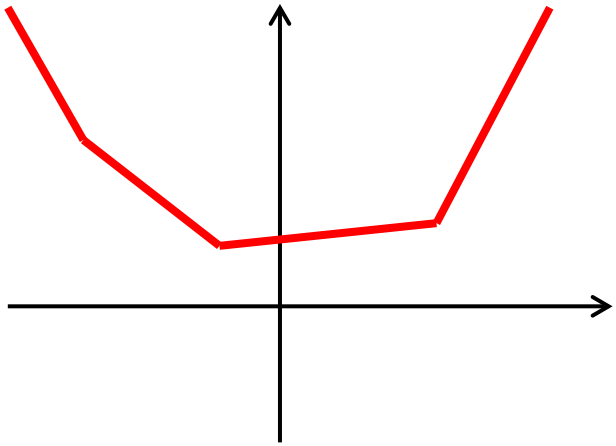
$$\frac{1}{1+\exp(-x)}$$

サチると勾配が出ない!



Rectified linear units (ReLU)
[Nair & Hinton, 2010]

$$\max(0, x)$$

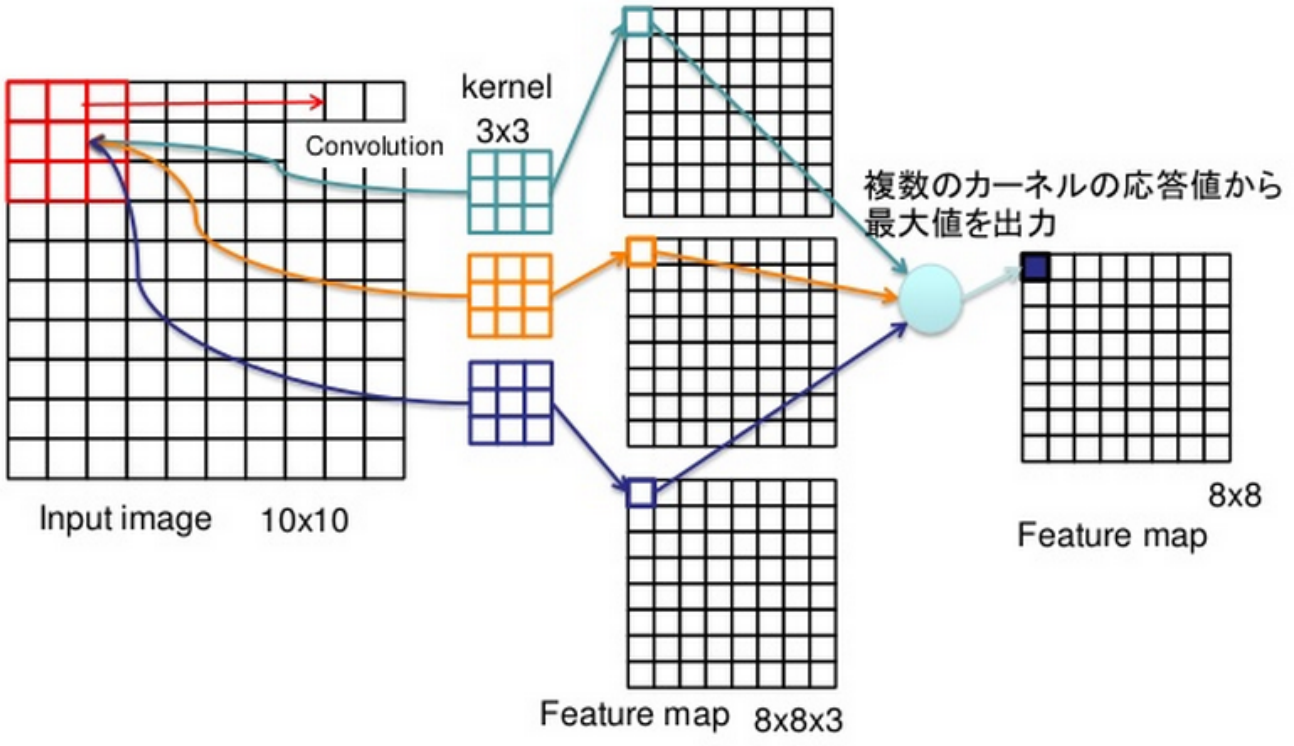


Maxout [Goodfellow, 2013]

多数の線形関数のmax
（任意の閾値関数を近似）

ConvNetにおけるMaxout

- 複数のカーネル（畳み込みフィルタ）を束ね、最大値をとる

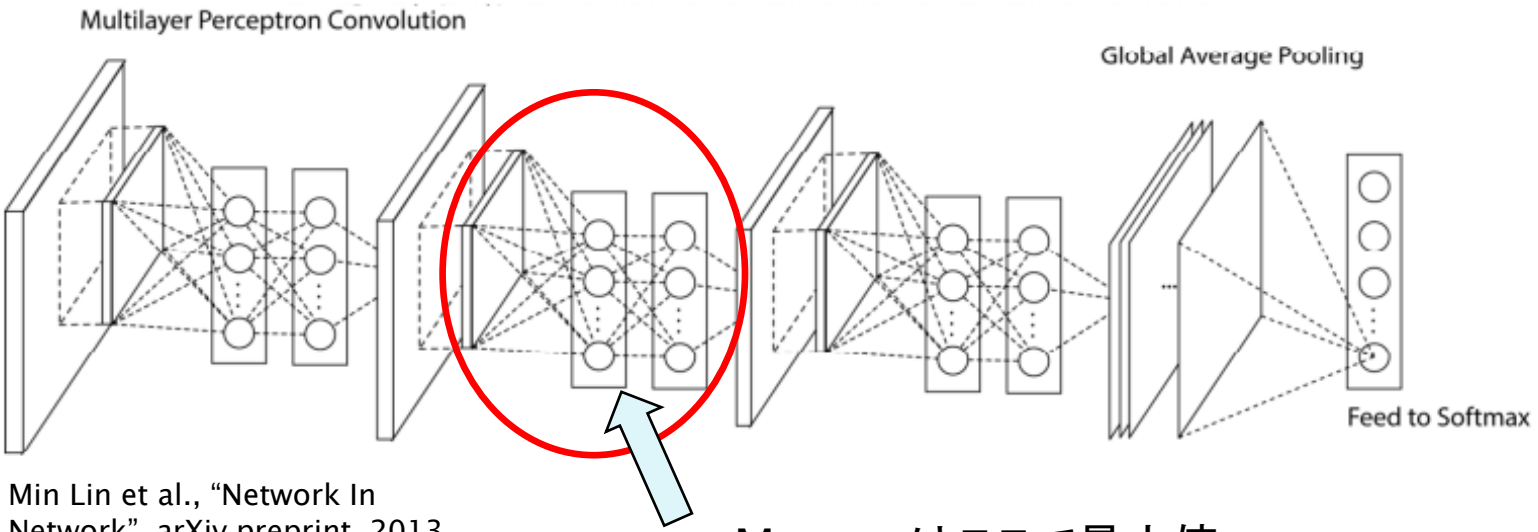


[山下隆義先生、MIRU'14 チュートリアルスライドより]

Network in network (NIN)

- 現在、画像認識において最も性能がよいアーキテクチャ
  - ILSVRC'14トップのGoogleチームもNINがベース

- 活性化関数自体を多層パーセプトロンで学習（Maxoutの一般化）
- 最後に全結合層をおかなくても十分な性能
  - 見た目は複雑だが実はパラメータ数は減っている



Min Lin et al., "Network In Network", arXiv preprint, 2013.

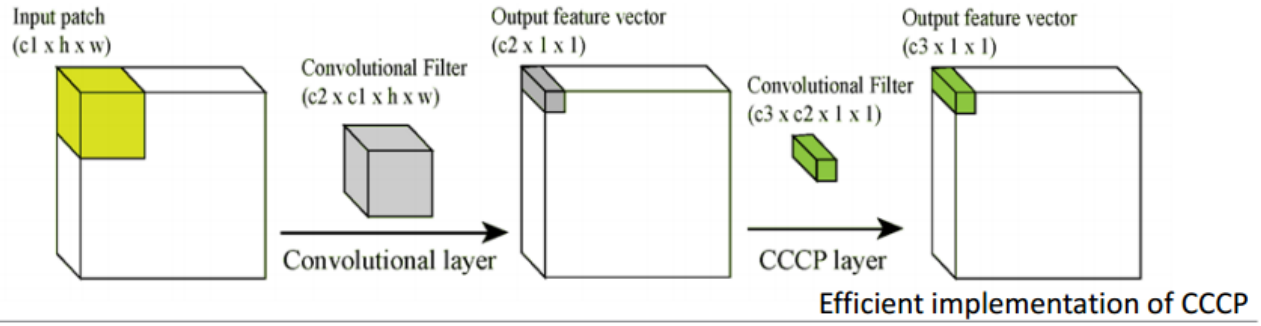
Maxoutはここで最大値をとるだけ（固定的）

Network in network (NIN)

Min Lin et al., “Network In Network”, arXiv preprint, 2013.

- 実装上、1x1の畳み込み層を重ねることと等価
  （本来の意味で畳み込みではないが）
- Deep learning的にはこちらの解釈の方が自然か

Better Local Abstraction ≈ Cascaded 1x1 Convolution



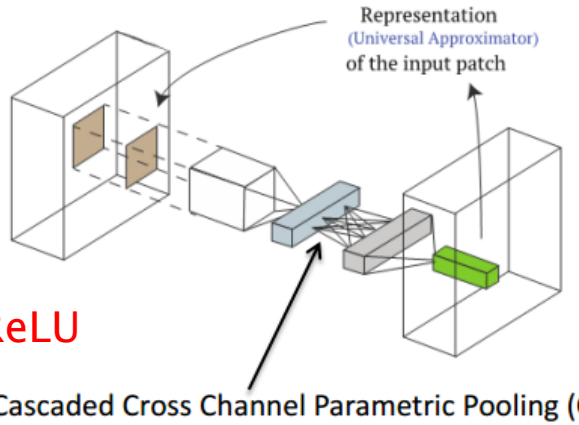
Efficient implementation of CCCP

Local patch is projected to its value in a feature map using **a small network**

$$y_i = \phi(w_i^T y_{i-1} + b_i)$$

$$y_0 = x$$

活性化関数はいずれもReLU



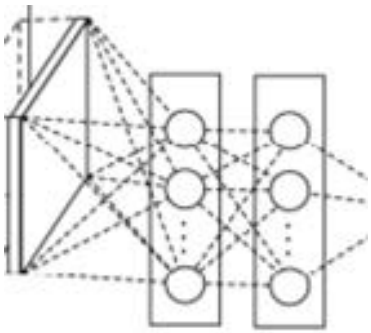
http://www.image-net.org/challenges/LSVRC/2014/slides/ILSVRC2014_NUS_release.pdf

高次統計量に基づくBoVWとの関係

- Fisher vector、VLAD等は受容野内の特徴の低次多項式表現を入力とする活性化関数を設計していると解釈できる
  - これは、普通のConvNetでは得られない構造

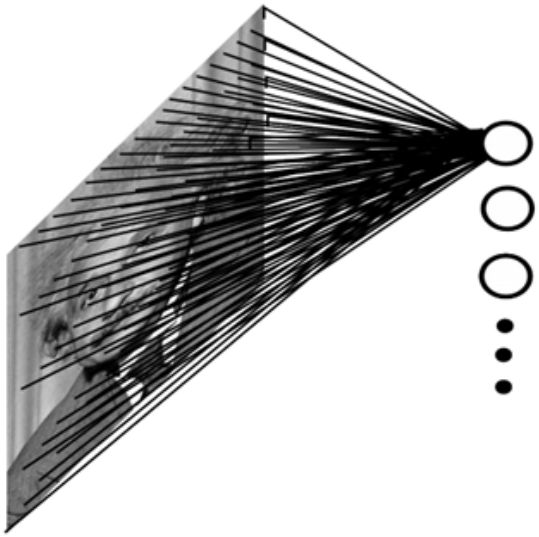
- 比較的小規模な多層ネットワークにより、多項式表現はモデル化可能
  - Andoni et al., “Learning Polynomials with Neural Networks”, ICML’14.

- つまり、NINでは活性化関数自体を多層ネットワークに分解することにより、Fisher vectorやVLADと同じ（あるいはさらに高次の）構造を、識別の点でより効率よく学習できていると期待できる
  - 結局、どこまでを活性化関数と考えるかの問題



ConvNet以外のアーキテクチャはどうか？

- 全結合ネットワーク
  - 極めて多くのパラメータ
  - 最適化が困難
    - 収束まで時間がかかる
    - そもそもメモリにのらない

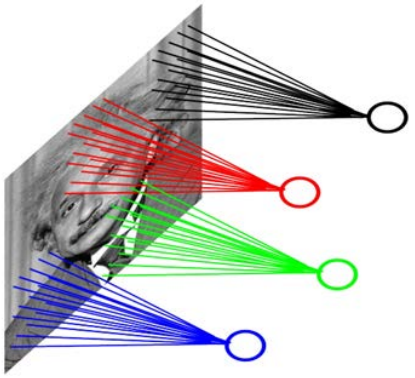


Source: M. Ranzato, CVPR'14 tutorial slides

MNISTデータセット（28x28ピクセル）のような小さい画像を用いて古くから研究されているが、今のところConvNetには遠く及ばない

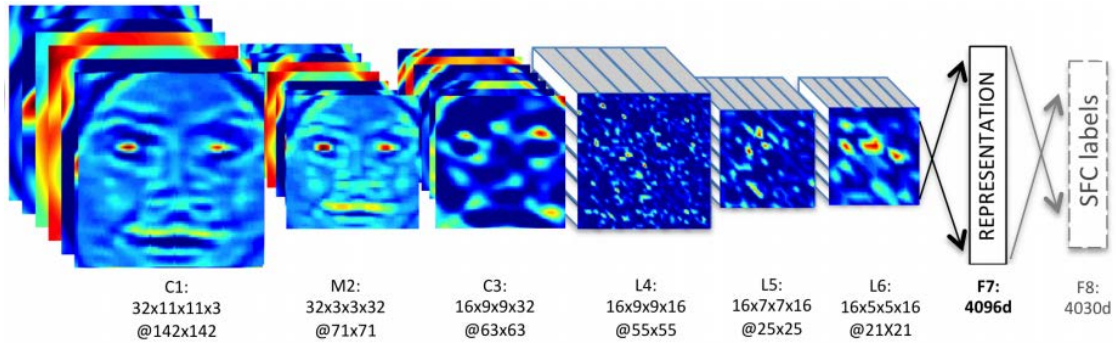
ConvNet以外のアーキテクチャはどうか？

- 局所結合ネットワーク
  - 構造はConvNetと同じだが、フィルタのパラメータに場所ごとで異なる
  - つまり、平行移動不変性がない



Source: M. Ranzato, CVPR'14 tutorial slides

- 入力画像の正確なアラインメントが前提となっている場合、state-of-the-art を達成している場合もある
  - DeepFace [Taigman et al., CVPR'14]

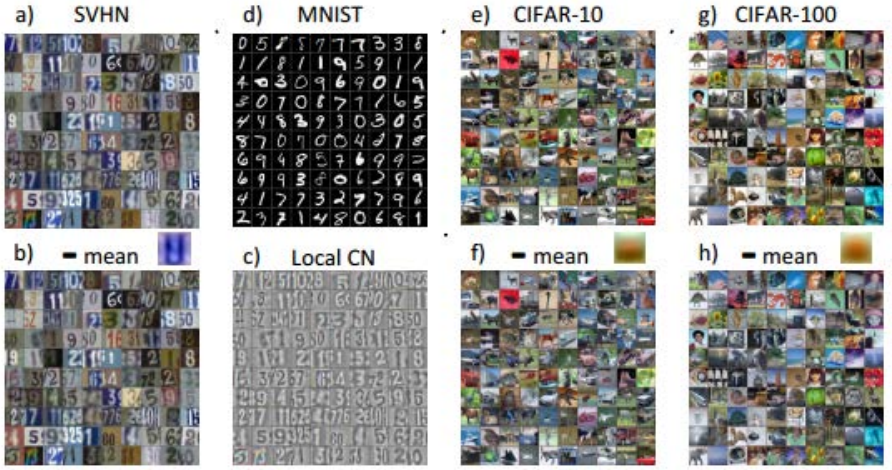


- 一般的な画像認識ではまだConvNetに劣る

その他、画像認識特有のノウハウ

- データの前処理（実はかなり重要）
  - ZCA whitening （白色化）
    コントラスト正規化など
  - 最終的な識別性能に大きく影響する

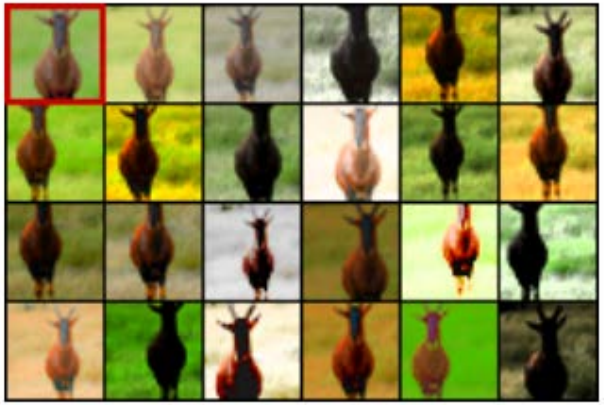
深層学習のまだ美しくないところその1



[Zeiler and Fergus, 2013]

- Data augmentation
  - アフィン変換、クロップなど、人工的に
    さまざまな変換を学習データに加える
  - 不変性を学習させる

深層学習のまだ美しくないところその2



[Dosovitskiy et al., 2014]

ここまでまとめ

- 局所的な畳み込み＋poolingという基本構造は今までの画像認識(BoVW)と変わらない。
  - 正確には、BoVW系が2000年代に一旦追いつき追い越し、再び逆転されたと見るべきか
  - 多層化、活性化関数の工夫、結合パラメータの全層最適化

	深さ	活性化関数	学習
BoVW	1層(デスクリプタは除く)	複雑	識別器の層以外(多くは)生成的識別層を独立に構築
ConvNet	多層	シンプル(ReLU)	識別的

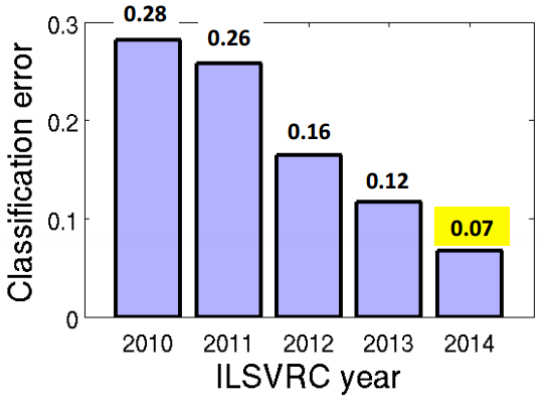
- より一般的な全結合・局所結合ネットワークなどはいまひとつ
  - おそらく構造に不変性がないのがネック
  - 今後の発展に期待（データがもっと増えればよくなる？）

本日の内容

- １．画像認識分野におけるdeep learningの歴史
- ２．一般画像認識：Deep learning 以前と以後で何が変わったか
  - Bag-of-visual-words（VLAD, Fisher Vector)
  - Convolutional neural network (ConvNets)
- ３．最新の動向・今後の展望
  - ILSVRC 2014
  - さらに高度な知能へ
- ４．実践するにあたって
  - 適切に利用するために必要な知識
  - 汎用ソフトウェア：Caffe

ImageNet Challenge (ILSVRC)

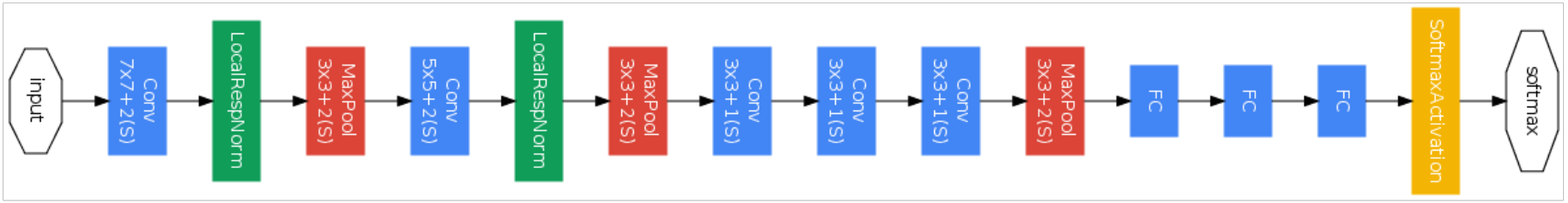
- 2012年のブレークスルー以降も、毎年識別性能が倍に…
  - まだ頭打ちの気配が見えない
  - ネットワークを深く、大きくすればするほど性能向上



http://www.image-net.org/challenges/LSVRC/2014/slides/ILSVRC2014_09_12_14_det.pdf

- 2014年
  - 優勝チーム(Google)は
    - 1000クラス識別タスクでの誤識別率が6.8%
    - 人間2人に同じタスクを試させたところ、それぞれ約5.1%、12.0%であった
  - 成績が良かったチーム
    - Google, Oxford, NUS
  - Network-in-networkで、とにかく深く大きくしたところが勝った
  - 多数のモデルのアンサンブル
  - 教師なし事前学習はほとんど使われていない

2013



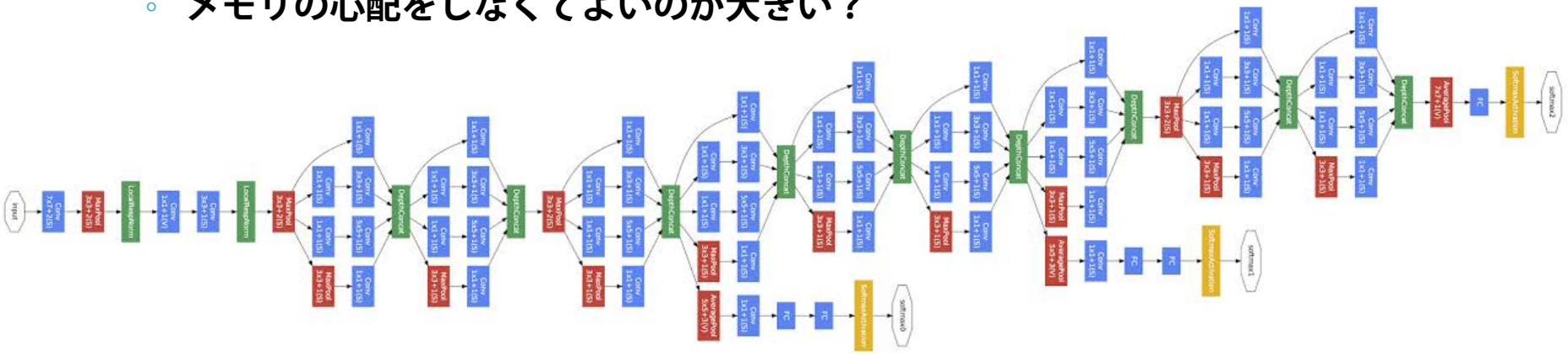
Zeiler-Fergus Architecture
(AlexNetとほぼ同じ)

Convolution
Pooling
Softmax
Other

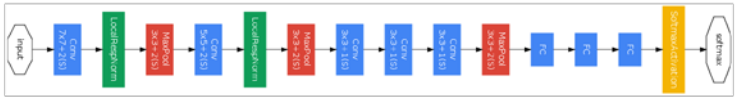
http://www.image-net.org/challenges/LSVRC/2014/slides/GoogLeNet.pptx

2014

- DistBeliefと呼ばれるGoogle独自の並列分散フレームワークで学習
  - CPUベース
  - メモリの心配をしなくてよいのが大きい？



GoogLeNet (22層)



Zeiler-Fergus Architecture
(AlexNetとほぼ同じ)

Convolution
Pooling
Softmax
Other

http://www.image-net.org/challenges/LSVRC/2014/slides/GoogLeNet.pptx

Oxford Model

5 days

3 weeks

Arch.	conv1	conv2	conv3	conv4	conv5	full1	full2	full3
CNN-F	64x11x11 st. 4, pad 0 LRN, x2 pool	256x5x5 st. 1, pad 2 LRN, x2 pool	256x3x3 st. 1, pad 1 -	256x3x3 st. 1, pad 1 -	256x3x3 st. 1, pad 1 x2 pool	4096 drop-out	4096 drop-out	1000 soft-max
CNN-M	96x7x7 st. 2, pad 0 LRN, x2 pool	256x5x5 st. 2, pad 1 LRN, x2 pool	512x3x3 st. 1, pad 1 -	512x3x3 st. 1, pad 1 -	512x3x3 st. 1, pad 1 x2 pool	4096 drop-out	4096 drop-out	1000 soft-max
CNN-S	96x7x7 st. 2, pad 0 LRN, x3 pool	256x5x5 st. 1, pad 1 x2 pool	512x3x3 st. 1, pad 1 -	512x3x3 st. 1, pad 1 -	512x3x3 st. 1, pad 1 x3 pool	4096 drop-out	4096 drop-out	1000 soft-max

Chatfield et al., "Return of the Devil in the Details: Delving Deep into Convolutional Nets", 2014.

	ILSVRC-2012 (top-5 error)
(a) FK IN 512	-
(b) CNN F	16.7
(c) CNN M	13.7
(d) CNN M 2048	13.5
(e) CNN S	**13.1**
(f) CNN S TUNE-CLS	**13.1**
(g) CNN S TUNE-RNK	**13.1**
(h) Zeiler & Fergus [16]	16.1
(i) Razavian *et al.* [9], [10]	14.7
(j) Oquab *et al.* [8]	18

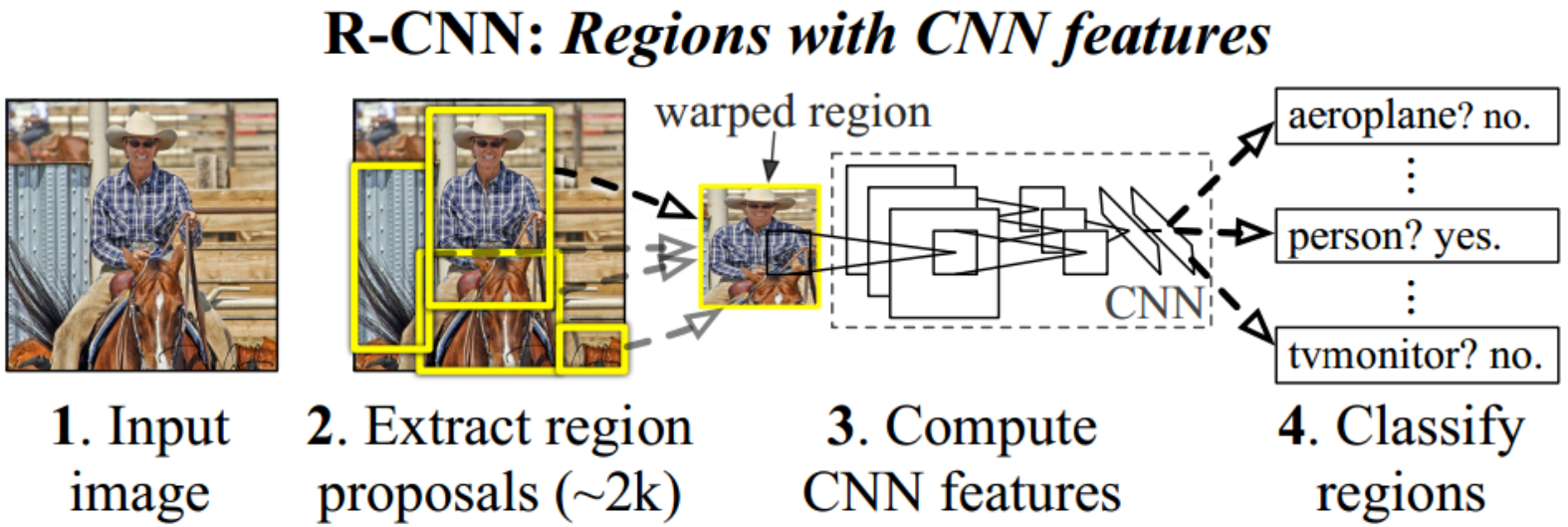
GoogLeNet

type	patch size/ stride	output size	depth	#1×1	#3×3 reduce	#3×3	#5×5 reduce	#5×5	pool proj	params	ops
convolution	7×7/2	112×112×64	1							2.7K	34M
max pool	3×3/2	56×56×64	0								
convolution	3×3/1	56×56×192	2		64	192				112K	360M
max pool	3×3/2	28×28×192	0								
inception (3a)		28×28×256	2	64	96	128	16	32	32	159K	128M
inception (3b)		28×28×480	2	128	128	192	32	96	64	380K	304M
max pool	3×3/2	14×14×480	0								
inception (4a)		14×14×512	2	192	96	208	16	48	64	364K	73M
inception (4b)		14×14×512	2	160	112	224	24	64	64	437K	88M
inception (4c)		14×14×512	2	128	128	256	24	64	64	463K	100M
inception (4d)		14×14×528	2	112	144	288	32	64	64	580K	119M
inception (4e)		14×14×832	2	256	160	320	32	128	128	840K	170M
max pool	3×3/2	7×7×832	0								
inception (5a)		7×7×832	2	256	160	320	32	128	128	1072K	54M
inception (5b)		7×7×1024	2	384	192	384	48	128	128	1388K	71M
avg pool	7×7/1	1×1×1024	0								
dropout (40%)		1×1×1024	0								
linear		1×1×1000	1							1000K	1M
softmax		1×1×1000	0								

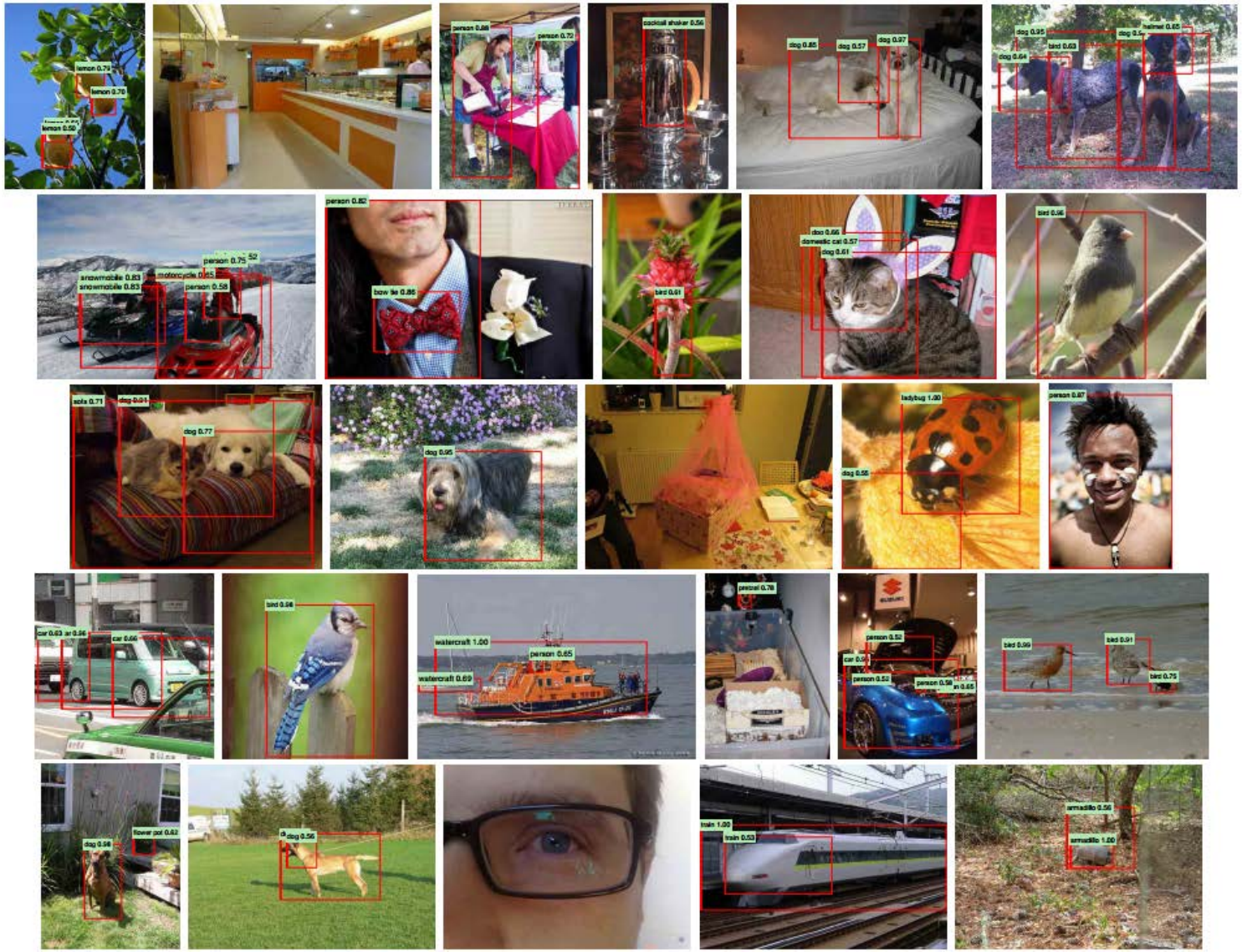
Szegedy et al., “Going deeper with convolutions”, 2014.

ConvNet for object detection

- **R-CNN** [Girshick et al., CVPR'2014]
  - 物体の領域候補を多数抽出（これ自体は別手法）
  - 無理やり領域を正規化し、CNNで特徴抽出
  - SVMで各領域を識別



R-CNNもCaffeと同じチームが開発・提供
（比較的簡単に試せます）



ランダムに選んだテスト画像の認識結果
（いいところだけ見せているのではない！）

Girshick et al., “Rich feature hierarchies for accurate object detection and semantic segmentation”, In arXiv, 2014.

2014 GoogLeNet detection results

- 基本構造はR-CNNと同じで、
  CNN部分をGoogLeNetに置き換え

- 検出率（mAP、200クラス)
  - ILSVRC 2013 winner： 22.6%
  - R-CNN： 31.4%
  - GoogLeNet：43.9%

  - Googleチームの続報(12月)：55.7%

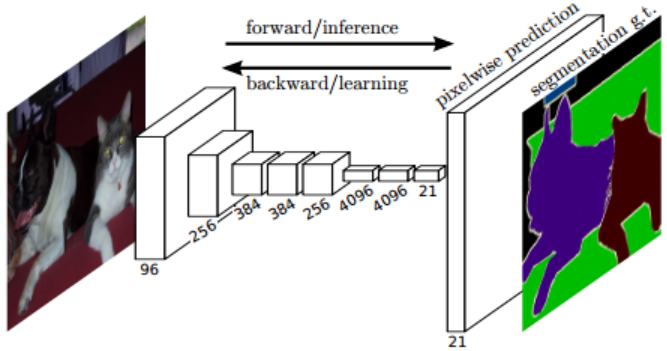
    Szegedy et al., “Scalable, High-Quality Object Detection”, In arXiv, 2014.

最近の話題

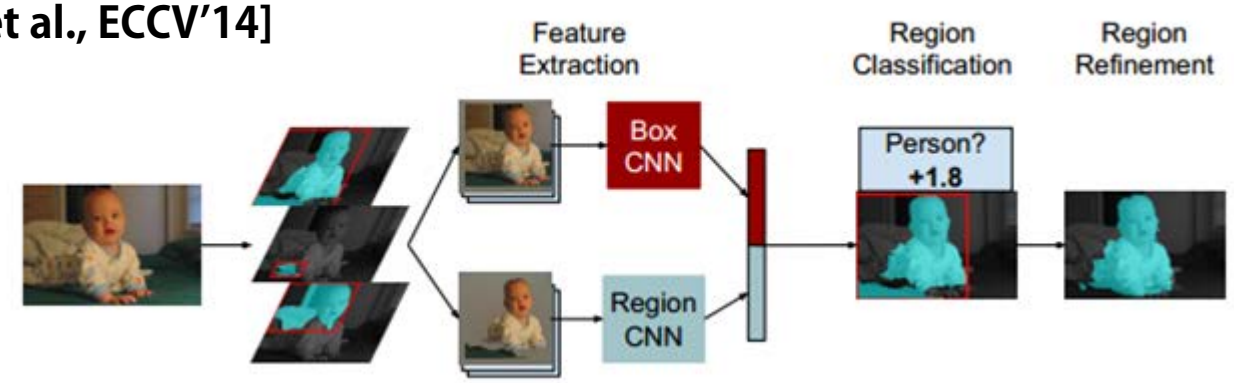
- より難しい認識タスクへ
  - セマンティック・セグメンテーション
  - 画像・動画像の文章による説明
- マルチモーダル学習
- 計画・行動
  - 強化学習とのコラボレーション

より難しい認識タスクへ

- **Semantic segmentation**
  - ピクセルレベルで物体領域を認識
  - [Long et al., 2014]



- **Segmentation + Detection (同時最適化)**
  - [Hariharan et al., ECCV'14]



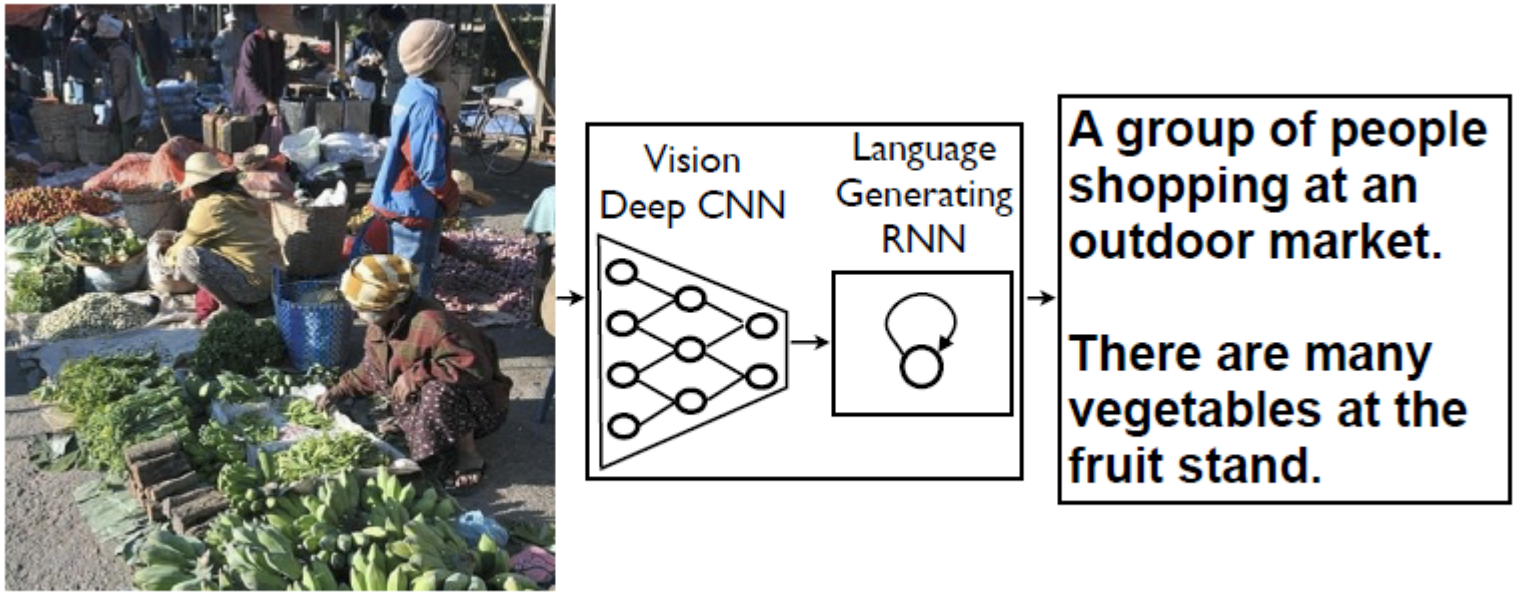
画像の説明文生成

- 2014年11月、同時多発的にいろんなグループが発表
  - arXivで公開。おそらくCVPR 2015に投稿したもの。
  - Recurrent Neural Network (RNN) が言語モデルとして大人気

- Google
  - O. Vinyals et al., “Show and Tell: A Neural Image Caption Generator”, 2014.
- Microsoft
  - H. Fang et al., “From Captions to Visual Concepts and Back”, 2014.
- Stanford
  - A. Karpathy and L. Fei-Fei, “Deep Visual-Semantic Alignments for Generating Image Descriptions”, 2014.
- UC Berkeley
  - J. Donahue et al., “Long-term Recurrent Convolutional Networks for Visual Recognition and Description”, 2014.
- Univ. Toronto
  - R. Kiros et al., “Unifying Visual-Semantic Embeddings with Multimodal Neural Language Models”, 2014

Google のシステム

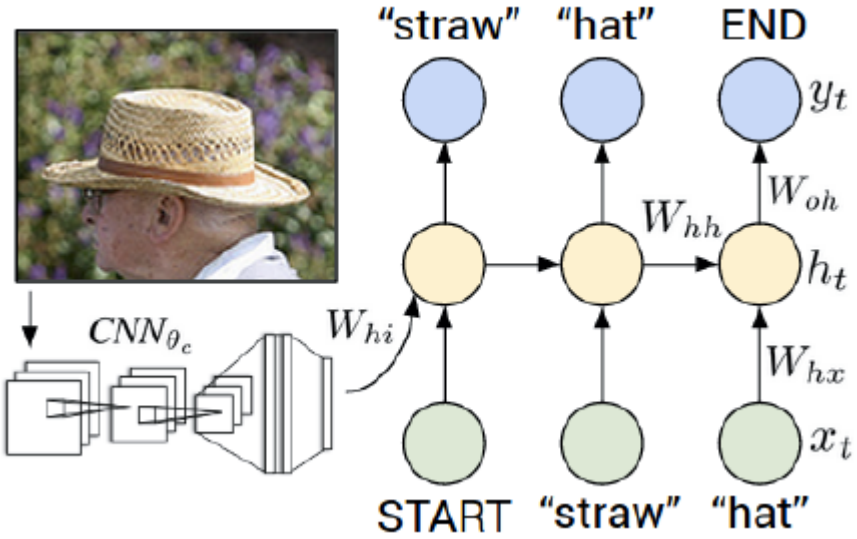
- ConvNet (画像側)の出力をRNN(言語側)へ接続
  - RNN側の誤差をConvNet側までフィードバック



O. Vinyals et al., “Show and Tell: A Neural Image Caption Generator”, 2014

Stanford のシステム

- 領域ベース (RCNNを利用)



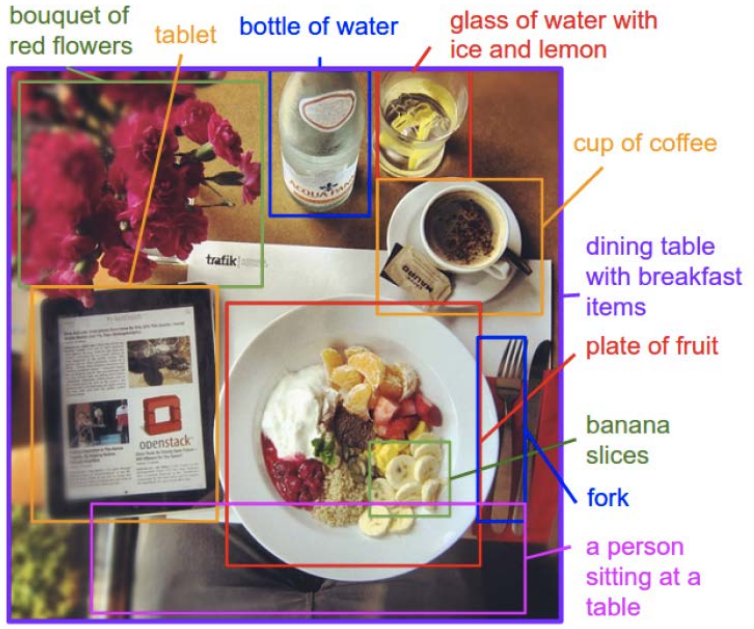


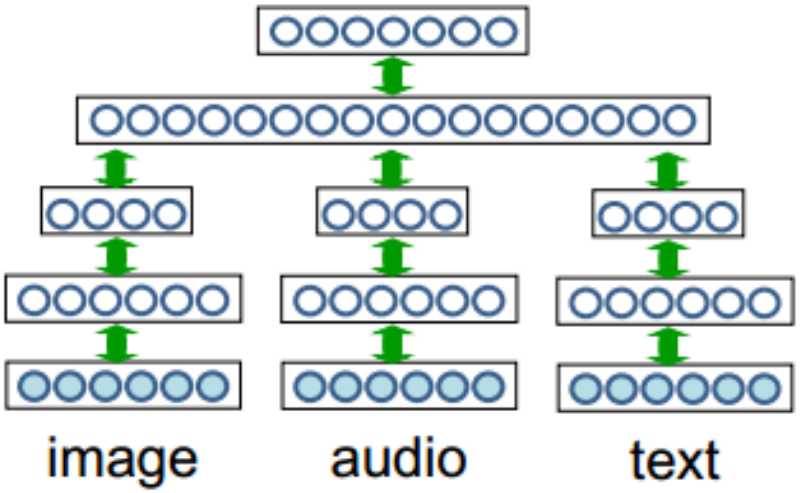
Figure 1. Our model generates free-form natural language descriptions of image regions.

A. Karpathy and L. Fei-Fei, "Deep Visual-Semantic Alignments for Generating Image Descriptions", 2014.

マルチモーダル学習

- 複数のモダリティを一つの枠組で統合
  - よりロバスト・汎用的な知能へ

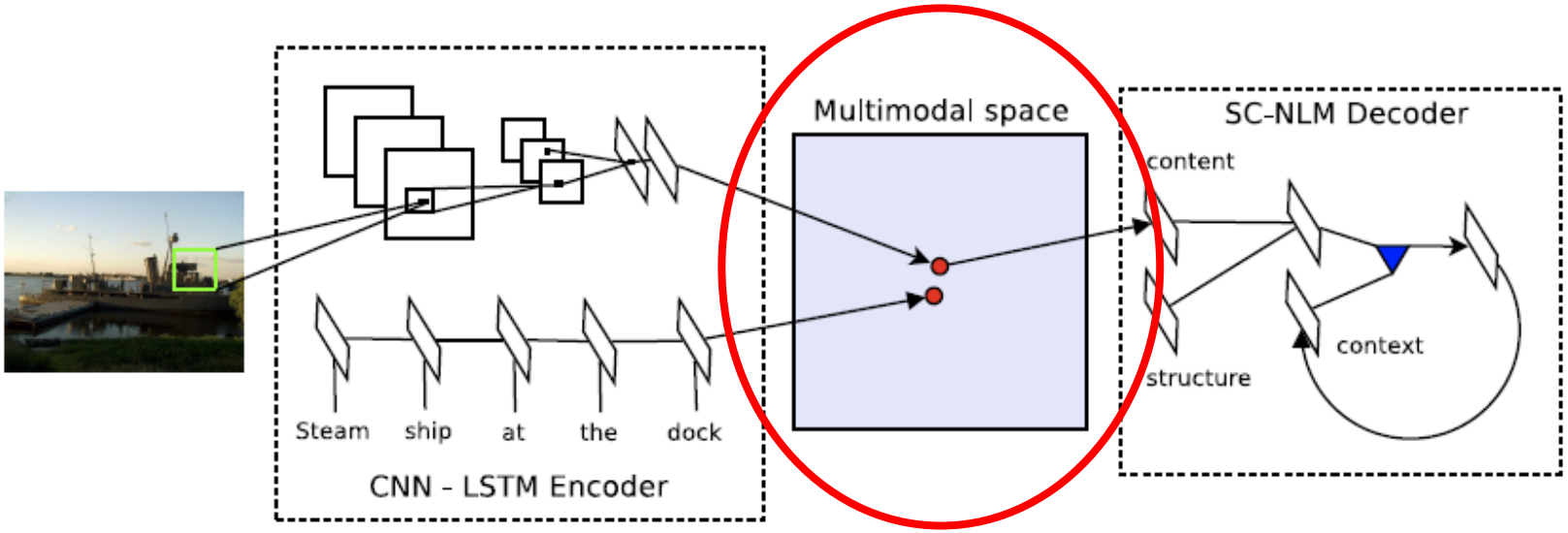
- Images
- Audio & speech
- Video
- Text
- Robotic sensors
- Time-series data
- Others

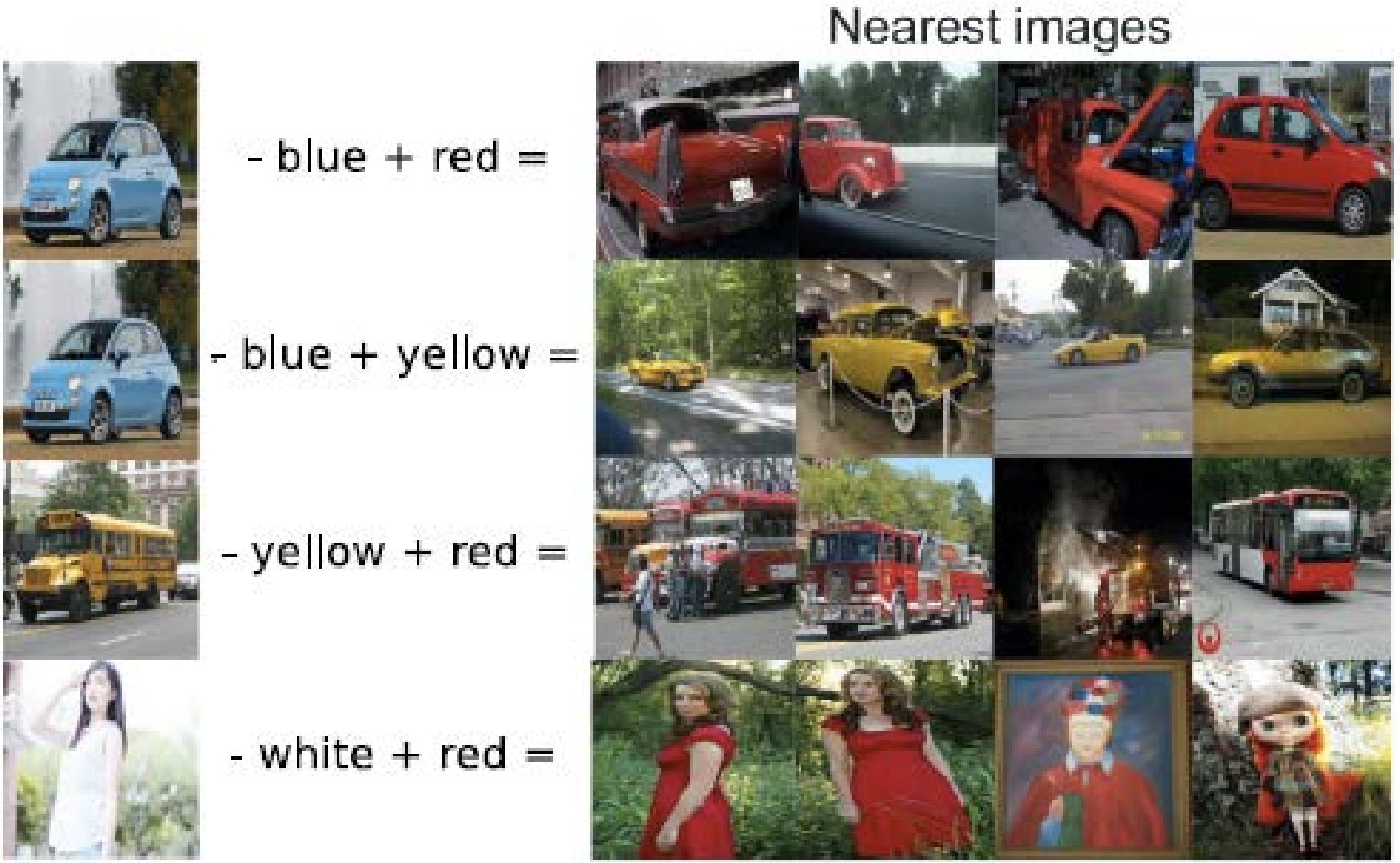


（CVPR'12 チュートリアルスライドより引用）

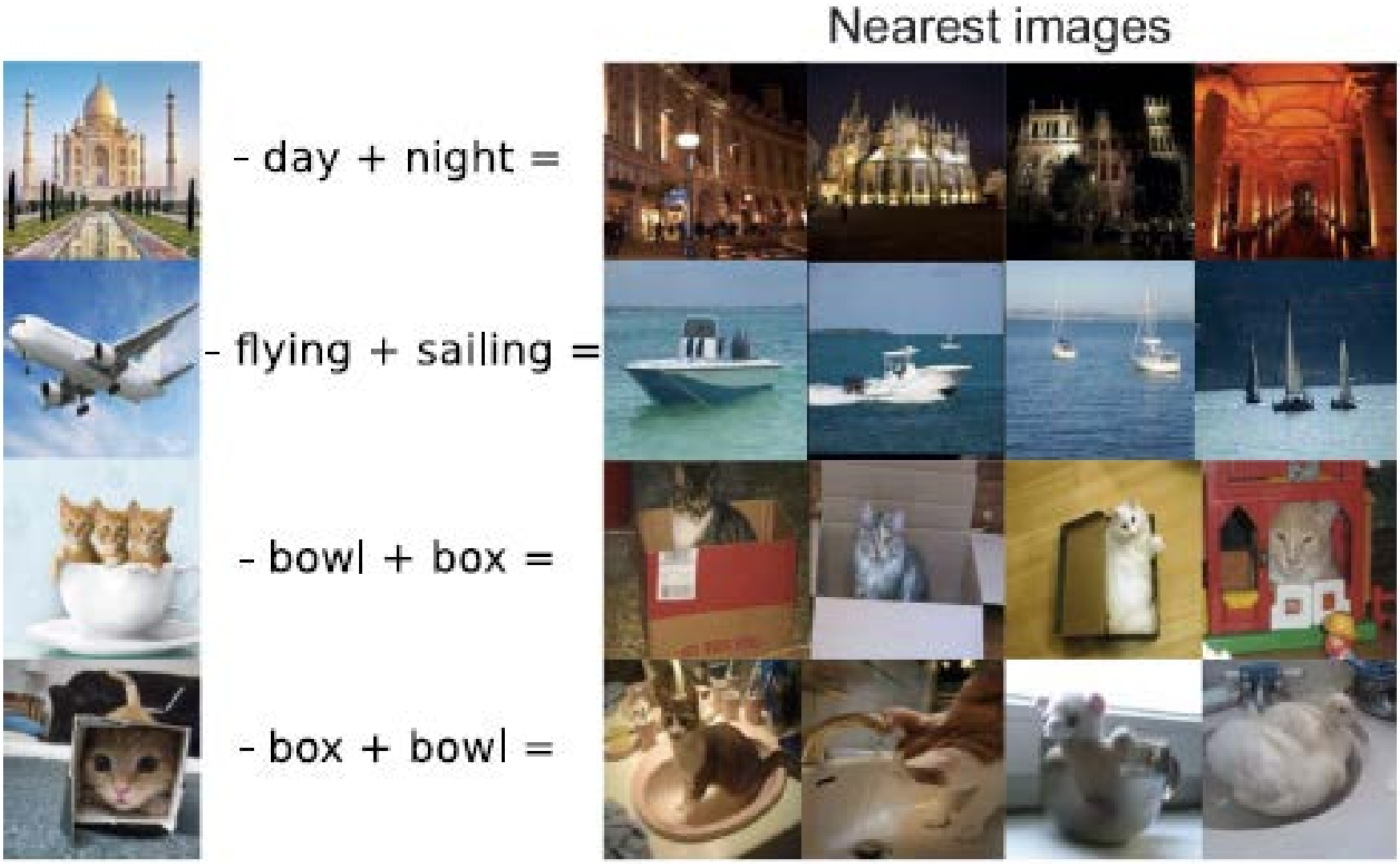
画像＋テキスト

- 共通の上位レイヤ(潜在空間)へマッピング [Kiros et al., 2014]
  - 異なるモダリティ間での“演算”が可能





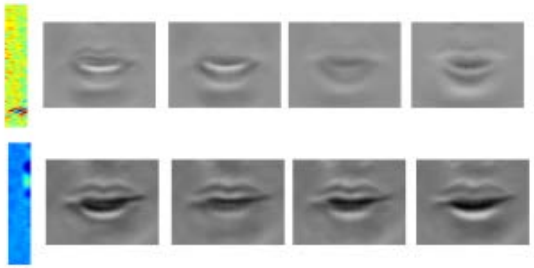
[Kiros et al., 2014]

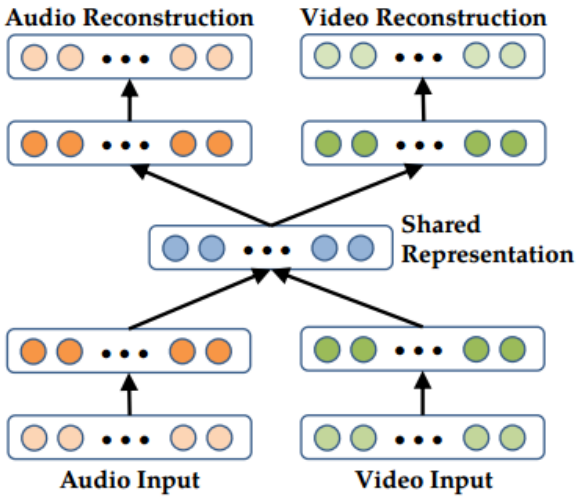


[Kiros et al., 2014]

画像＋音声

- **Bimodal Deep Autoencoder** [Ngiam et al., ICML'11]
  - 音声＋画像(唇)による発話音認識





  - 音声側にノイズが大きい時にもロバスト

Feature Representation	Accuracy (Clean Audio)	Accuracy (Noisy Audio)
(a) Audio RBM (Figure 2a)	**95.8%**	75.8% ± 2.0%
(b) Video-only Deep Autoencoder (Figure 3a)	68.7%	68.7%
(c) Bimodal Deep Autoencoder (Figure 3b)	90.0%	77.3% ± 1.4%
(d) **Bimodal + Audio RBM**	94.4%	**82.2%** ± 1.2%
(e) Video-only Deep AE + Audio-RBM	87.0%	76.6% ± 0.8%

計画・行動へ

- **Deep Q-learning** [Mnih et al, NIPS'13]
  - DeepMind（Googleに買収されたベンチャー）の発表
  - 強化学習の報酬系の入力に畳み込みネットワークを利用（生画像を入力）
  - アタリのクラッシックゲームで人間を超える腕前

今後の展望

- ConvNetの深層化、巨大化による性能向上はまだまだ続きそう
  - 一般的には、GPUのビデオメモリがボトルネック
  - データが少ない領域（映像、３次元物体認識等）では発展途上
    （最近は急速にデータが増えつつあるが）

- しかしながら、依然としてConvNetの構造に依存している
  - 全結合ネットワークなどは今後成功するか？
  - 真の意味でブラックボックスになるか？

- より汎用的な人工知能へ近づくことはできるか？
  - 深い意味構造の理解、記憶、思考、計画、創造…

本日の内容

- １．画像認識分野におけるdeep learningの歴史
- ２．一般画像認識：Deep learning 以前と以後で何が変わったか
  - Bag-of-visual-words（VLAD, Fisher Vector)
  - Convolutional neural network (ConvNets)
- ３．最新の動向・今後の展望
  - ILSVRC 2014
  - さらに高度な知能へ
- ４．実践するにあたって
  - 適切に利用するために必要な知識
  - 汎用ソフトウェア：Caffe

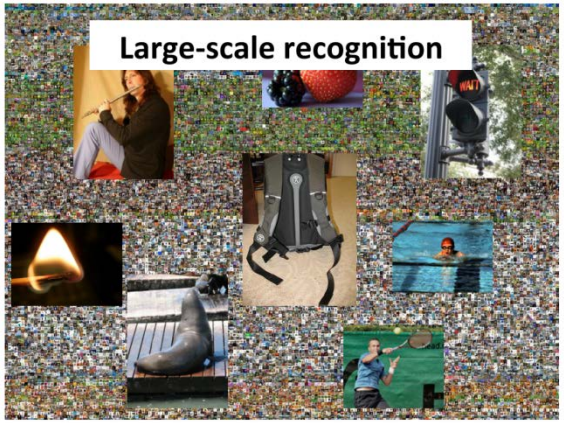
最初に考えるべきこと

- 自分の問題について、どのようにdeep learningを使うべきか？
  - 十分な効果を得るには、かなり多くの教師付データが必要
  - 必ずしもフルスクラッチから学習することが賢いとは限らない
- そもそもdeep learningを利用できる問題か？

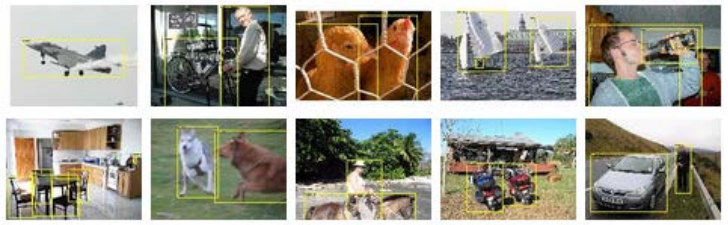
外部大規模データを用いたpre-training

- あらかじめ**汎用性の高い大規模教師付データセット**でネットワークを学習しておき、これを初期値としてターゲットタスクの学習データでさらに細かい学習を進める（＝**Fine-tuning**）
(Unsupervised pre-trainingとは違う概念であることに注意)

- 例えば…



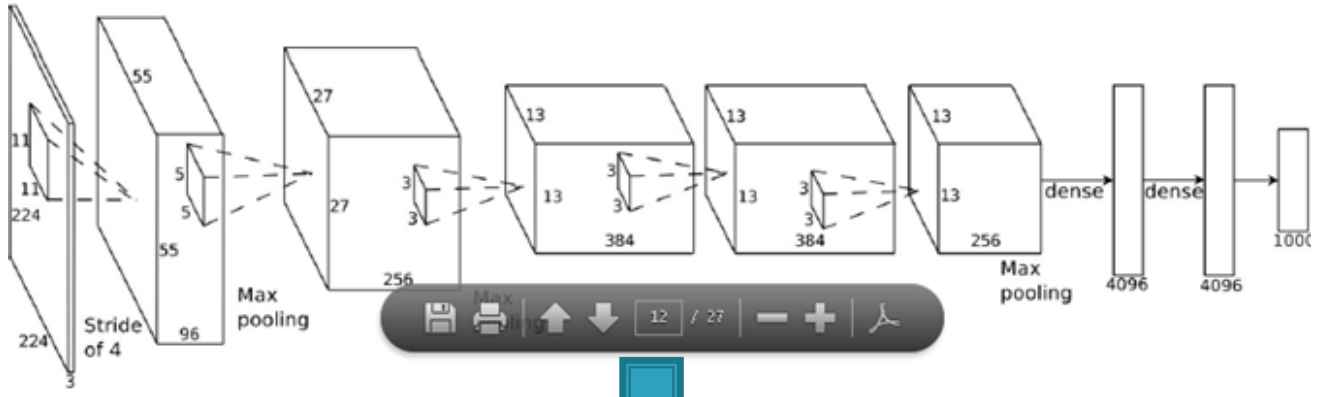
ImageNet ILSVRC'12
130万枚、1000クラス



PASCAL VOC 2007
5千枚、20クラス

Fine-tuning

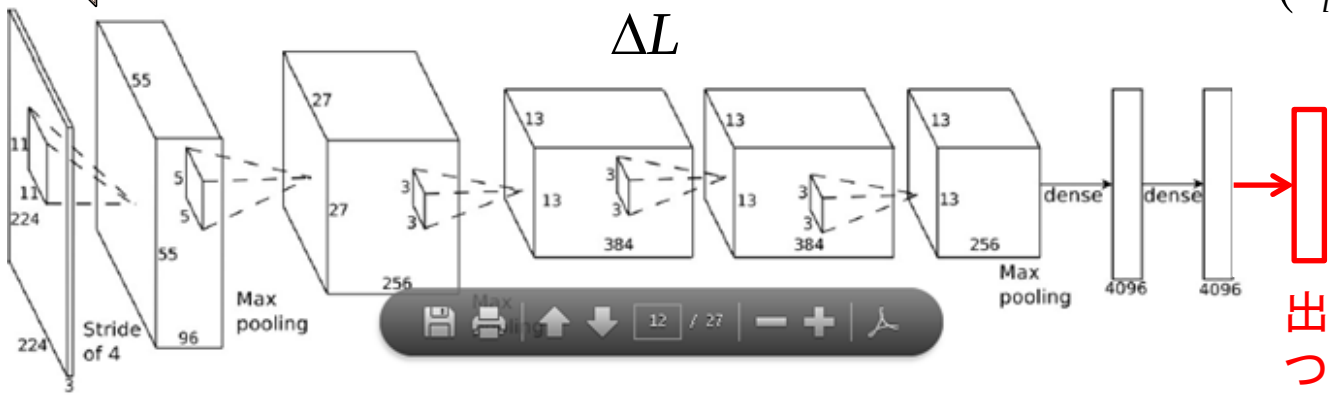
Pre-trained network



適用先データセット

$L(\mathbf{x}_i, \mathbf{y}_i)$

ΔL

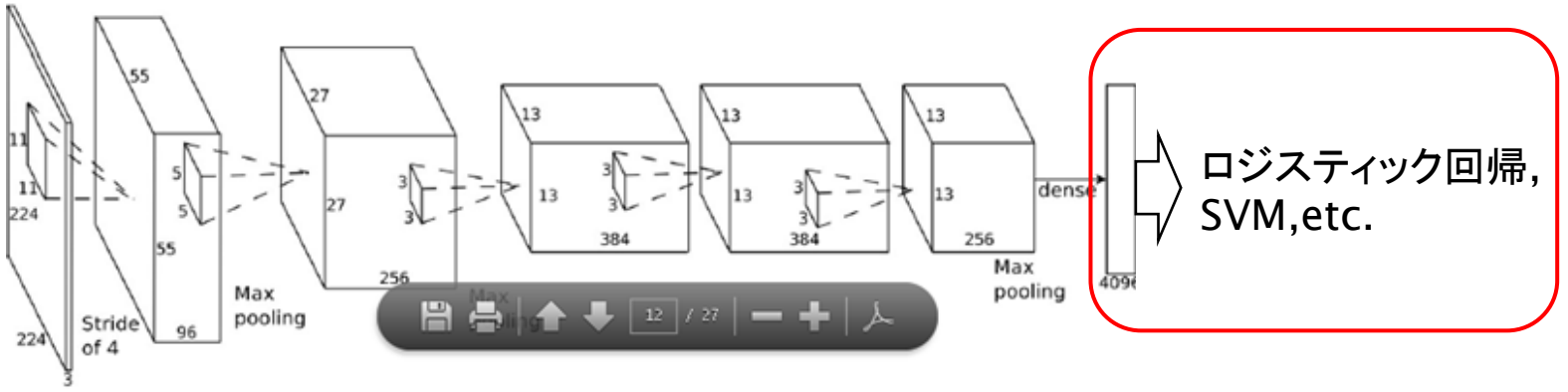


出力層だけつけかえ

Using pre-trained features

- Pre-trainedネットワークを特徴抽出器として用いる
  - 中間層の出力を利用して識別器を構築

Pre-trained network

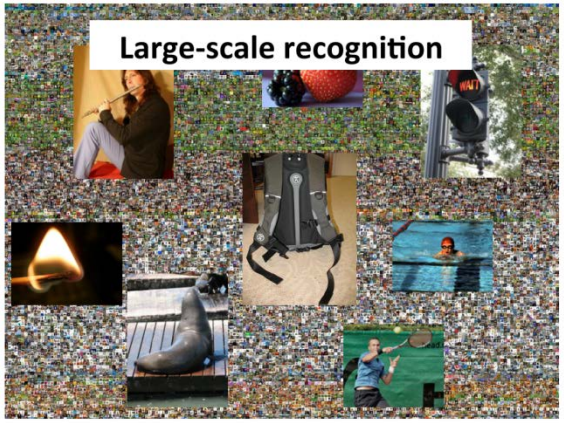


- 最終層だけfine-tuningしているとも解釈できる

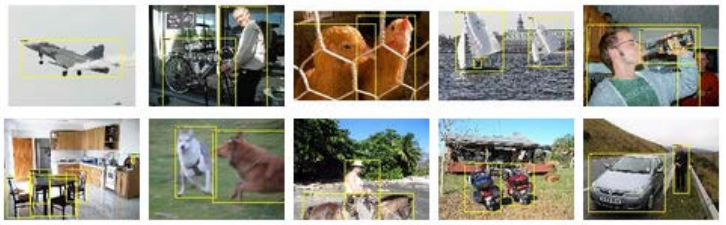
Pre-trainingの効果

- ILSVRC 2012 → VOC 2007 の例 (検出成功率、mAP%)
  - フルスクラッチConvNet:　40.7
  - Pre-trained feature: 45.5
  - Fine tuning: 54.1

Agrawal et al., “Analyzing the Performance of Multilayer Neural Networks for Object Recognition”, In Proc. ECCV, 2014.



ImageNet ILSVRC’12
130万枚、1000クラス



PASCAL VOC 2007
5千枚、20クラス

注意：何でもうまくいくわけではない

- Pre-trainingに用いる外部データセットが、所望のタスクを内包するものでなければ効果が薄い（むしろ悪化）
  - ImageNetはあくまで物体認識のデータセット

- 参考：Fine-grained competition 2013

https://sites.google.com/site/fgcomp2013/





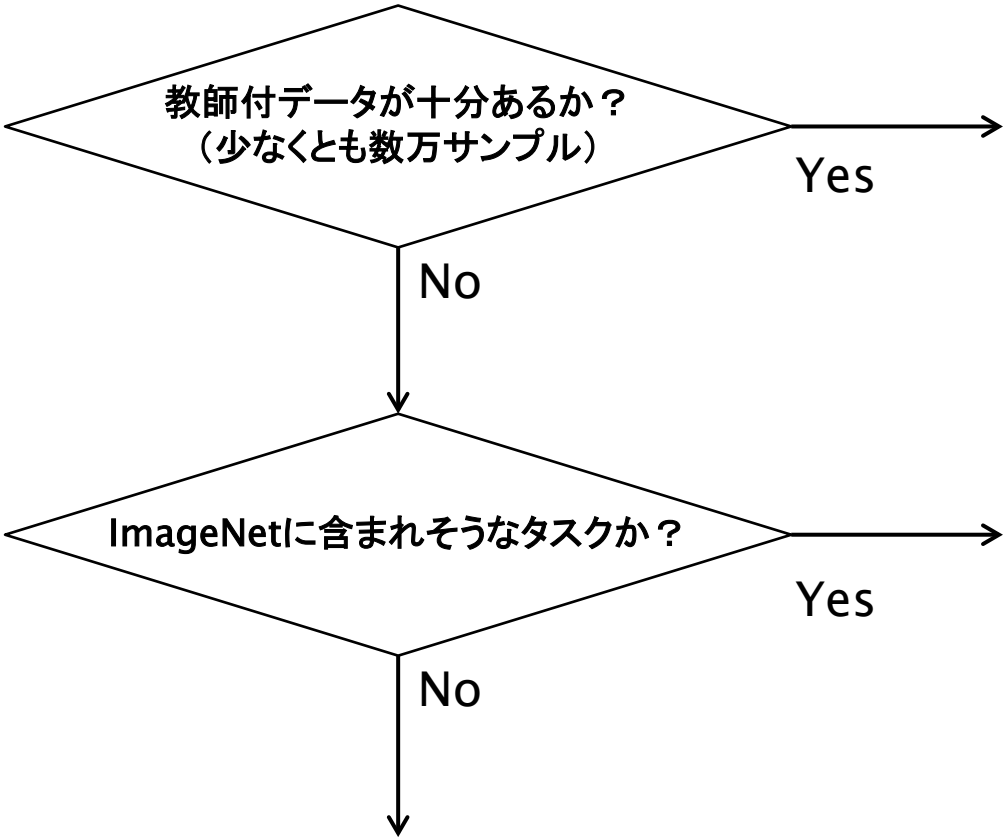
Team	Aircraft	Birds	Cars	Dogs	Shoes	Overall
Inria-Xerox	**81.46**	71.69	**87.79**	52.90	**91.52**	**77.07**
CafeNet	78.85	**73.01**	79.58	57.53	90.12	75.82

Fisher vector

ConvNet (fine-tuning)

飛行機、車、靴データセットなど、ImageNet上にあまりデータが存在しないドメインに関してはターゲットの学習データのみ用いたFisher vectorの方が良かった

手法選択の方針



- フルスクラッチCNN
  - ランダムに初期化

- Pre-trained feature (一番簡単)
- Fine tuning

- 教師なし事前学習
- Stackingによる深層モデル
- あきらめてshallow learning

計算環境の準備

- ハードウェア
  - 学習にはGPU計算機が必要 （CUDAを利用）
  - ビデオメモリの容量がボトルネックになる場合が多い
    - メインメモリとの通信は遅い
    - ネットワークのパラメータはもちろん、できるだけ多くの学習サンプルをビデオメモリに積みたい

- Titan Black （約15万円）
  - コストパフォーマンス的にお薦め
  - 当研究室では、これを積んだPCが6台ほど

- Tesla K20 (約40万円), K40 (約80万円)
  - より信頼性が高い

オープンソースソフトウェア

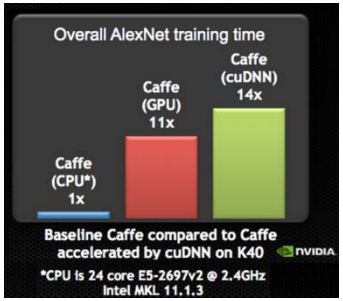
- 2012年頃から、著名な研究チームによる主導権争い
  - Caffe/Decaf： UC Berkeley
  - Theano/Pylearn2： Univ. Montreal
  - Torch7： Univ. New York
  - Cuda-convnet2: Univ. Toronto (Alex Krizhevsky)

Framework	License	Core language	Binding(s)	CPU	GPU	Open source	Training	Pretrained models	Development
Caffe	BSD	C++	Python, MATLAB	✓	✓	✓	✓	✓	distributed
cuda-convnet [7]	unspecified	C++	Python		✓	✓	✓		discontinued
Decaf [2]	BSD	Python		✓		✓	✓	✓	discontinued
OverFeat [9]	unspecified	Lua	C++,Python	✓				✓	centralized
Theano/Pylearn2 [4]	BSD	Python		✓	✓	✓	✓		distributed
Torch7 [1]	BSD	Lua		✓	✓	✓	✓		distributed

Y. Jia et al., “Caffe: Convolutional Architecture for Fast Feature Embedding”, ACM Multimedia Open Source Competition, 2014.

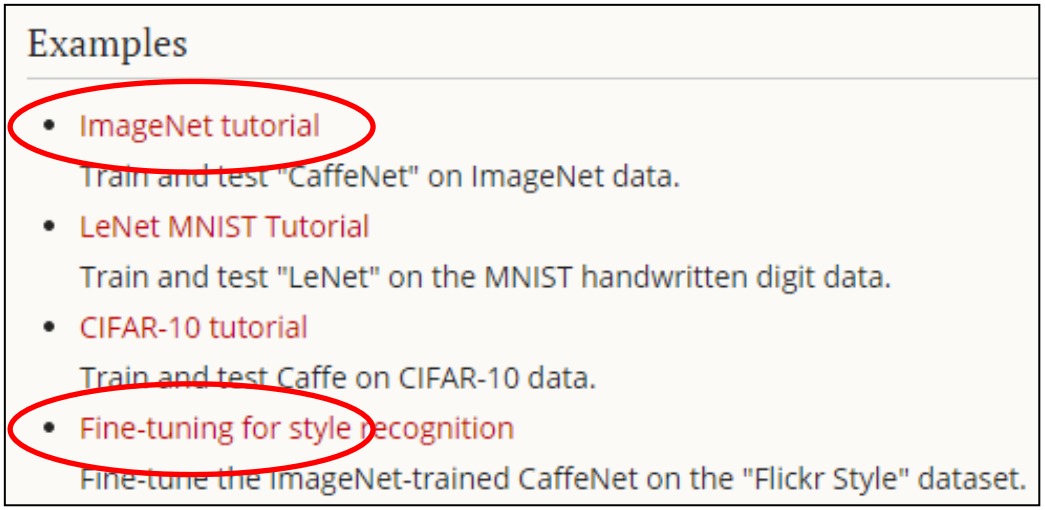
Caffe (BSD 2-Clause license)

- 頭一つ抜けた印象（個人的な感想ですが）
  - トップクラスに高速
  - オープンソースコミュニティとして確立しつつある
- 多くの研究者が既に自分の研究に利用
  - Oxford visual geometry group など
- Model Zoo
  - 各研究者の学習済みネットワークを共有
  - AlexNetはもちろん、Network-in-network、GoogLeNet モデルなども
  - 最新の成果を極めて容易に試せる
- NVIDIAの手厚いサポート
  - cuDNNのいち早い実装



Caffe

- Webドキュメントが充実　http://caffe.berkeleyvision.org/
  - ImageNet等の結果を再現可能
  - IPython notebookによる
    コード実例多数



- ECCV 2014でのチュートリアル
  - http://tutorial.caffe.berkeleyvision.org/

Brewing by the Numbers...

- Speed with Krizhevsky's 2012 model:
  - K40 / Titan: 2 ms / image, K20: 2.6ms
  - 40 million images / day
  - Caffe + cuDNN: **1.17ms / image** on K40
  - 8-core CPU: ~20 ms/image
- ~ 9K lines of C/C++ code
  - with unit test: ~20k

* Not counting image I/O time. Details at http://caffe.berkeleyvision.org/performance_hardware.html

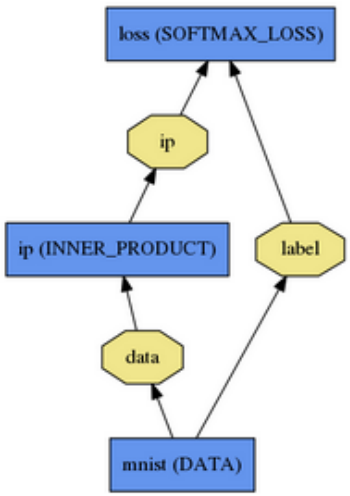
ECCV'14 チュートリアルスライド「DIY Deep Learning for Vision: a Hands-On Tutorial with Caffe」より引用

Net

- A network is a set of layers connected as a DAG:

```
name: "dummy-net"
layers { name: "data" …}
layers { name: "conv" …}
layers { name: "pool" …}
    … more layers …
layers { name: "loss" …}
```

- Caffe creates and checks the net from the definition.
- Data and derivatives flow through the net as *blobs* – a an array interface



LogReg ↑

LeNet →

ImageNet, Krizhevsky 2012 →

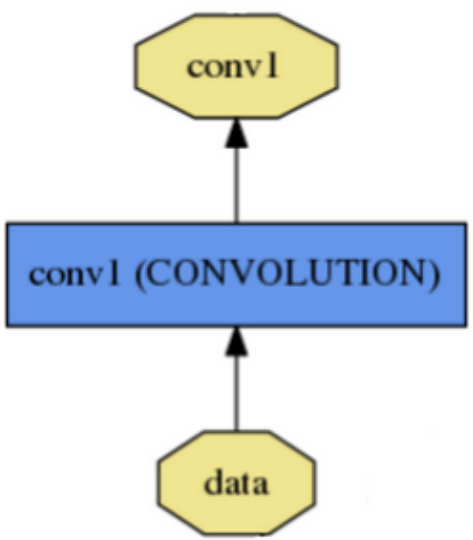
ECCV'14 チュートリアルスライド「DIY Deep Learning for Vision: a Hands-On Tutorial with Caffe」より引用

Layer

```
name: "conv1"
type: CONVOLUTION
bottom: "data"
top: "conv1"
convolution_param {
    num_output: 20
    kernel_size: 5
    stride: 1
    weight_filler {
        type: "xavier"
    }
}
```

name, type, and the connection structure (input blobs and output blobs)

layer-specific parameters



ECCV'14 チュートリアルスライド「DIY Deep Learning for Vision: a Hands-On Tutorial with Caffe」より引用

Solving: Training a Net

Optimization like model definition is configuration.

```
train_net: "lenet_train.prototxt"
base_lr: 0.01
momentum: 0.9
weight_decay: 0.0005
max_iter: 10000
snapshot_prefix: "lenet_snapshot"
solver_mode: GPU
```

All you need to run things on the GPU.

```
> caffe train -solver lenet_solver.prototxt
```

ECCV'14 チュートリアルスライド「DIY Deep Learning for Vision: a Hands-On Tutorial with Caffe」より引用

Fine-tuningも簡単

- Simply change a few lines in the layer definition

```
layers {
  name: "data"
  type: DATA
  data_param {
    source: "ilsvrc12_train_leveldb"
    mean_file: "../../data/ilsvrc12"
    ...
  }
  ...
}
...
layers {
  name: "fc8"
  type: INNER_PRODUCT
  blobs_lr: 1
  blobs_lr: 2
  weight_decay: 1
  weight_decay: 0
  inner_product_param {
    num_output: 1000
    ...
  }
}
```

```
layers {
  name: "data"
  type: DATA
  data_param {
    source: "style_leveldb"
    mean_file: "../../data/ilsvrc12"
    ...
  }
  ...
}
...
layers {
  name: "fc8-style"
  type: INNER_PRODUCT
  blobs_lr: 1
  blobs_lr: 2
  weight_decay: 1
  weight_decay: 0
  inner_product_param {
    num_output: 20
    ...
  }
}
```

new name = new params

Input:
A different source

Last Layer:
A different classifier

ECCV'14 チュートリアルスライド「DIY Deep Learning for Vision: a Hands-On Tutorial with Caffe」より引用