2014/06/18 上智大学 山中高夫

Deformable Part Modelとその発展

DPM [1, 2]

HSC [3]

R-CNN [4]

[1] P. Felzenszwalb, D. Mcallester, and D. Ramanan, “A Discriminatively Trained , Multiscale , Deformable Part Model,” in IEEE Conference on Computer Vision and Pattern Recognition, 2008.

[2] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, “Object detection with discriminatively trained part-based models,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 32, no. 9, pp. 1627–45, Sep. 2010.

[3] X. Ren and D. Ramanan, “Histograms of Sparse Codes for Object Detection,” in IEEE Conference on Computer Vision and Pattern Recognition, 2013.

[4] R. Girshick, J. Donahue, T. Darrell, and J. Malik, “Rich feature hierarchies for accurate object detection and semantic segmentation,” in IEEE Conference on Computer Vision and Pattern Recognition, 2014.

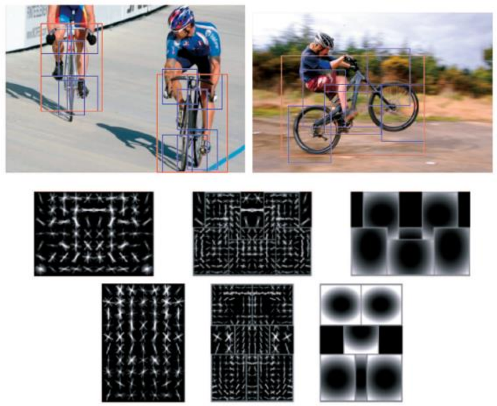


Fig. 2. Detections obtained with a two-component bicycle model. These examples illustrate the importance of deformations and mixture models. In this model, the first component captures sideways views of bicycles while the second component captures frontal and near frontal views. The sideways component can deform to match a “wheelie.”

Deformable Part Model (DPM)

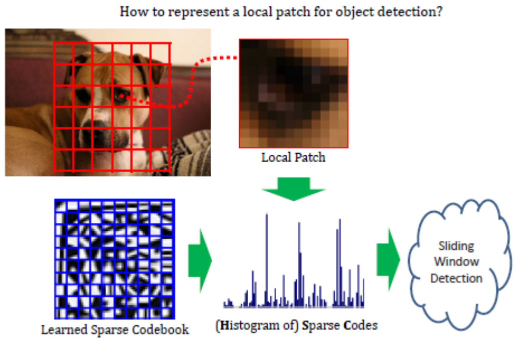

How to represent a local patch for object detection?
Local Patch
Learned Sparse Codebook
(Histogram of) Sparse Codes
Sliding Window Detection

Figure 1: Can we find better features than HOG for object detection? We develop Histograms-of-Sparse-Codes (HSC), which represents local patches through learned sparse codes instead of gradients and outperforms HOG by a large margin in state-of-the-art sliding window detection.

Histograms of Sparse Codes (HSC)

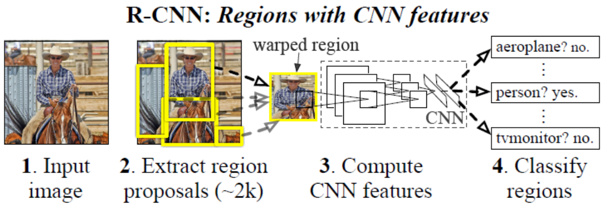
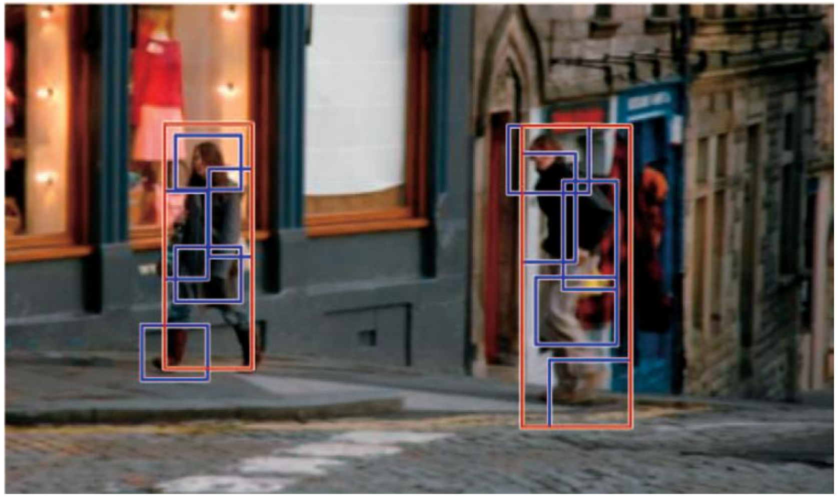

R-CNN: *Regions with CNN features*
warped region
aeroplane? no.
person? yes.
tvmonitor? no.
CNN
1. Input image
2. Extract region proposals (~2k)
3. Compute CNN features
4. Classify regions

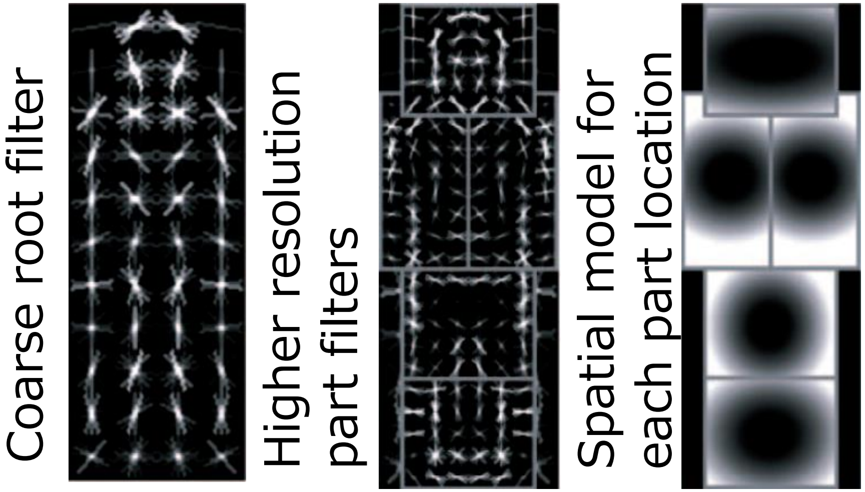
Figure 1: Object detection system overview. Our system (1) takes an input image, (2) extracts around 2000 bottom-up region proposals, (3) computes features for each proposal using a large convolutional neural network (CNN), and then (4) classifies each region using class-specific linear SVMs. R-CNN achieves a mean average precision (mAP) of **53.7% on PASCAL VOC 2010**. For comparison, [34] reports 35.1% mAP using the same region proposals, but with a spatial pyramid and bag-of-visual-words approach. The popular deformable part models perform at 33.4%.

Regions with CNN features (R-CNN)

Deformable Part Model (DPM) [1, 2]



物体のモデル化
(1) Root filter: 大まかな形状
(2) Parts model:
Part filters(移動可能な部品)
+ Spatial models(移動コスト)


Coarse root filter
Higher resolution part filters
Spatial model for each part location

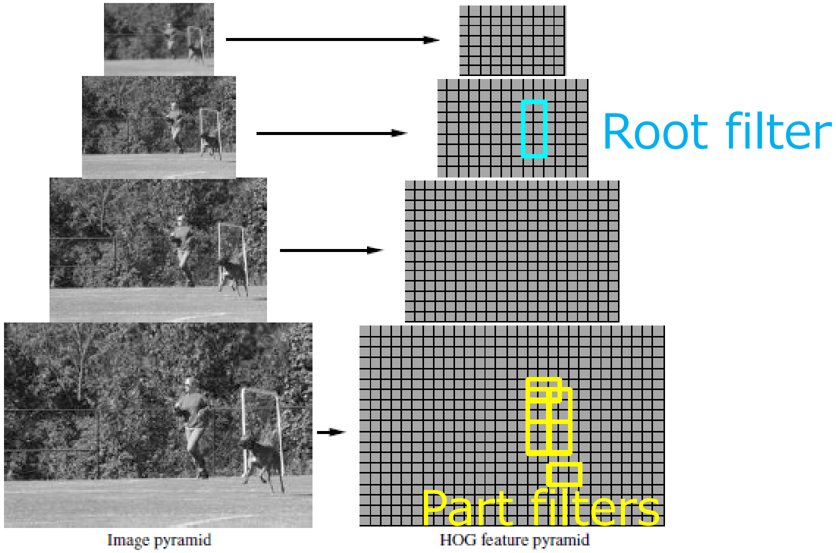


Figure 2. The HOG feature pyramid and an object hypothesis defined in terms of a placement of the root filter (near the top of the pyramid) and the part filters (near the bottom of the pyramid).

DPMの特徴量表現

- HOG(Histograms of Oriented Gradients)特徴量[Dalal et al., CVPR2005]を利用
- 8×8画素のオーバーラップのないセルを作り，画素毎の輝度勾配から勾配の大きさで重み付けした勾配方向ヒストグラムを構成→規格化
- 2×2セルのヒストグラムを連結して36次元のベクトルを作成 → 規格化した特徴ベクトルがHOG特徴量
- 複数解像度の画像（画像ピラミッド）に対して，8画素間隔(セルの幅)でHOG特徴量を算出

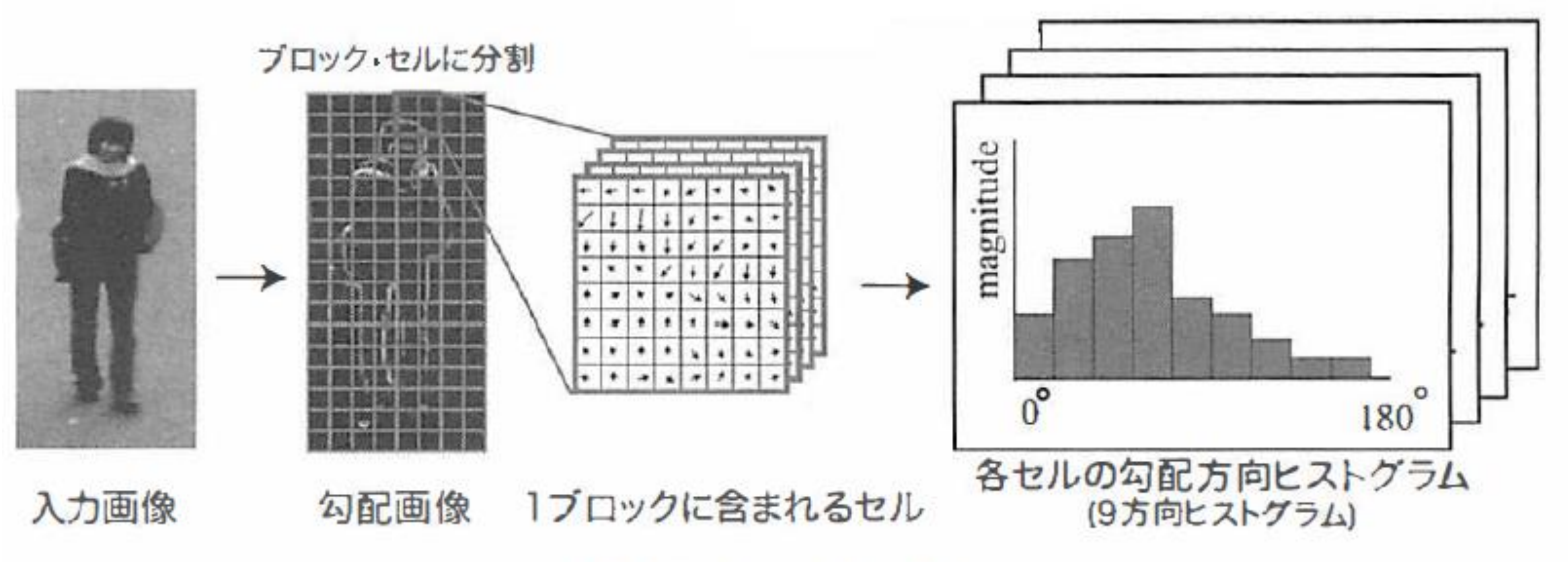


図 1.36 HOG 特徴量

[コンピュータビジョン最先端ガイド2, 第1章, 2010]

Filterの表現

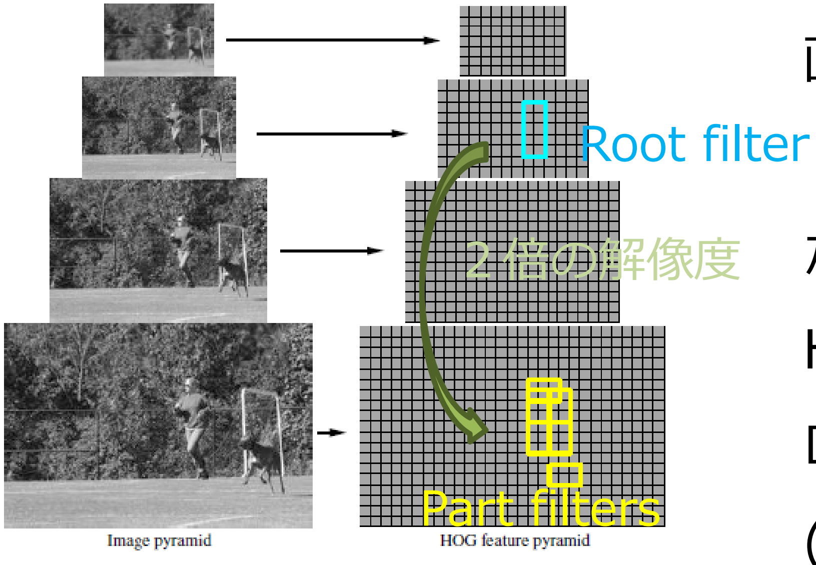
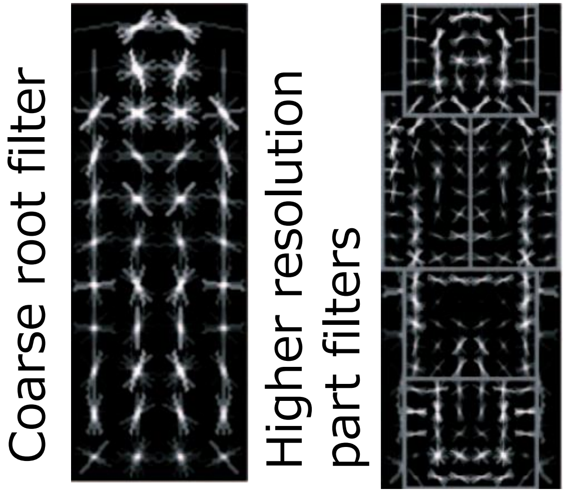


Figure 2. The HOG feature pyramid and an object hypothesis defined in terms of a placement of the root filter (near the top of the pyramid) and the part filters (near the bottom of the pyramid).



画像ピラミッドにおける位置を表すベクトル

$$p = (x, y, l)$$

ただし，lは画像ピラミッドの解像度レベル

HOGピラミッドHの位置pにおいて，$w \times h$ブロックのHOG特徴量を連結したベクトル($w \times h \times 36$次元)を

$$\phi(H, p, w, h)$$

$w \times h$次元のフィルタ係数ベクトルをFとしたときのフィルタ出力

$$F \cdot \phi(H, p, w, h) = F \cdot \phi(H, p)$$

$F_0 \cdot \phi(H, p_0)$ ：Root filter

$F_1 \cdot \phi(H, p_1), \cdots, F_n \cdot \phi(H, p_n)$：Part filters

評価関数

ある場所$z=(p_0,p_1,\cdots,p_n)$における評価関数（大きい値を取るようにzを求める）

$$\sum_{i=0}^{n} F_i\cdot\phi(H,p_i)-\sum_{i=1}^{n} d_i\cdot\phi_d(dx_i,dy_i)+b=\beta\cdot\psi(H,z)$$

$$\beta=(F_0,F_1,\cdots,F_n,d_1,\cdots,d_n,b)$$

$$\psi(H,z)=\underline{(\phi(H,p_0),\phi(H,p_1),\cdots,\phi(H,p_n),}$$
$$\underline{-\phi_d(dx_1,dy_1),\cdots,-\phi_d(dx_n,dy_n)},1)$$

HOG特徴量を連結したベクトル

ただし，

$$\phi_d(dx_i,dy_i)=\left(dx_i,dy_i,dx_i^2,dy_i^2\right)$$

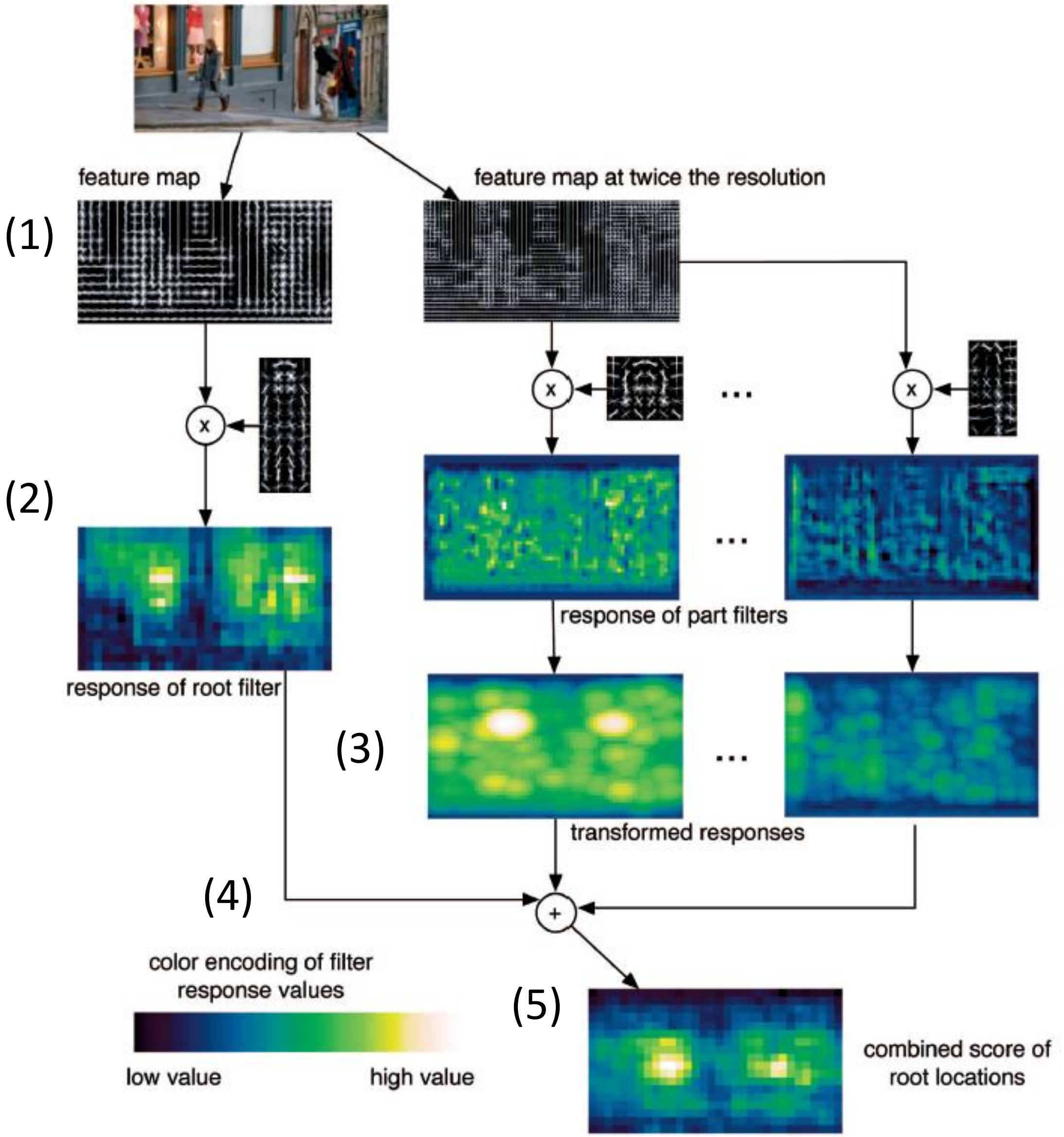
$$(dx_i,dy_i)=(x_i,y_i)-(2(x_0,y_0)+v_i)$$

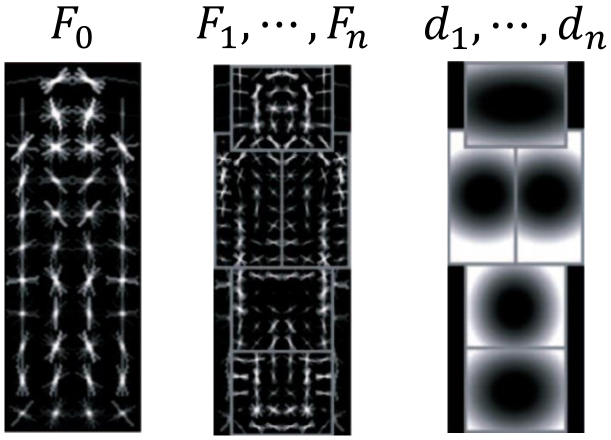
$$d_i=\underline{(d_{1i},d_{i2},d_{i3},d_{i4})}$$

各部品の位置ずれに対する重みパラメータ

アンカーポジション（ルートフィルタを基準とした各部品のデフォルト相対位置）

Matching (モデルと最も一致する場所の検出)





(1) 画像ピラミッドからHOGピラミッドを計算

(2) Root FilterとPart Filtersの応答を計算

(3) Part Filter応答は移動コストを考慮した応答を計算

(4) 全ての和を取り，評価関数の値を計算

(5) 値の大きい場所が物体検出結果

Learning (学習によるモデルパラメータ推定)

◆Latent Support Vector Machines (Latent SVM)

変数xに対する判別関数を以下で定義する

$$f_\beta(x) = \max_{z \in Z(x)} \beta \cdot \Phi(x, z)$$

ここで，βはモデルパラーメータ， zは潜在変数（DPMでは部品の位置$p_1, \cdots, p_n$が潜在変数）

通常のSVMにならって，与えられたデータ組$D = \big((x_1, y_1), \cdots, (x_N, y_N)\big)$から，以下の損失関数によりパラメータ$\beta$を求める。ただし，$y_i = 1/-1 \; for \; x \in \{positive\ samples\}/\{negative\ samples\}$

$$\beta^*(D) = \operatorname*{argmin}_{\beta} \left\{ \lambda \|\beta\|^2 + \sum_{i=1}^{N} \max\Big(0, 1 - y_i f_\beta(x_i)\Big) \right\}$$

$f_\beta(x)$はβに関して下に凸の関数なので，$y_i = -1$のとき$\max\Big(0, 1 - y_i f_\beta(x_i)\Big)$は下に凸の関数。したがって，Negative Samplesに対しては，上の最適化問題は，通常のSVMと同様に，凸最適化問題となり，最小値を比較的簡単に求めることができる（極小値が存在しない）

Latent SVMにおける最適化

$y_i = 1$のPositive samplesを含めると凸最適化問題にならないので，以下の反復計算によりβを求める。

1. モデルパラメータβを固定して，$z_i = \underset{z \in Z(x_i)}{\operatorname{argmax}} \beta \cdot \Phi(x, z)$によりPositive Examplesに対する潜在変数を最適化する
2. Positive Samplesに対する潜在変数z_iを固定して，凸最適化問題としてβを求める

Latent SVMの判別関数

$$f_\beta(x) = \max_{z \in Z(x)} \beta \cdot \Phi(x, z)$$

損失関数の最適化

$$\beta^*(D) = \underset{\beta}{\operatorname{argmin}} \left\{ \lambda \|\beta\|^2 + \sum_{i=1}^{N} \max\left(0, 1 - y_i f_\beta(x_i)\right) \right\}$$

Mixture Models

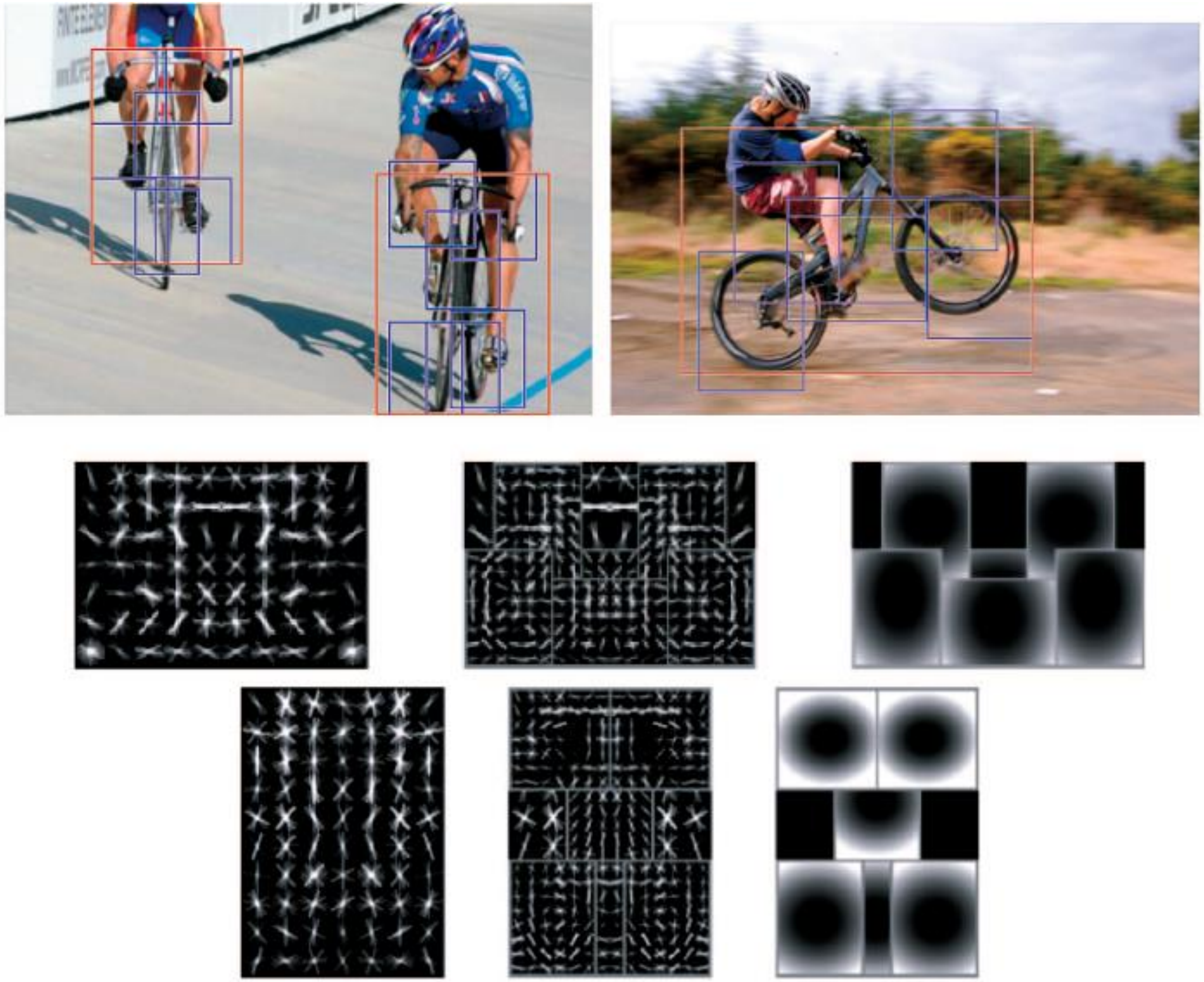
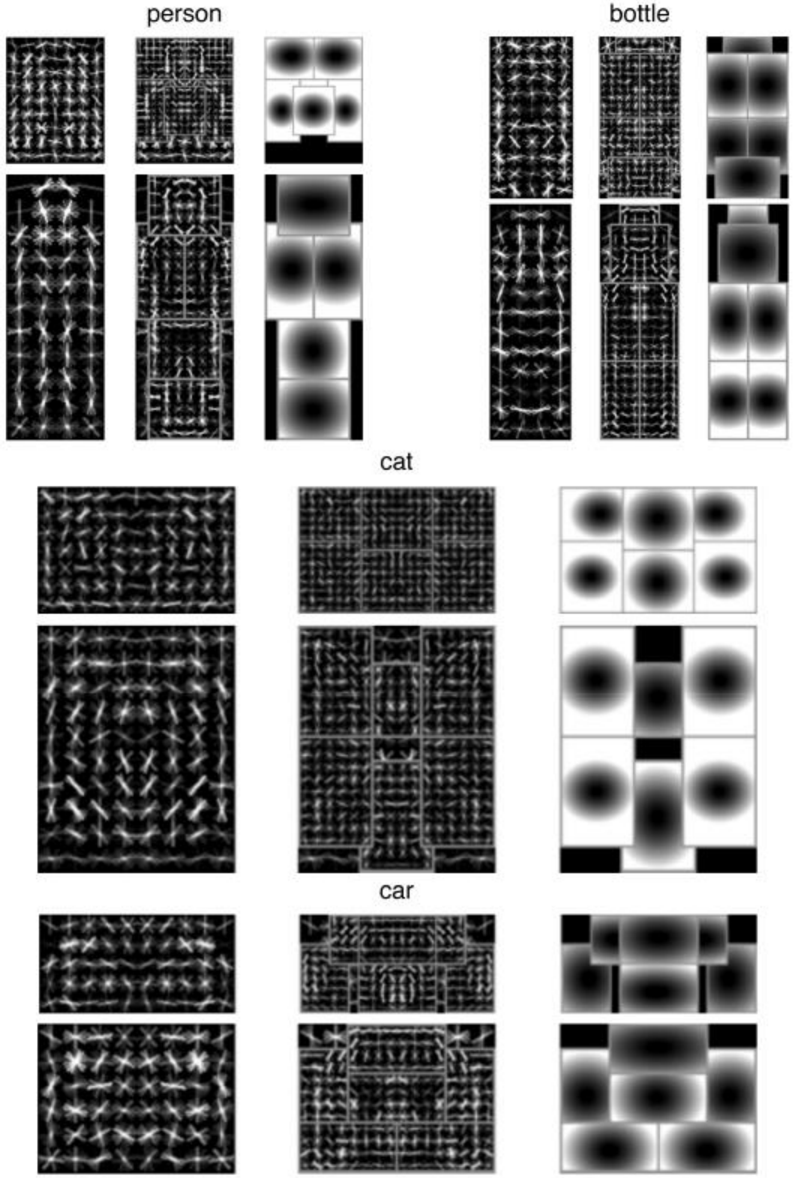
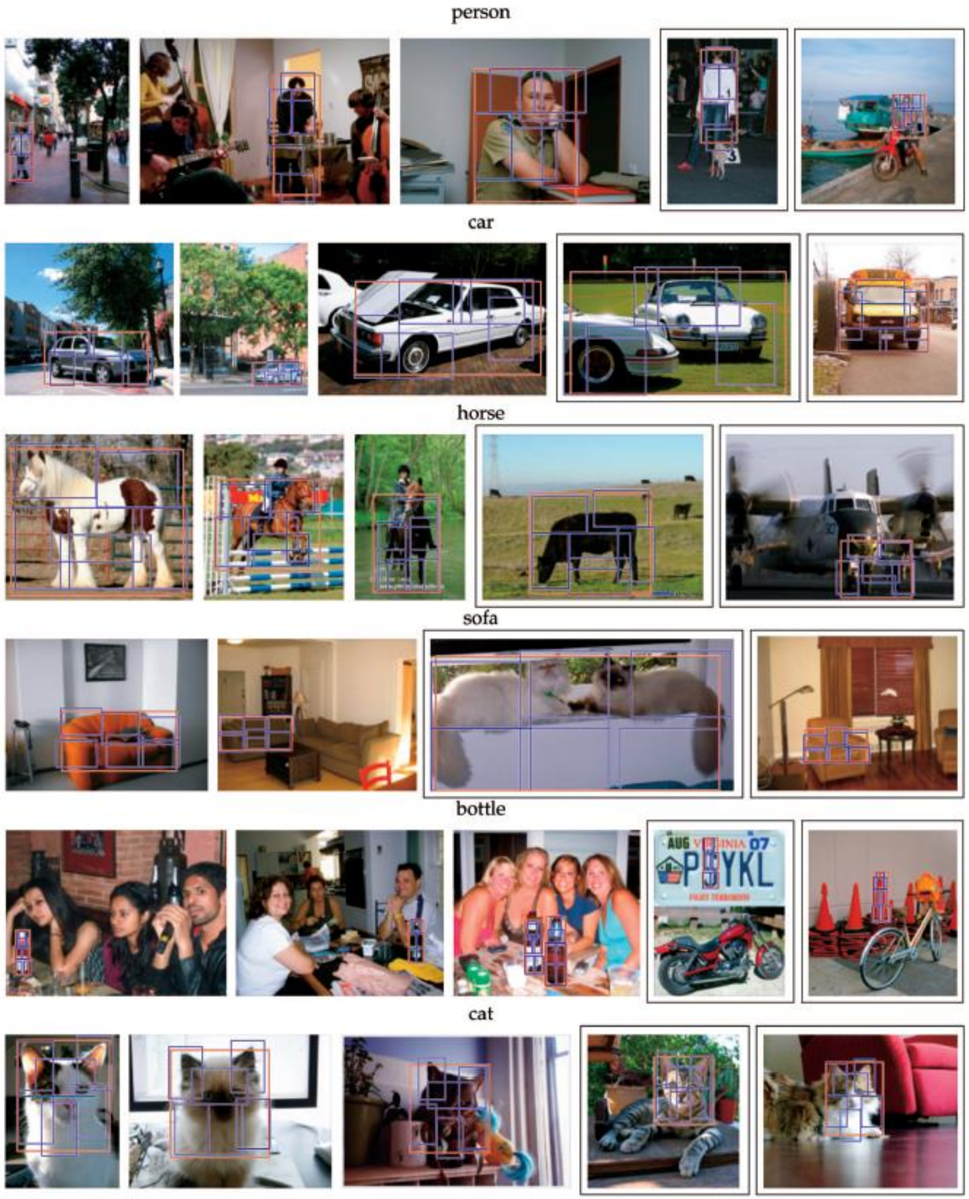


Fig. 2. Detections obtained with a two-component bicycle model. These examples illustrate the importance of deformations and mixture models. In this model, the first component captures sideways views of bicycles while the second component captures frontal and near frontal views. The sideways component can deform to match a “wheelie.”

Pascal 2007に対するDPMモデルと検出例





Deformable Part Modelとその発展

DPM [1, 2]

[1] P. Felzenszwalb, D. Mcallester, and D. Ramanan, “A Discriminatively Trained , Multiscale , Deformable Part Model,” in IEEE Conference on Computer Vision and Pattern Recognition, 2008.

[2] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, “Object detection with discriminatively trained part-based models,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 32, no. 9, pp. 1627–45, Sep. 2010.

HSC [3]

[3] X. Ren and D. Ramanan, “Histograms of Sparse Codes for Object Detection,” in IEEE Conference on Computer Vision and Pattern Recognition, 2013.

R-CNN [4]

[4] R. Girshick, J. Donahue, T. Darrell, and J. Malik, “Rich feature hierarchies for accurate object detection and semantic segmentation,” in IEEE Conference on Computer Vision and Pattern Recognition, 2014.

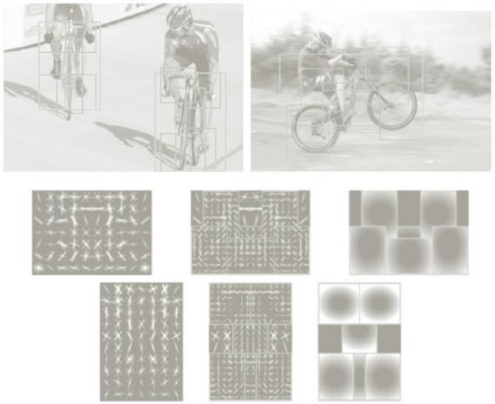


Fig. 2. Detections obtained with a two-component bicycle model. These examples illustrate the importance of deformations and mixture models. In this model, the first component captures sideways views of bicycles while the second component captures frontal and near frontal views. The sideways component can deform to match a “wheelie.”

Deformable Part Model (DPM)

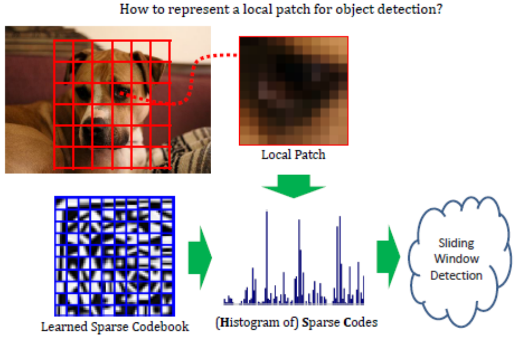

How to represent a local patch for object detection?
Local Patch
Learned Sparse Codebook
(Histogram of) Sparse Codes
Sliding Window Detection

Figure 1: Can we find better features than HOG for object detection? We develop Histograms-of-Sparse-Codes (HSC), which represents local patches through learned sparse codes instead of gradients and outperforms HOG by a large margin in state-of-the-art sliding window detection.

Histograms of Sparse Codes (HSC)

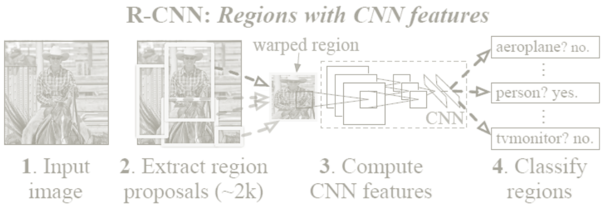
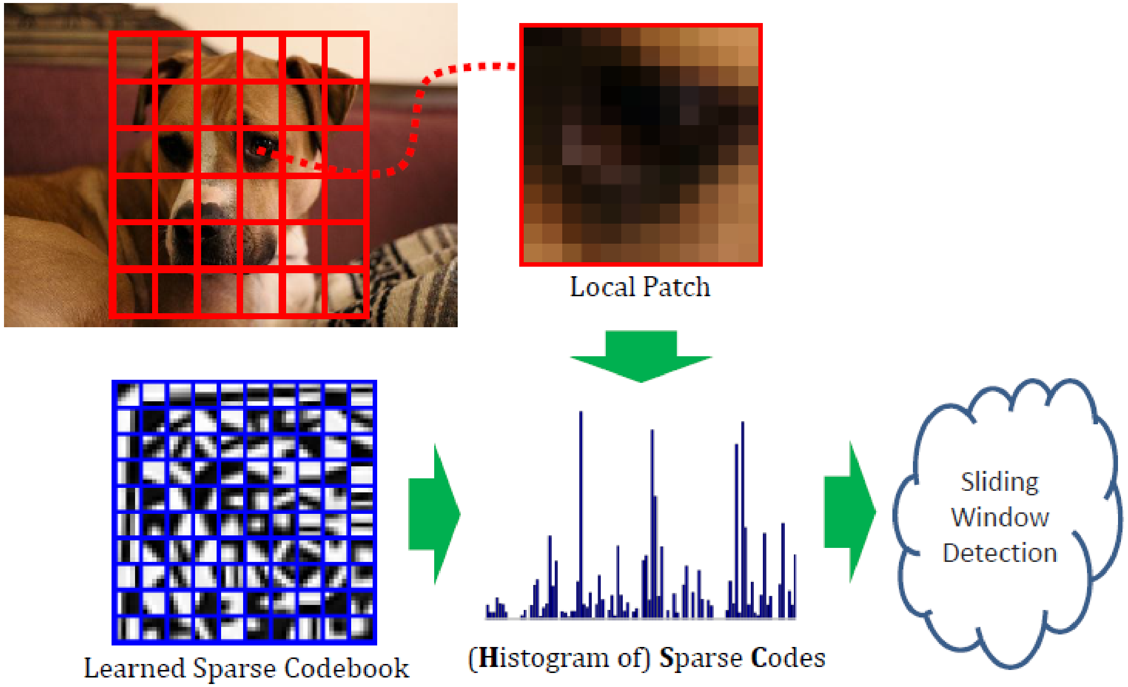


Figure 1: Object detection system overview. Our system (1) takes an input image, (2) extracts around 2000 bottom-up region proposals, (3) computes features for each proposal using a large convolutional neural network (CNN), and then (4) classifies each region using class-specific linear SVMs. R-CNN achieves a mean average precision (mAP) of **53.7% on PASCAL VOC 2010**. For comparison, [34] reports 35.1% mAP using the same region proposals, but with a spatial pyramid and bag-of-visual-words approach. The popular deformable part models perform at 33.4%.

Regions with CNN features (R-CNN)

Histograms of Sparse Codes (HSC) [3]

- DPMにおいて，HOG特徴ベクトルの代わりに，Sparse Codingにより求めた特徴ベクトル(HSC)を利用する
- Hand-designedな特徴量(HOG)よりも，データから学習した特徴量の方が表現力が豊かで，データから学習できるので応用範囲も広いのではないか？



Sparse Coding (1)

◆スパース符号化

観測信号を$\boldsymbol{x}$，辞書(基底ベクトルの集合)を$\boldsymbol{D}$としたときに，以下の式を満たす係数$\boldsymbol{c}$を求める

$$\min_{\boldsymbol{c}} \|\boldsymbol{x} - \boldsymbol{D}\boldsymbol{c}\|_2^2 + \lambda \|\boldsymbol{c}\|_0 \qquad \lambda > 0$$

ただし，$\|\boldsymbol{c}\|_0$は$\boldsymbol{c}$のl_0ノルムであり，$\boldsymbol{c}$の非ゼロの要素数を表す。

観測信号$\boldsymbol{x}$と辞書$\boldsymbol{D}$からスパース符号$\boldsymbol{c}$を求める代表的な手法：
Orthogonal Matching Pursuit (OMP) [Pati et al., Conf. Rec. 27th Asilomar Conf. Signals Syst. Comput., 1993]

```
-----------------------------------------------------------
produce $OMP\left(\boldsymbol{D} \in \mathbb{R}^{d\times m}, \boldsymbol{x} \in \mathbb{R}^{d}, \epsilon > 0\right)$
    $\boldsymbol{c} \leftarrow 0$
    $\boldsymbol{r} \leftarrow \boldsymbol{x} - \boldsymbol{D}\boldsymbol{c} = \boldsymbol{x}$
    $S = supp\{\boldsymbol{c}\} \leftarrow \emptyset$
    while $\|\boldsymbol{r}\|_2 \geq \epsilon$ do
        $\epsilon(j) \leftarrow \min_{z_j, j\notin S} \left\|\boldsymbol{d}_j z_j - \boldsymbol{r}\right\|_2^2$          基底の選択
        $S = S \cup \left\{\underset{j\notin S}{\operatorname{argmin}}\, \epsilon(j)\right\}$          非ゼロ成分の更新
        $\boldsymbol{c} \leftarrow \min_{\boldsymbol{c}, supp\{\boldsymbol{c}\}\subset S} \|\boldsymbol{D}\boldsymbol{c} - \boldsymbol{x}\|_2^2$          基底の更新
    end while
    return $\boldsymbol{c}$
end procedure
-----------------------------------------------------------
```

Sparse Coding (2)

◆Sparse Codingにおける辞書学習

複数の観測信号を$\boldsymbol{X}$としたとき，以下の式を満たす辞書(基底ベクトルの集合) $\boldsymbol{D}$と係数行列$\boldsymbol{C}$を求める

$$\min_{C} \|\boldsymbol{X} - \boldsymbol{D}\boldsymbol{C}\|_2^2 + \lambda \|\boldsymbol{C}\|_0 \qquad \lambda > 0$$

ただし，$\|\boldsymbol{C}\|_0 = \max_{i} \|\boldsymbol{c}_i\|_0$は列ごとの$l_0$ノルムの最大値である。

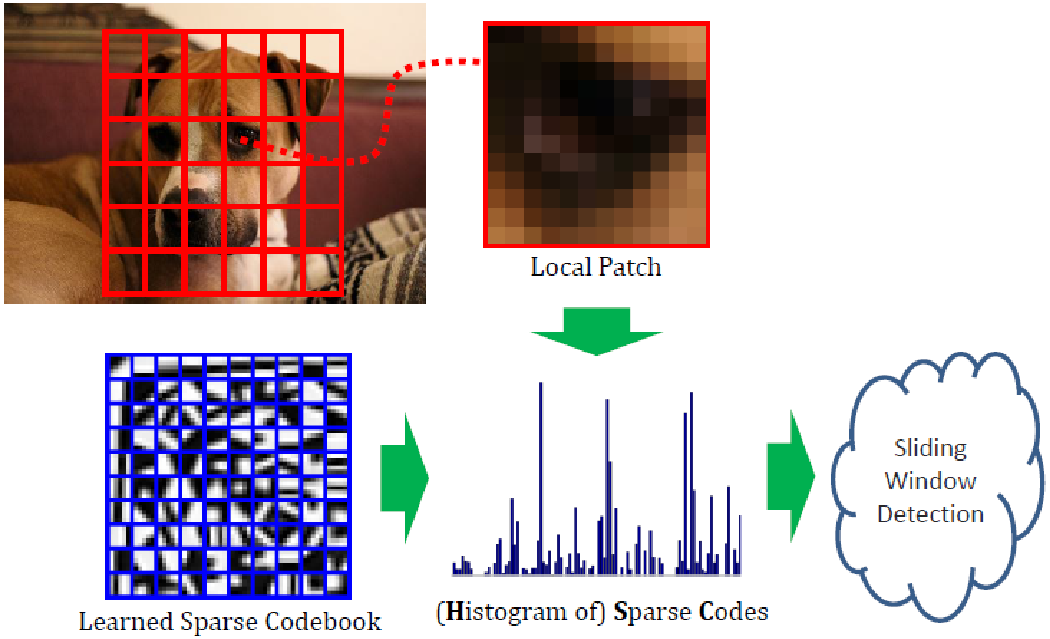
観測信号$\boldsymbol{x}$から辞書$\boldsymbol{D}$とスパース符号$\boldsymbol{c}$を求める代表的な手法：

k-SVD *[Aharon et al., IEEE TSP2006]*

```
------------------------------------------------------------
produce $KSVD\ (\boldsymbol{D} \in \mathbb{R}^{d \times m}, \boldsymbol{X} \in \mathbb{R}^{d \times n}, \boldsymbol{C} \in \mathbb{R}^{m \times n})$
    for $l = 1, \cdots, m$ do
        $\Omega_l \leftarrow \{i \in \{1, \cdots, n\} | C_{li} \neq 0\}$        基底$l$を用いる信号
        $R_l \leftarrow X^{\Omega_l} - \sum_{j \neq l} \boldsymbol{d}_j \boldsymbol{c}^j$        基底$l$を除いた残差
        calculate SVD $R_l = U \Sigma V^T$
        $\boldsymbol{d}_l \leftarrow \boldsymbol{u}_1$        基底$l$の更新
        $\boldsymbol{c}_l \leftarrow \sigma_1 \boldsymbol{v}_1$        係数の更新
    end for
    return $\boldsymbol{D}$ and $\boldsymbol{C}$
end procedure
------------------------------------------------------------
```

HSC特徴量

- 学習データから多数の画像パッチをとり，K-SVDにより辞書を学習する
- 得られた辞書を使ってOMPにより画像ピラミッドの各画素に対してスパース符号$\boldsymbol{c}$を求める（辞書の大きさのベクトルとなる）
- ベクトル$\boldsymbol{c}$の各要素に対して，$[|c_i|, \max(c_i, 0), \max(-c_i, 0)]$を連結して辞書の大きさx3次元のベクトルを求める
- 8×8画素のオーバーラップのないセルを作り，セル毎に周辺の2x2セル内で和を計算する → L2正規化を行ったベクトルをFとする
- $\bar{F} = F^{\alpha}$によりパワー正規化を行う（αはパラメータ）→ *HSC*特徴量



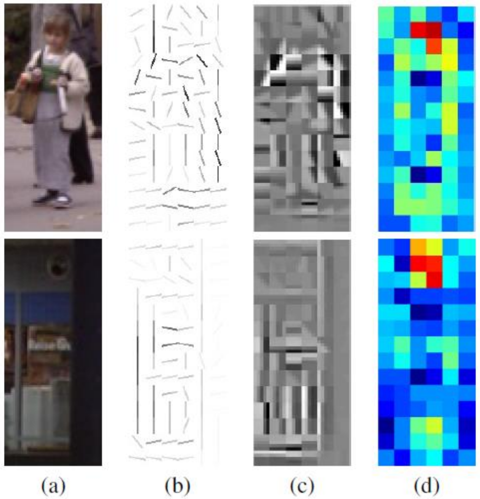
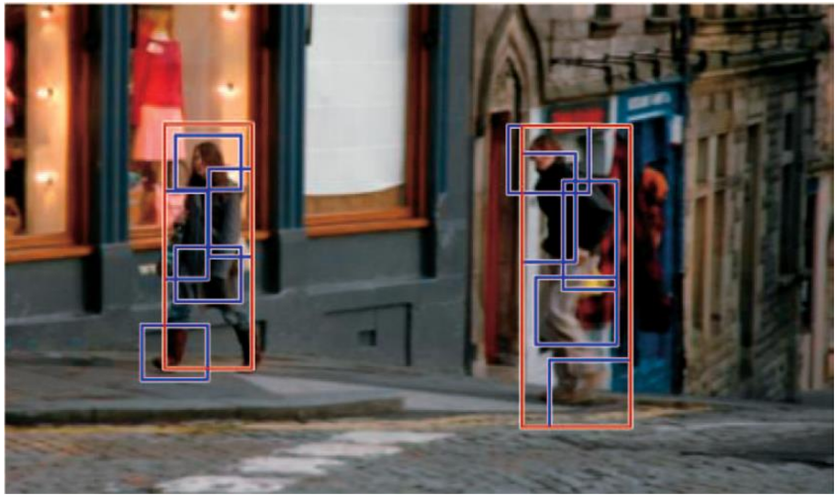
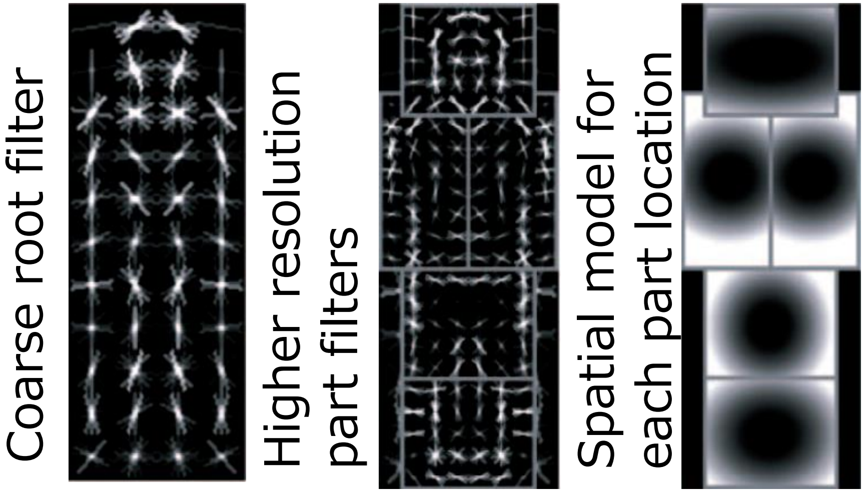


Figure 3: Visualizing HSC vs HOG: (a) image; (b) dominant orientation in HOG, weighted by gradient magnitude; (c) dominant codeword in HSC, weighted by histogram value; (d) per-cell responses of HSC features when multiplied with a linear SVM model trained on INRIA (colors are on the same scale).

DPM



- HOG特徴量をHSC特徴量に置き換える以外はオリジナルのDPMと全く同じ構成
- ただし，学習時に，各部品の最適な位置はオリジナルのDPMで計算を行い，その位置情報を利用（実装を簡単にするため？）


Coarse root filter
Higher resolution part filters
Spatial model for each part location

$$\sum_{i=0}^{n} F_i \cdot \phi(H, p_i) - \sum_{i=1}^{n} d_i \cdot \phi_d(dx_i, dy_i) + b$$

$i = 0$：Root filter応答

$i = 1, \cdots, n$：Part filter応答

mixture bias

Spatial models for part locations

実験

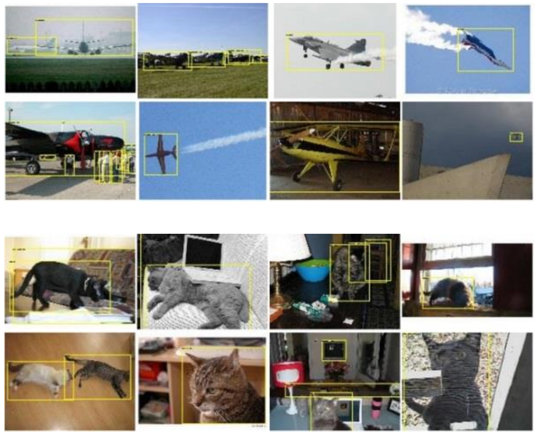
INRIA Person Dataset

- 学習データ：1208枚のPositiveサンプル（64x128画素, 立っている人の部分だけ抽出した画像），1218枚のNegativeサンプル
- テストデータ：741枚の画像
- Part filtersは用いずに，Root filterだけ用いて各種パラメータ（辞書サイズ，Sparsityレベル，パッチサイズ，パワー正規化）の影響を検討

[Dalal, PhD Thesis, 2006]
http://pascal.inrialpes.fr/data/human/

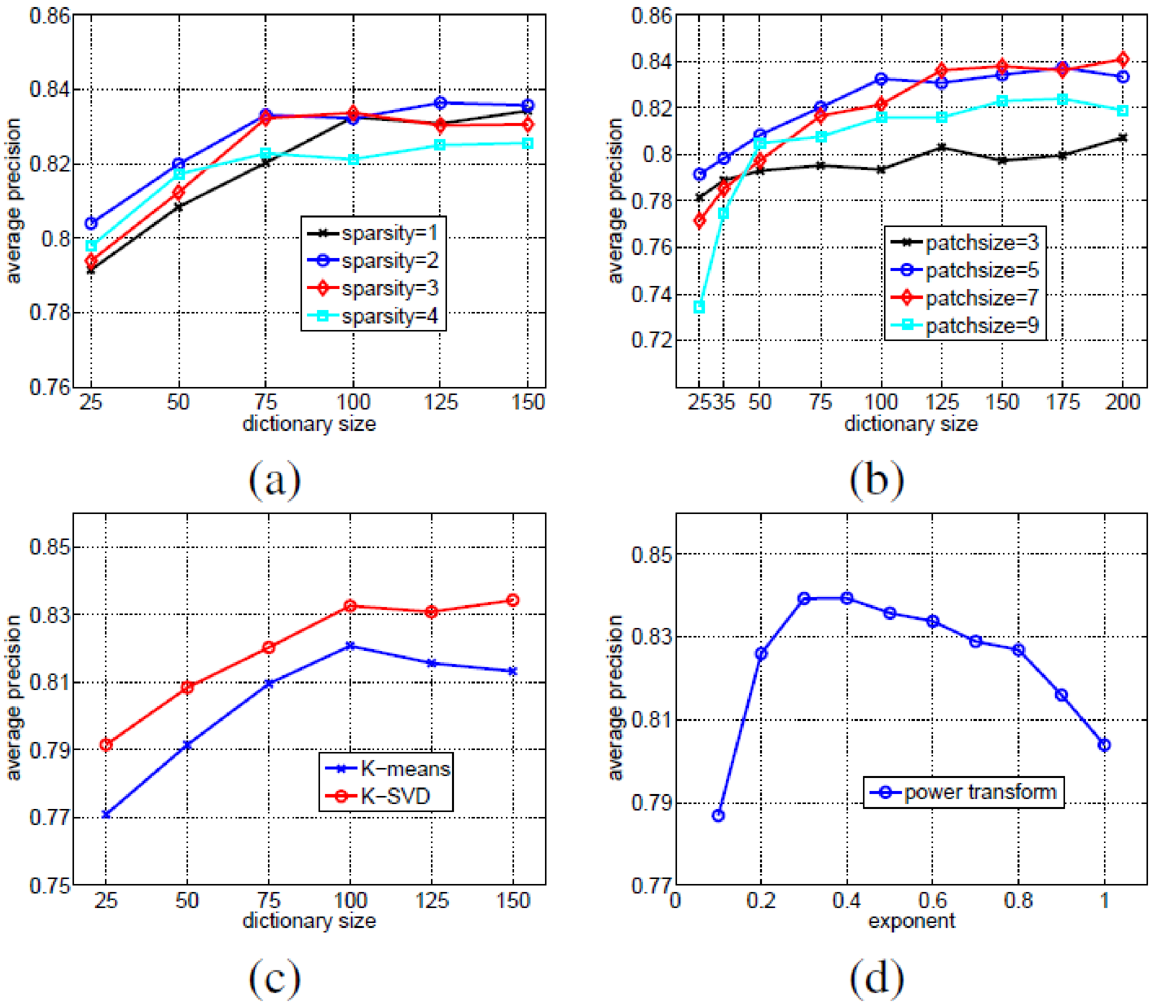
PASCAL 2007

- 9963枚の20クラスの画像
- 物体検出の評価によく利用されている標準的なデータセット
- Root filterだけ用いた場合とPart filtersもあわせて用いた場合の両方で評価



[Pascal VOC 2007]
http://pascallin.ecs.soton.ac.uk/challenges/VOC/voc2007/

INRIA Person Dataset



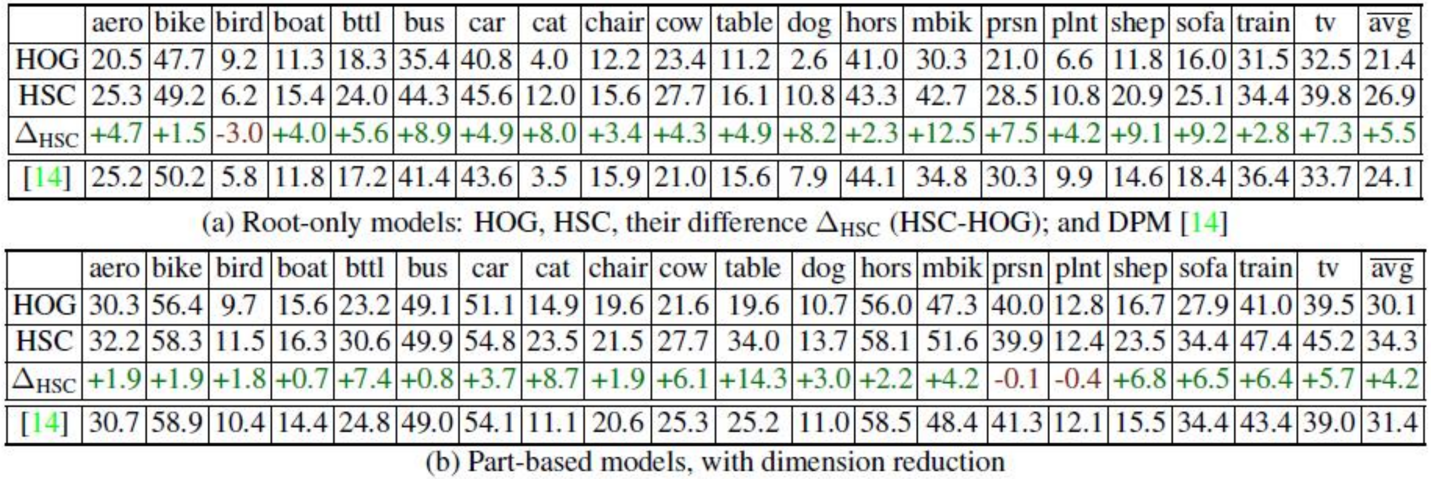
(a) (b)

(c) (d)

HOG	HSC_{3x3}	HSC_{5x5}	HSC_{7x7}	[14]
80.2%	80.7%	84.0%	84.9%	84.9%

HOG with part filters

PASCAL 2007



	aero	bike	bird	boat	bttl	bus	car	cat	chair	cow	table	dog	hors	mbik	prsn	plnt	shep	sofa	train	tv	$\overline{\text{avg}}$
HOG	20.5	47.7	9.2	11.3	18.3	35.4	40.8	4.0	12.2	23.4	11.2	2.6	41.0	30.3	21.0	6.6	11.8	16.0	31.5	32.5	21.4
HSC	25.3	49.2	6.2	15.4	24.0	44.3	45.6	12.0	15.6	27.7	16.1	10.8	43.3	42.7	28.5	10.8	20.9	25.1	34.4	39.8	26.9
Δ_{HSC}	+4.7	+1.5	-3.0	+4.0	+5.6	+8.9	+4.9	+8.0	+3.4	+4.3	+4.9	+8.2	+2.3	+12.5	+7.5	+4.2	+9.1	+9.2	+2.8	+7.3	+5.5
[14]	25.2	50.2	5.8	11.8	17.2	41.4	43.6	3.5	15.9	21.0	15.6	7.9	44.1	34.8	30.3	9.9	14.6	18.4	36.4	33.7	24.1

(a) Root-only models: HOG, HSC, their difference Δ_{HSC} (HSC-HOG); and DPM [14]

	aero	bike	bird	boat	bttl	bus	car	cat	chair	cow	table	dog	hors	mbik	prsn	plnt	shep	sofa	train	tv	$\overline{\text{avg}}$
HOG	30.3	56.4	9.7	15.6	23.2	49.1	51.1	14.9	19.6	21.6	19.6	10.7	56.0	47.3	40.0	12.8	16.7	27.9	41.0	39.5	30.1
HSC	32.2	58.3	11.5	16.3	30.6	49.9	54.8	23.5	21.5	27.7	34.0	13.7	58.1	51.6	39.9	12.4	23.5	34.4	47.4	45.2	34.3
Δ_{HSC}	+1.9	+1.9	+1.8	+0.7	+7.4	+0.8	+3.7	+8.7	+1.9	+6.1	+14.3	+3.0	+2.2	+4.2	-0.1	-0.4	+6.8	+6.5	+6.4	+5.7	+4.2
[14]	30.7	58.9	10.4	14.4	24.8	49.0	54.1	11.1	20.6	25.3	25.2	11.0	58.5	48.4	41.3	12.1	15.5	34.4	43.4	39.0	31.4

(b) Part-based models, with dimension reduction

Table 2: Results on the PASCAL2007 dataset. HSC and HOG results are from our supervised training system using identical settings and directly comparable. We achieve improvements over virtually all classes, in many cases by a large margin.

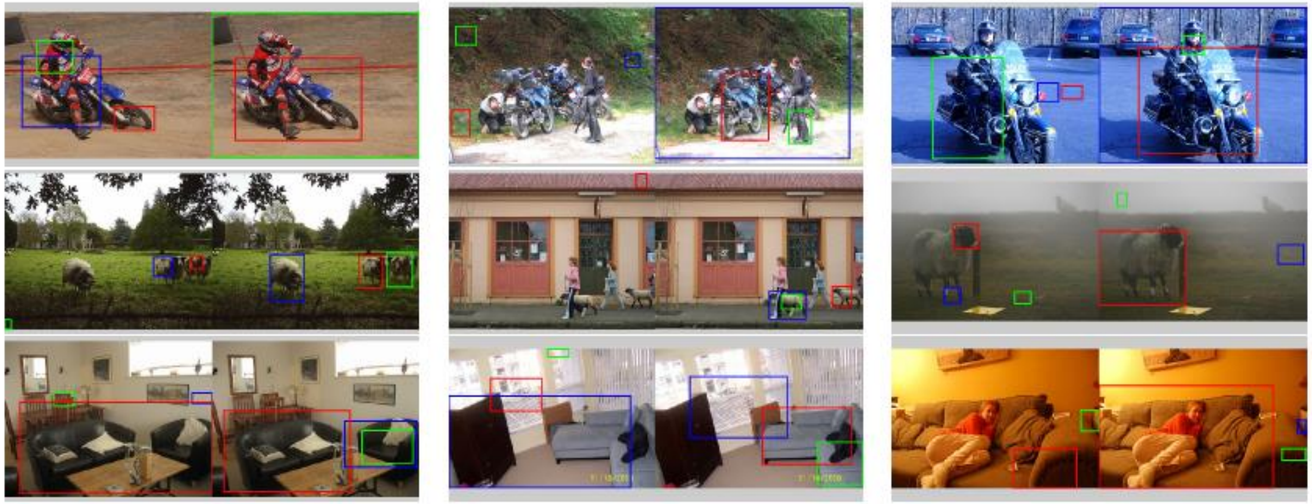


Figure 6: A few examples of HOG (left) vs HSC (right) based detection (root-only), showing top three candidates (in the order of red, green, blue). HSC behaves differently than HOG and tends to have different modes of success (and failure).

Deformable Part Modelとその発展

DPM [1, 2]

[1] P. Felzenszwalb, D. Mcallester, and D. Ramanan, “A Discriminatively Trained , Multiscale , Deformable Part Model,” in IEEE Conference on Computer Vision and Pattern Recognition, 2008.

[2] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, “Object detection with discriminatively trained part-based models,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 32, no. 9, pp. 1627–45, Sep. 2010.

HSC [3]

[3] X. Ren and D. Ramanan, “Histograms of Sparse Codes for Object Detection,” in IEEE Conference on Computer Vision and Pattern Recognition, 2013.

R-CNN [4]

[4] R. Girshick, J. Donahue, T. Darrell, and J. Malik, “Rich feature hierarchies for accurate object detection and semantic segmentation,” in IEEE Conference on Computer Vision and Pattern Recognition, 2014.

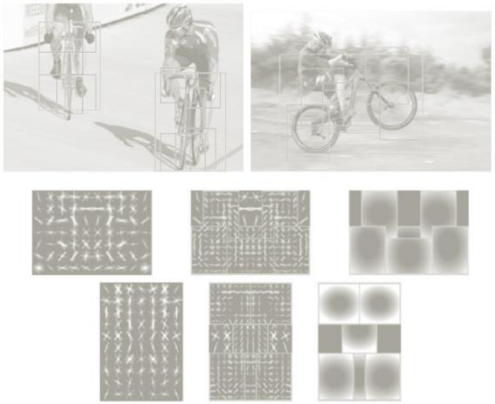


Fig. 2. Detections obtained with a two-component bicycle model. These examples illustrate the importance of deformations and mixture models. In this model, the first component captures sideways views of bicycles while the second component captures frontal and near frontal views. The sideways component can deform to match a “wheelie.”

Deformable Part Model (DPM)

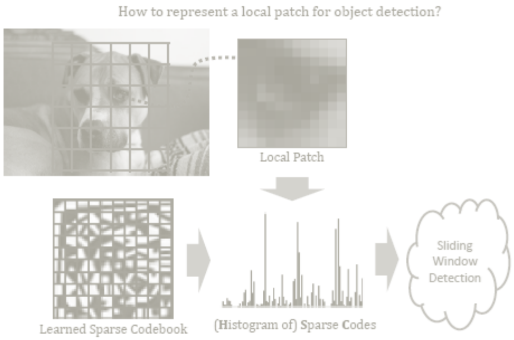

How to represent a local patch for object detection?
Local Patch
Learned Sparse Codebook
(Histogram of) Sparse Codes
Sliding Window Detection

Figure 1: Can we find better features than HOG for object detection? We develop Histograms-of-Sparse-Codes (HSC), which represents local patches through learned sparse codes instead of gradients and outperforms HOG by a large margin in state-of-the-art sliding window detection.

Histograms of Sparse Codes (HSC)

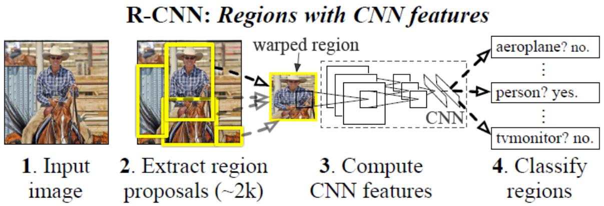


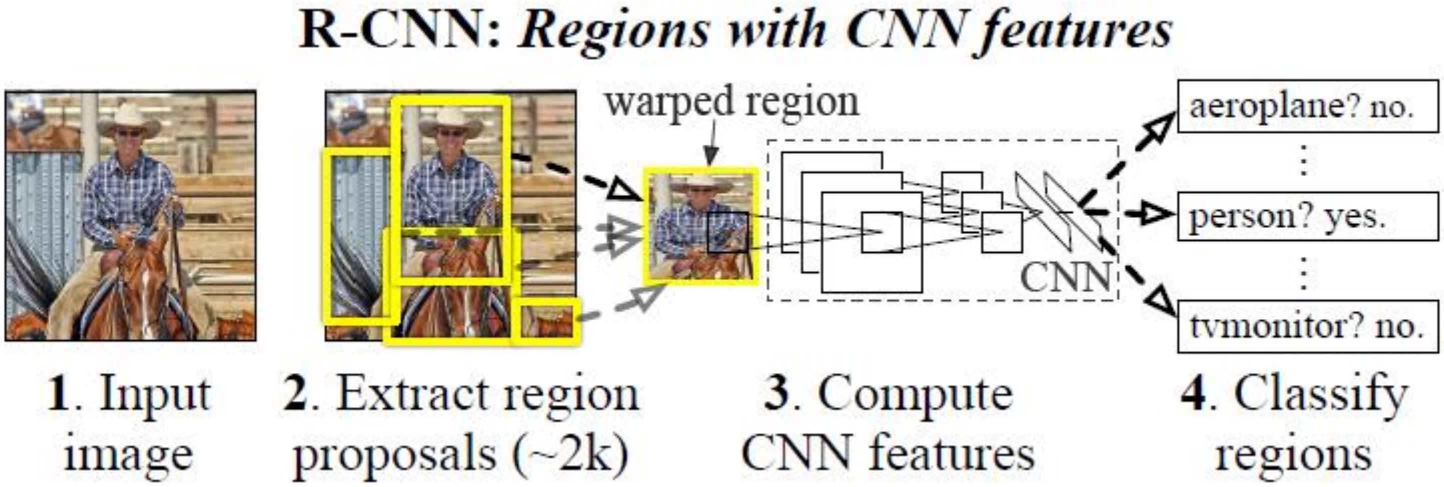
Figure 1: Object detection system overview. Our system (1) takes an input image, (2) extracts around 2000 bottom-up region proposals, (3) computes features for each proposal using a large convolutional neural network (CNN), and then (4) classifies each region using class-specific linear SVMs. R-CNN achieves a mean average precision (mAP) of **53.7% on PASCAL VOC 2010**. For comparison, [34] reports 35.1% mAP using the same region proposals, but with a spatial pyramid and bag-of-visual-words approach. The popular deformable part models perform at 33.4%.

Regions with CNN features (R-CNN)

Regions with CNN features (R-CNN) [4]

物体検出の精度及び検出速度を向上するため，従来用いられてきたものから以下のように改良

- Sliding windows → Region proposals
- Hand-designed features → deep leaned features
- deformable part-based model → linear SVM



Category-independent region proposals

- 従来Slideing Windowsの方式を利用（画像ピラミッドの全てのセルに対してDPMの評価値を計算し，最も一致する場所を検出）
  - 欠点(1) 画像ピラミット全体に対して網羅的に計算するため，計算コストが高い
  - 欠点(2) 物体のサイズ変化には対応できるが，アスペクト比（縦横比）の変化に対応することが難しい
- そこで，物体領域の候補となる場所を検出するアルゴリズムを利用 → 2,000個程度の候補領域に対して詳細に判定

  物体領域候補の検出アルゴリズム例

  (1) Objectness *[Alexe et al., TPAMI 2012]*

  (2) Selective Search *[Uijlings et al., IJCV2013]* → この手法を利用

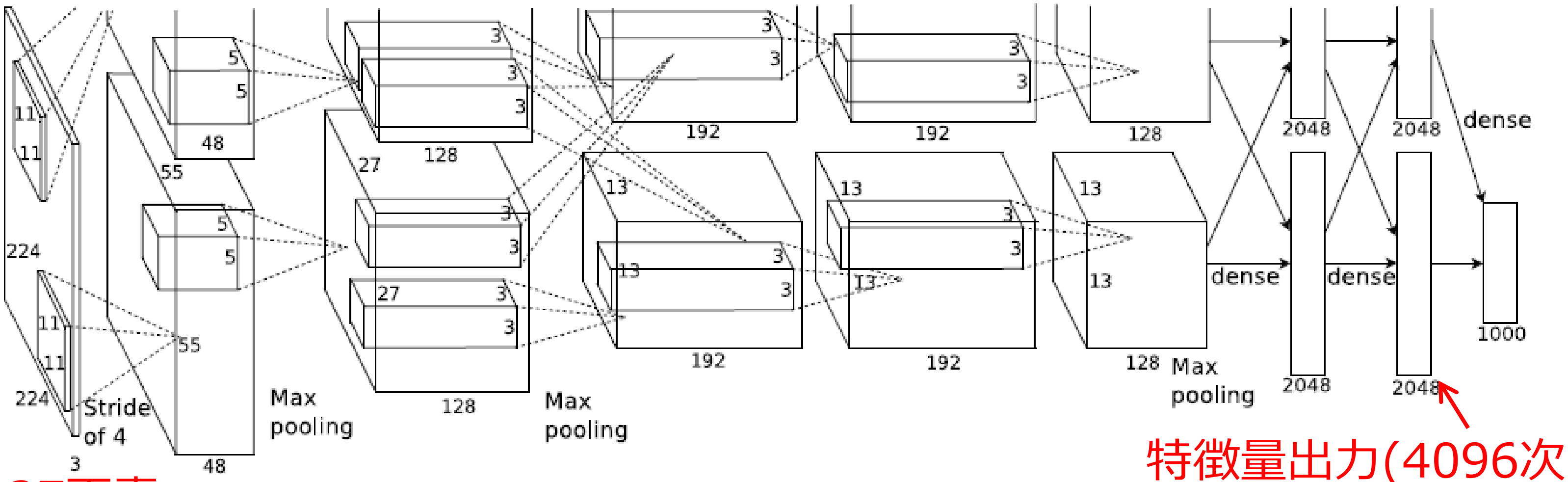
  (3) Category-independent object proposals *[Endres & Hoiem, ECCV2010]*

  (4) Constrained parametric min-cuts *[Carreira & Sminchisescu., TPAMI2012]*

  (5) Multi-scale combinatorial grouping *[Arbelaez et al., CVPR2014]*

Feature Extraction (1)

- SIFT/HOG特徴量：人の脳におけるV1の処理に対応
- 脳のより高次の機能も画像認識に有用では？→ 特徴抽出手法も階層的に行うことにより，画像認識に対して有効な情報抽出ができる可能性
- Object Recognition（物体認識）で近年高い性能を示している階層的な認識手法：Deep Convolutional Neural Network(DCNN)をObject Detection（物体検出）の特徴抽出手法として活用



227x227画素
RGB画像

特徴量出力(4096次元)

[Krizhevsky et al., NIPS 2012]

Feature Extraction (2)

- DCNNの入力は227x227画素の画像
- Selective Searchで検出された候補領域より少し大きい領域を227x227画素にリサイズして入力（周辺領域の情報を付け加えるため，リサイズ後のサイズで周囲16画素分）
- 特徴量出力は4096次元のベクトル（7層目の出力）

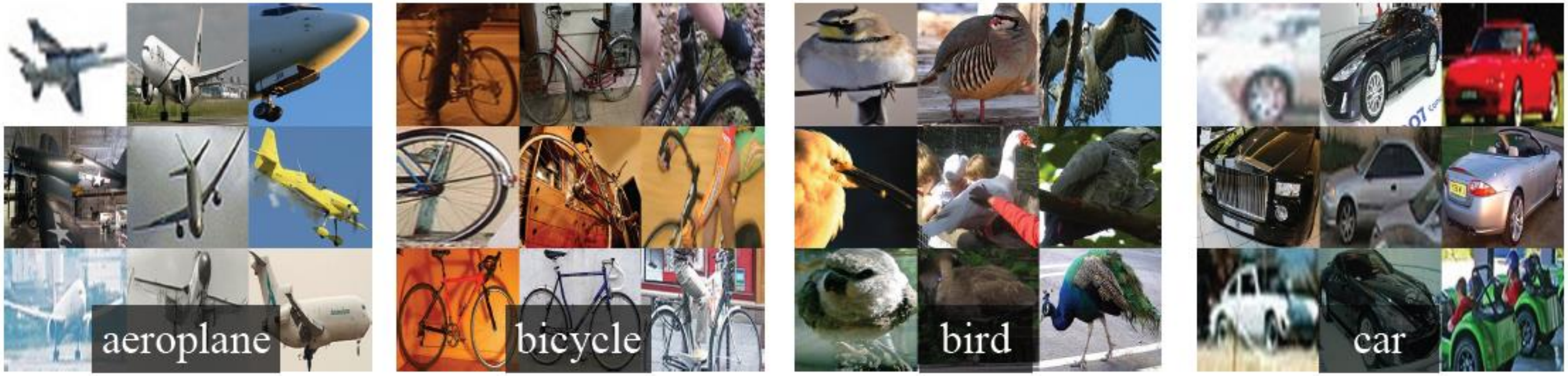


Figure 2: Warped training samples from VOC 2007 train.

DCNNの事前学習

DCNN

- Deep Learningでは事前学習に過学習を防ぐ効果があることが知られている
- そこで，評価するデータベース(PASCAL VOC 2007, 2010-12)とは異なる大規模データベース(ILSVRC 2012)を利用してDCNNの教師付き事前学習を行う
- その後，実際に評価に利用するデータベースで詳細な学習を行う。

SVMによる物体検出

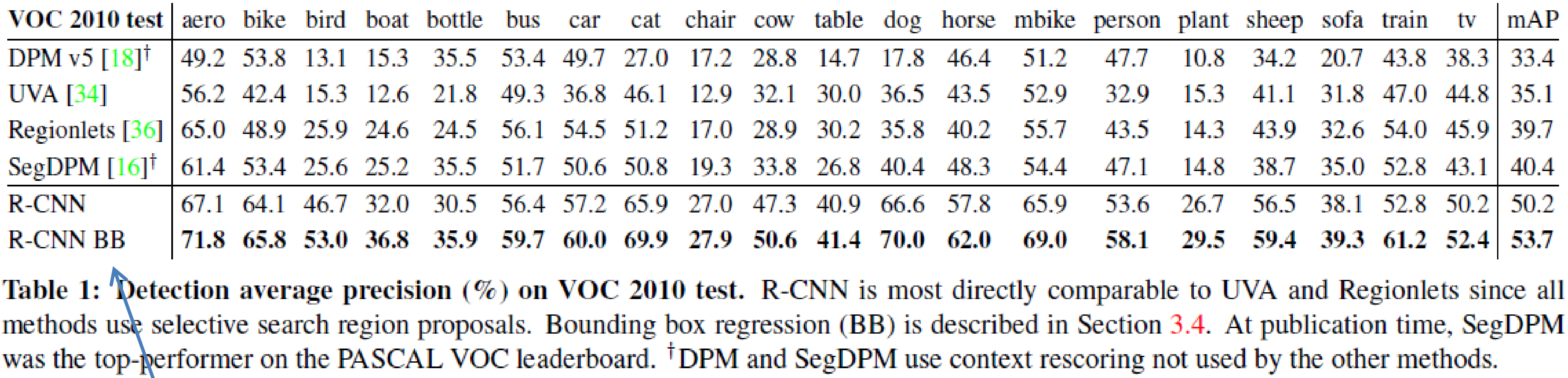
SVMによる詳細な判定

- DCNNで抽出された特徴ベクトルを各カテゴリごとに学習した線形SVMで識別
- オーバーラップした検出結果はSVMのスコアが小さい方を除去 (non-maximum suppression)

計算量の検討

- Selective SearchとDCNNはCategory-independentなので，全カテゴリに対して共通に計算できる。カテゴリ依存の計算は線形SVMの識別とnon-maximum suppressionだけで効率的に計算できる。
- 特徴量ベクトルのサイズも従来(e.g. 360k次元)に比べ，大幅に少なく，計算コスト・必要なメモリ容量の点で有利である。

実験結果 PASCAL VOC 2010



VOC 2010 test	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv	mAP
DPM v5 [18]†	49.2	53.8	13.1	15.3	35.5	53.4	49.7	27.0	17.2	28.8	14.7	17.8	46.4	51.2	47.7	10.8	34.2	20.7	43.8	38.3	33.4
UVA [34]	56.2	42.4	15.3	12.6	21.8	49.3	36.8	46.1	12.9	32.1	30.0	36.5	43.5	52.9	32.9	15.3	41.1	31.8	47.0	44.8	35.1
Regionlets [36]	65.0	48.9	25.9	24.6	24.5	56.1	54.5	51.2	17.0	28.9	30.2	35.8	40.2	55.7	43.5	14.3	43.9	32.6	54.0	45.9	39.7
SegDPM [16]†	61.4	53.4	25.6	25.2	35.5	51.7	50.6	50.8	19.3	33.8	26.8	40.4	48.3	54.4	47.1	14.8	38.7	35.0	52.8	43.1	40.4
R-CNN	67.1	64.1	46.7	32.0	30.5	56.4	57.2	65.9	27.0	47.3	40.9	66.6	57.8	65.9	53.6	26.7	56.5	38.1	52.8	50.2	50.2
R-CNN BB	**71.8**	**65.8**	**53.0**	**36.8**	**35.9**	**59.7**	**60.0**	**69.9**	**27.9**	**50.6**	**41.4**	**70.0**	**62.0**	**69.0**	**58.1**	**29.5**	**59.4**	**39.3**	**61.2**	**52.4**	**53.7**

Table 1: Detection average precision (%) on VOC 2010 test. R-CNN is most directly comparable to UVA and Regionlets since all methods use selective search region proposals. Bounding box regression (BB) is described in Section 3.4. At publication time, SegDPM was the top-performer on the PASCAL VOC leaderboard. †DPM and SegDPM use context rescoring not used by the other methods.

R-CNN BBは，線形SVMで物体領域を検出した後に，線形回帰でBounding Boxの位置を補正する補足的な処理

実験結果 PASCAL VOC 2007

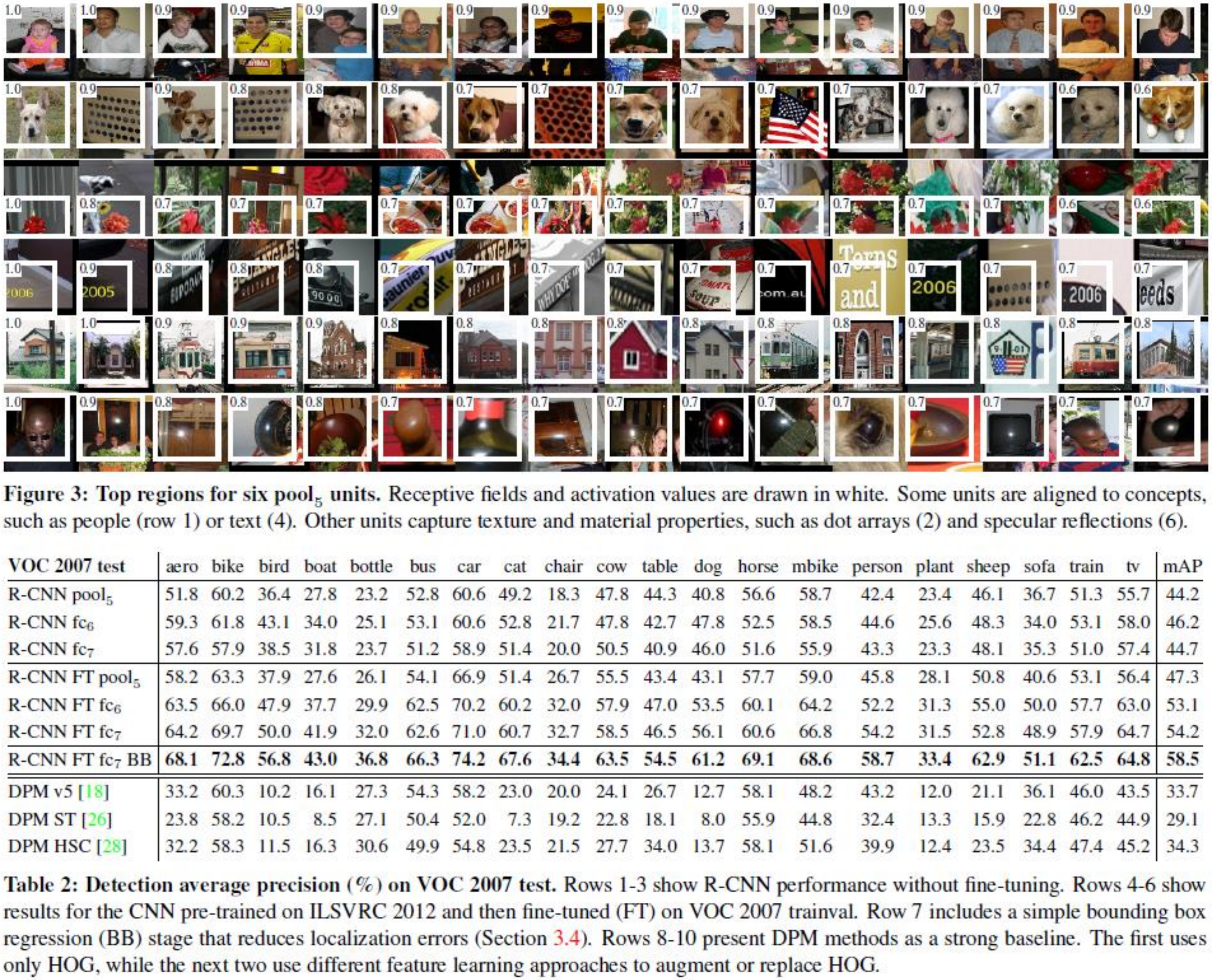

1.0 1.0 0.9 0.9 0.9 0.9 0.9 0.9 0.9 0.9 0.9 0.9 0.9 0.9 0.9 0.9
1.0 0.9 0.9 0.8 0.8 0.8 0.7 0.7 0.7 0.7 0.7 0.7 0.7 0.7 0.6 0.6
1.0 0.8 0.7 0.7 0.7 0.7 0.7 0.7 0.7 0.7 0.7 0.7 0.7 0.7 0.6 0.6
1.0 0.9 0.8 0.8 0.8 0.7 0.7 0.7 0.7 0.7 0.7 0.7 0.7 0.7 0.7 0.7
1.0 1.0 0.9 0.9 0.9 0.8 0.8 0.8 0.8 0.8 0.8 0.8 0.8 0.8 0.8 0.8
1.0 0.9 0.8 0.8 0.8 0.7 0.7 0.7 0.7 0.7 0.7 0.7 0.7 0.7 0.7 0.7

Figure 3: Top regions for six pool$_5$ units. Receptive fields and activation values are drawn in white. Some units are aligned to concepts, such as people (row 1) or text (4). Other units capture texture and material properties, such as dot arrays (2) and specular reflections (6).

VOC 2007 test	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv	mAP
R-CNN pool$_5$	51.8	60.2	36.4	27.8	23.2	52.8	60.6	49.2	18.3	47.8	44.3	40.8	56.6	58.7	42.4	23.4	46.1	36.7	51.3	55.7	44.2
R-CNN fc$_6$	59.3	61.8	43.1	34.0	25.1	53.1	60.6	52.8	21.7	47.8	42.7	47.8	52.5	58.5	44.6	25.6	48.3	34.0	53.1	58.0	46.2
R-CNN fc$_7$	57.6	57.9	38.5	31.8	23.7	51.2	58.9	51.4	20.0	50.5	40.9	46.0	51.6	55.9	43.3	23.3	48.1	35.3	51.0	57.4	44.7
R-CNN FT pool$_5$	58.2	63.3	37.9	27.6	26.1	54.1	66.9	51.4	26.7	55.5	43.4	43.1	57.7	59.0	45.8	28.1	50.8	40.6	53.1	56.4	47.3
R-CNN FT fc$_6$	63.5	66.0	47.9	37.7	29.9	62.5	70.2	60.2	32.0	57.9	47.0	53.5	60.1	64.2	52.2	31.3	55.0	50.0	57.7	63.0	53.1
R-CNN FT fc$_7$	64.2	69.7	50.0	41.9	32.0	62.6	71.0	60.7	32.7	58.5	46.5	56.1	60.6	66.8	54.2	31.5	52.8	48.9	57.9	64.7	54.2
R-CNN FT fc$_7$ BB	**68.1**	**72.8**	**56.8**	**43.0**	**36.8**	**66.3**	**74.2**	**67.6**	**34.4**	**63.5**	**54.5**	**61.2**	**69.1**	**68.6**	**58.7**	**33.4**	**62.9**	**51.1**	**62.5**	**64.8**	**58.5**
DPM v5 [18]	33.2	60.3	10.2	16.1	27.3	54.3	58.2	23.0	20.0	24.1	26.7	12.7	58.1	48.2	43.2	12.0	21.1	36.1	46.0	43.5	33.7
DPM ST [26]	23.8	58.2	10.5	8.5	27.1	50.4	52.0	7.3	19.2	22.8	18.1	8.0	55.9	44.8	32.4	13.3	15.9	22.8	46.2	44.9	29.1
DPM HSC [28]	32.2	58.3	11.5	16.3	30.6	49.9	54.8	23.5	21.5	27.7	34.0	13.7	58.1	51.6	39.9	12.4	23.5	34.4	47.4	45.2	34.3

Table 2: Detection average precision (%) on VOC 2007 test. Rows 1-3 show R-CNN performance without fine-tuning. Rows 4-6 show results for the CNN pre-trained on ILSVRC 2012 and then fine-tuned (FT) on VOC 2007 trainval. Row 7 includes a simple bounding box regression (BB) stage that reduces localization errors (Section 3.4). Rows 8-10 present DPM methods as a strong baseline. The first uses only HOG, while the next two use different feature learning approaches to augment or replace HOG.

実験結果 PASCAL VOC 2007

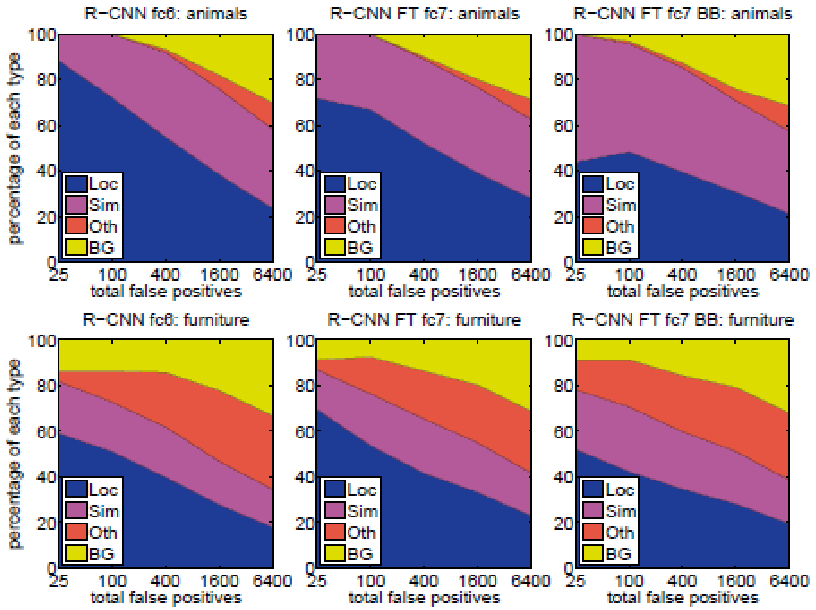


Figure 4: Distribution of top-ranked false positive (FP) types. Each plot shows the evolving distribution of FP types as more FPs are considered in order of decreasing score. Each FP is categorized into 1 of 4 types: Loc—poor localization (a detection with an IoU overlap with the correct class between 0.1 and 0.5, or a duplicate); Sim—confusion with a similar category; Oth—confusion with a dissimilar object category; BG—a FP that fired on background. Compared with DPM (see [21]), significantly more of our errors result from poor localization, rather than confusion with background or other object classes, indicating that the CNN features are much more discriminative than HOG. Loose localization likely results from our use of bottom-up region proposals and the positional invariance learned from pre-training the CNN for whole-image classification. Column three shows how our simple bounding box regression method fixes many localization errors.

[Hoiem *et al., ECCV2012*]

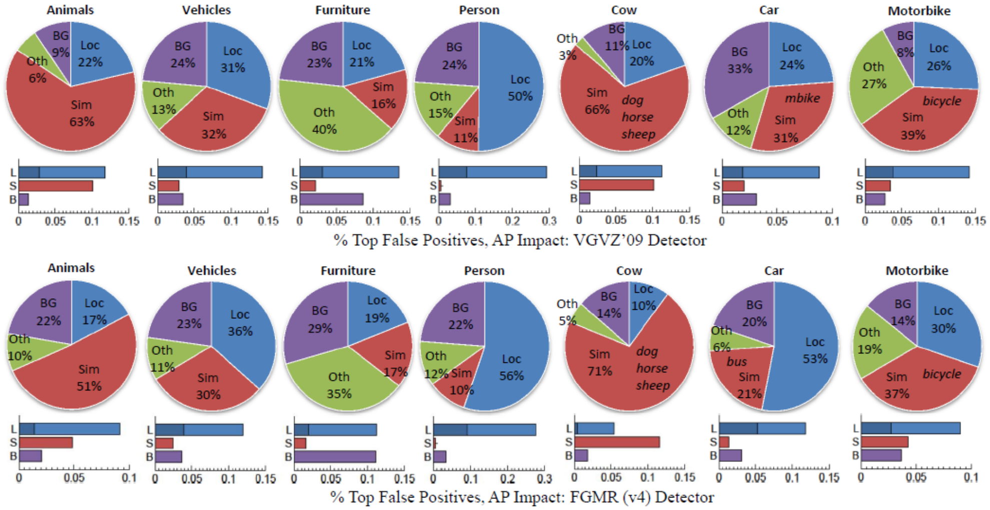


Fig. 2. Analysis of Top-Ranked False Positives. Pie charts: fraction of top-ranked false positives that are due to poor localization (Loc), confusion with similar objects (Sim), confusion with other VOC objects (Oth), or confusion with background or unlabeled objects (BG). Each category named within 'Sim' is the source of at least 10% of the top false positives. Bar graphs display absolute AP improvement by removing all false positives of one type ('B' removes confusion with background and non-similar objects). 'L': the first bar segment displays improvement if duplicate or poor localizations are removed; the second displays improvement if the localization errors were corrected, turning false detections into true positives.

実験結果 PASCAL VOC 2007

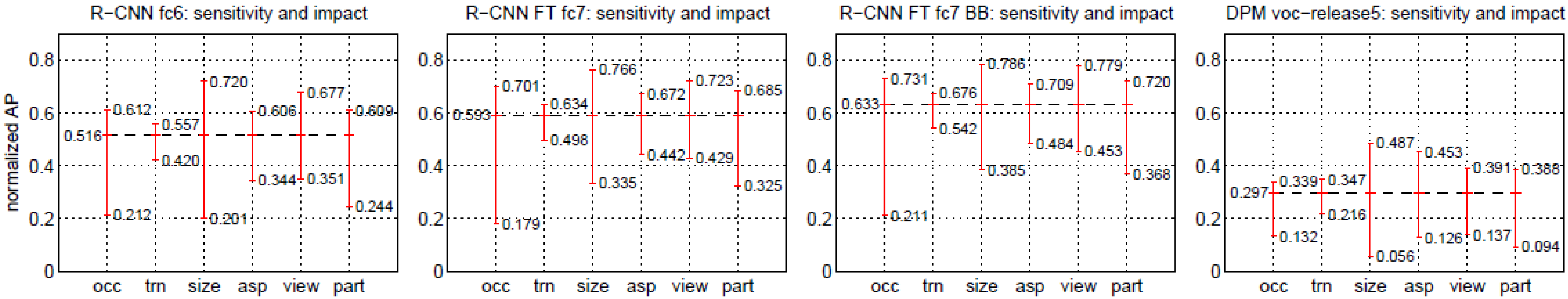


Figure 5: Sensitivity to object characteristics. Each plot shows the mean (over classes) normalized AP (see [21]) for the highest and lowest performing subsets within six different object characteristics (occlusion, truncation, bounding box area, aspect ratio, viewpoint, part visibility). We show plots for our method (R-CNN) with and without fine-tuning (FT) and bounding box regression (BB) as well as for DPM voc-release5. Overall, fine-tuning does not reduce sensitivity (the difference between max and min), but does substantially improve both the highest and lowest performing subsets for nearly all characteristics. This indicates that fine-tuning does more than simply improve the lowest performing subsets for aspect ratio and bounding box area, as one might conjecture based on how we warp network inputs. Instead, fine-tuning improves robustness for all characteristics including occlusion, truncation, viewpoint, and part visibility.

まとめ

- 物体検出の標準的な手法としてDeformable Part Modelを紹介した。
- 現在は，様々な改良手法が提案されているが，その中でもSparse Codingを利用した手法(HSC)とDeep Convolutional Neural Network (DCNN)を利用した手法について紹介した。
- DCNNの論文で紹介したように，近年高い性能を示している手法は，Selective Searchのように候補領域を予め検出する方法を用いていることが多い。このような手法は，Sliding Windowsのような画像の全領域を網羅的に計算する必要がないのに加えて，様々なアスペクト比にも対応することができる。
- DCNNで抽出した特徴量は，HOG特徴量に比べて高い記述能力を持つことが示唆されている。