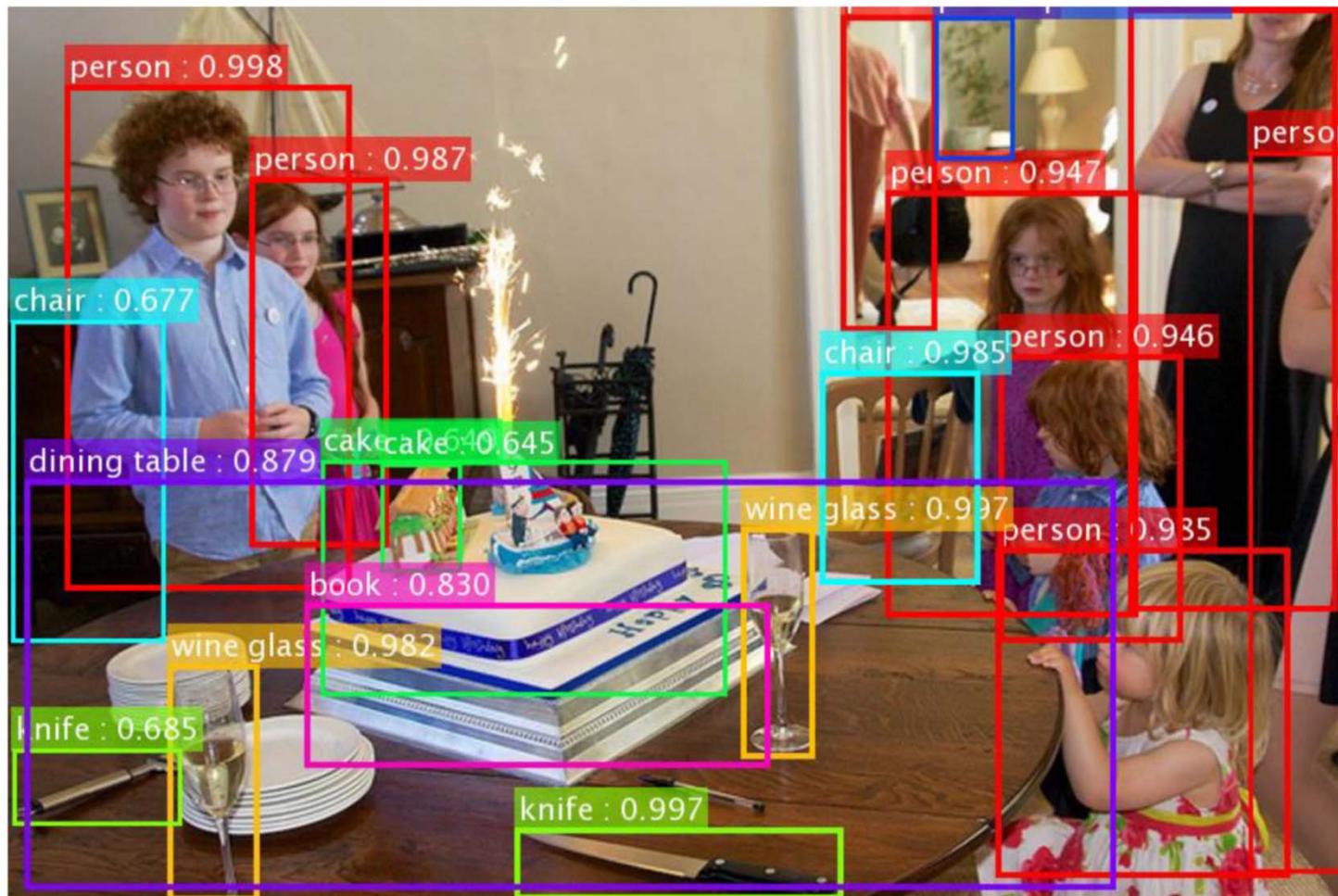


# 知能情報論 物体検出

2016年7月13日  
東京大学 大学院情報理工学系研究科  
原田達也

# MSRA's method

- [http://image-net.org/challenges/talks/ilsvrc2015\\_deep\\_residual\\_learning\\_kaiminghe.pdf](http://image-net.org/challenges/talks/ilsvrc2015_deep_residual_learning_kaiminghe.pdf)



# Viola Jones Face Detector

- リアルタイム物体検出手法
- 訓練は遅いが検出は非常に高速
- 主要なアイデア
  - 高速な特徴評価のための積分画像
  - 特徴抽出のためのBoosting
  - 非顔領域を高速に排除するためのAttentional cascade

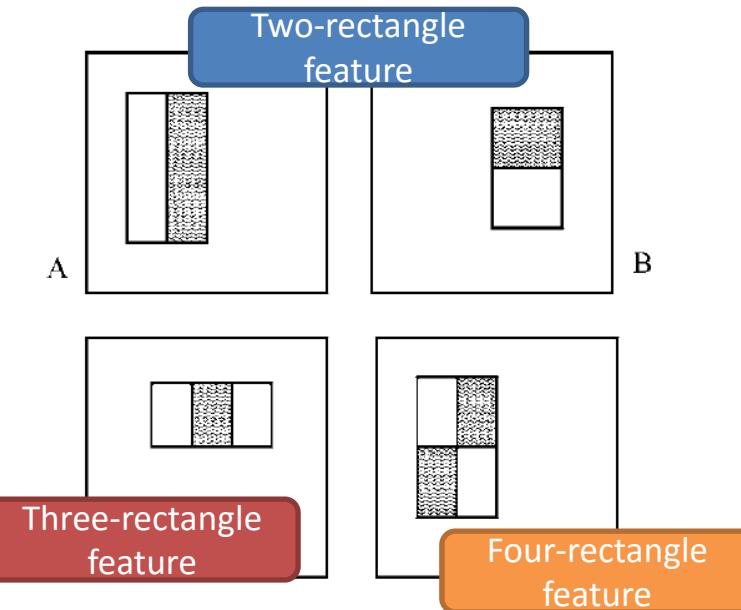


Viola and Jones. Robust Real-Time Face Detection. International Journal of Computer Vision 57(2), 137–154, 2004.

# 画像特徴

## Rectangle Features

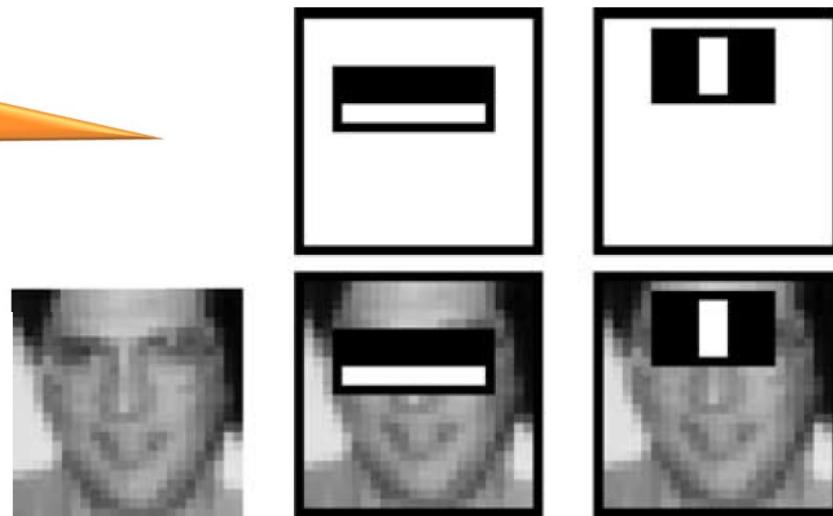
- Haar基底関数に類似
- 3種類の特徴
  - Two rectangle feature
  - Three rectangle feature
  - Four rectangle feature



特徴量の値 =  $\sum$  (白い領域のピクセルの値) -  $\sum$  (黒い領域のピクセルの値)

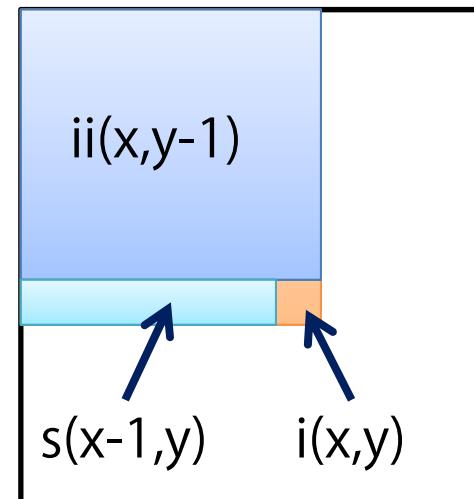
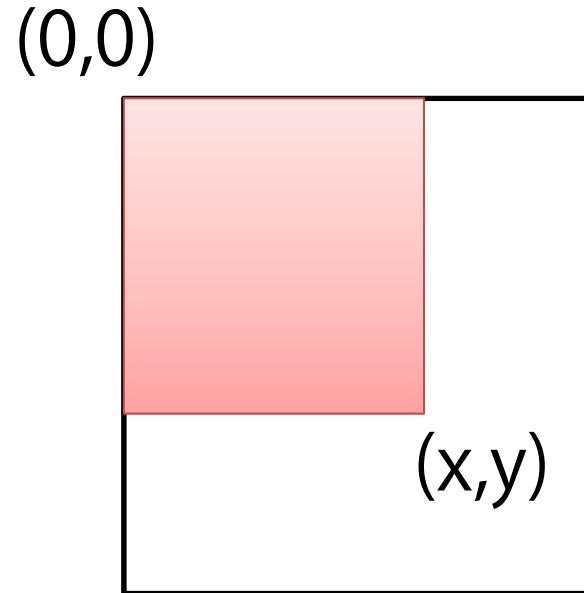
ちょうど目の位置と重なれば高い値が出力されると期待できる

→ うまいRectangle featureは弱い顔識別機と理解できる。



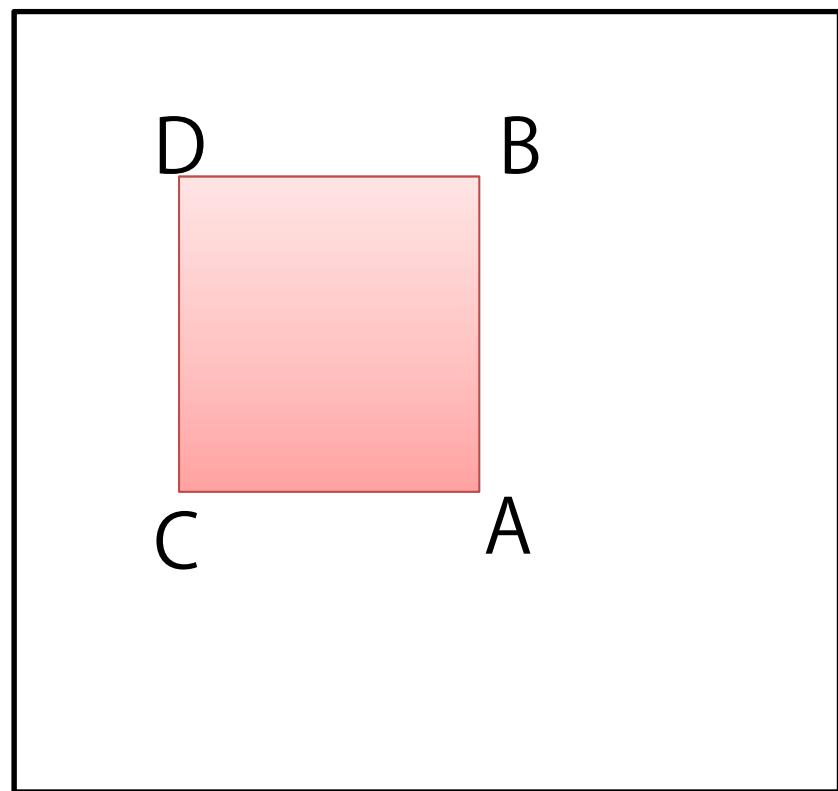
# 積分画像

- 積分画像 (integral image) は原点(0,0)と(x,y)を対角の頂点とする長方形で囲まれる領域内のピクセル値の和
- 一度のパスで高速計算可能
- 行のピクセル値の和
  - $s(x, y) = s(x-1, y) + i(x, y)$
- 積分画像
  - $ii(x, y) = ii(x, y-1) + s(x, y)$



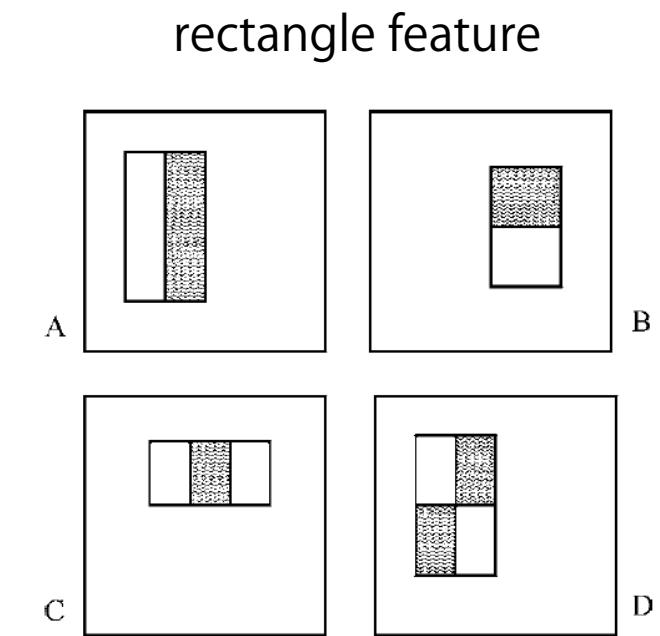
# 長方形領域内の和の計算

- A, B, C, Dをその位置における積分画像の値とする.
- 長方形領域内の画素値の合計
  - $A - B - C + D$
- 3加算だけで任意の長方形領域内の画素値の和を計算可能

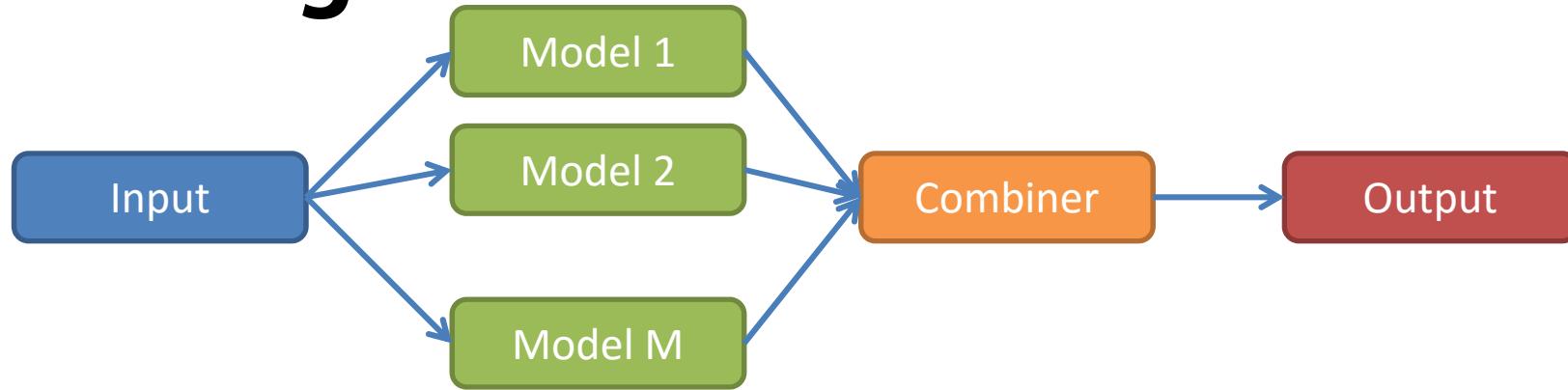


# 特徴選択

- 24×24の検出領域において、可能なrectangle featureの数は約160,000
- 検出時に全てのrectangle featureを評価するのは非現実的
- 全てのfeatureを利用せず、その一部のみを利用して良い識別器を構築するはどうすればよいか?
  - Boostingの利用



# Boosting



- Boostingとは
  - 複数の基本となる学習機械を組み合わせることで、精度の高い学習機械を構成する手法.
  - 以前に学習した学習機械の結果を用いて間違いの多い訓練サンプルに重み付けを行い、この重み付けされたサンプルで新たに学習機械を学習する.
  - 逐次的に生成される学習機械を統合して一つの学習機械とする.
- AdaBoostにおける訓練
  1. 各訓練サンプルに等価な重みを与える
  2. For  $m=1:M$ 
    - A) 訓練エラーを最小とする弱学習器を選択する
    - B) 現在の弱学習器によって誤識別された訓練サンプルの重みを増加
  3. 全ての弱学習器を重み付け線形結合し、最終的な識別器を構成する
    - 各弱学習器の重み付けは、識別精度に比例

# AdaBoostのアルゴリズム

- 各訓練サンプルに等価な重みを与える

$$\left\{ w_n^{(1)} = \frac{1}{N} \right\}_{n=1}^N$$

- For  $m=1:M$

- 訓練エラーを最小とする弱学習器を選択する

$$J_m = \sum_{n=1}^N w_n^{(m)} I(h_m(x_n) \neq y_n)$$

$$I(h_m(x_n) \neq y_n) = \begin{cases} 1 & \text{if } h_m(x_n) \neq y_n \\ 0 & \text{otherwise} \end{cases}$$

- 現在の弱学習器によって誤識別された訓練サンプルの重みを増加

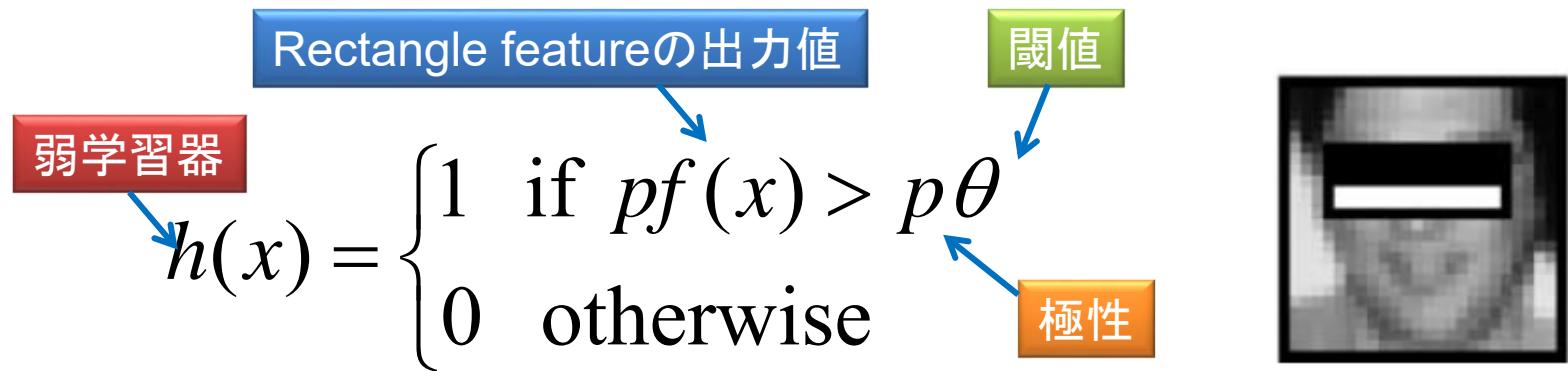
$$\begin{aligned} \epsilon_m &= \frac{\sum_{n=1}^N w_n^{(m)} I(h_m(x_n) \neq y_n)}{\sum_{n=1}^N w_n^{(m)}} & w_n^{(m+1)} &= w_n^{(m)} \exp\{\alpha_m I(h_m(x_n) \neq y_n)\} \\ \alpha_m &= \ln \left( \frac{1 - \epsilon_m}{\epsilon_m} \right) \end{aligned}$$

- 全ての弱学習器を重み付け線形結合し、最終的な識別器を構成する
  - 各弱学習器の重み付けは、識別精度に比例

$$H(x) = \operatorname{sign} \left( \sum_{m=1}^M \alpha_m h_m(x) \right)$$

# 顔検出におけるBoosting

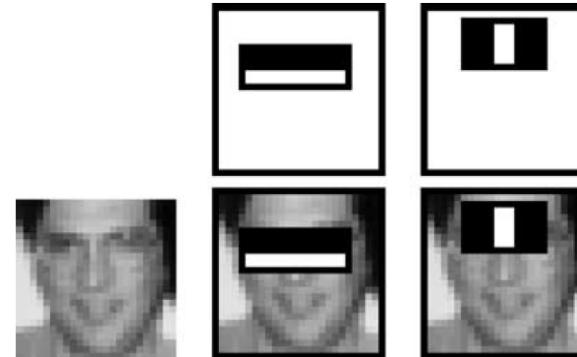
- Rectangle featureに基づく弱学習器の定義
  - 弱学習器とrectangle featureが1対1対応



- For  $m=1:M$ 
  - 各訓練サンプルに、弱学習器を適用
  - 各弱学習器に最良の閾値を選択
  - 最良の弱学習器と閾値の組合せを選択
  - 訓練サンプルに再重みづけ

# Boostingによって選択された特徴

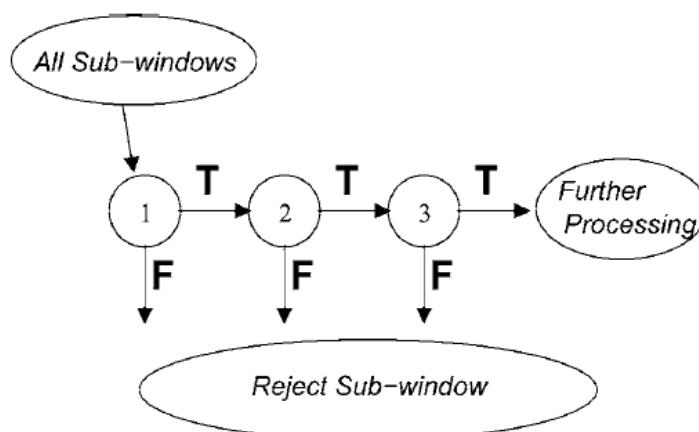
- Boostingによって選択された1番目, 2番目のrectangle feature
  - 100%の検出率
  - 50%のfalse positive



- 200個のfeatureを用いることで95%の検出率, 1/14,084のfalse positive
- 性能が不十分
  - False positiveを1/1,000,000未満としたい
  - Featureを増やせばfalse positiveを低下させられるが計算時間がかかる

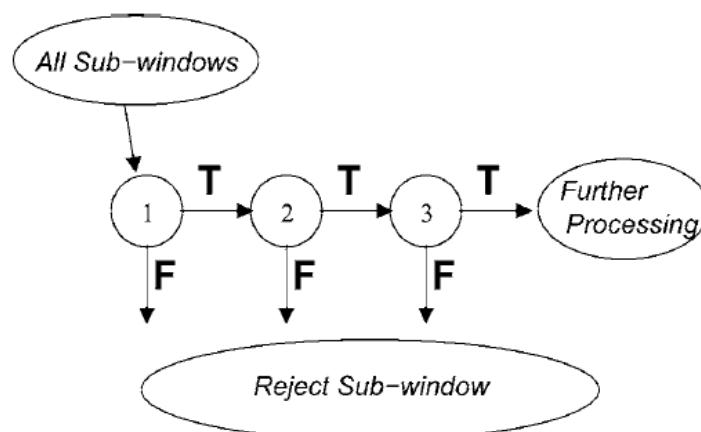
# Attentional cascade

- ほとんどのサブウィンドウが非顔画像であるので、非顔画像を素早く判定することが高速化の鍵
- ほとんど全ての顔画像を検出し、多くの非顔画像を拒絶する簡単な識別器から開始
  - 顔画像は必ず顔と判断するが、顔でない画像も顔として判断される可能性がある
- 第1識別器で顔と判断されたデータは、第2識別器に入力される
  - 第2識別器は第1識別器よりも複雑で顔でない画像を顔と判断してしまう率 (false positive) が低い
- 各識別器で顔でないと判断された場合、即座に顔でないと決断



# Attentional cascadeの学習

- 各ステージに目標検出率と目標false positive率を設定
- 目標検出率、目標false positive率が達成されるまで現在のステージに特徴を追加
  - AdaBoost閾値を下げる必要性
- 総合的なfalse positive率が十分低くない場合、別のステージを追加
- 次のステージのnegative訓練サンプルとして、現在のステージでfalse positiveと判定されたデータを利用



# 訓練データと学習

- 訓練データ
  - 5000枚顔画像
  - 顔画像は正規化
    - スケール, 移動
  - 顔画像には多様性
    - 複数人, 照明, 姿勢
- 数週間の訓練時間
  - 466 MHz Sun workstation
- 38 レイヤー, 6061 features
- 検出時間 : 15Hz
  - 700 Mhz Pentium III
  - 384 x 288 pixel



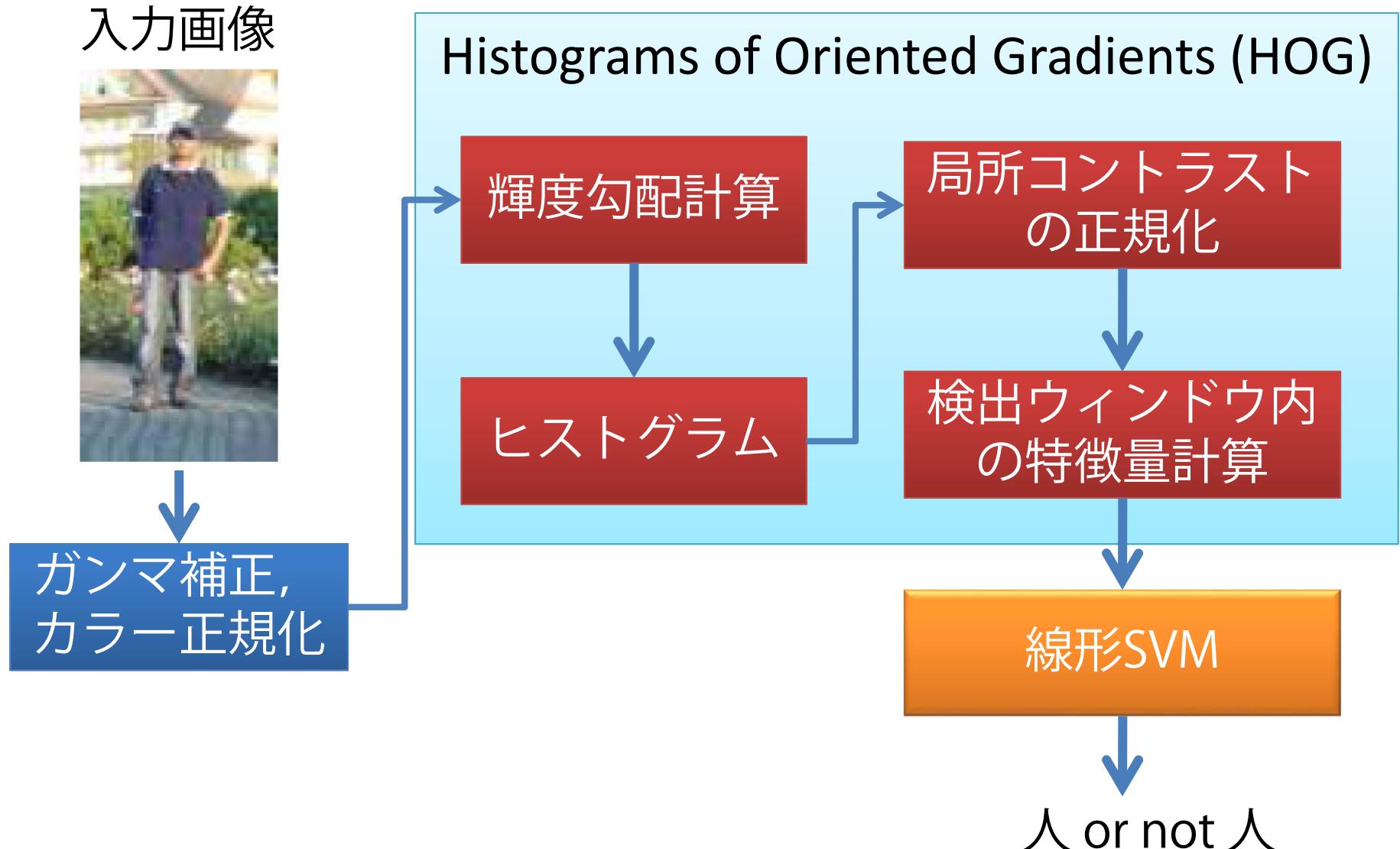
# Viola Jones 顔検出まとめ

- Rectangle features
- 高速な特徴評価のための積分画像
- 特徴抽出のためのBoosting
- 非顔領域を高速に排除するためのAttentional cascade

# 人検出

- Navneet Dalal and Bill Triggs. Histograms of Oriented Gradients for Human Detection. In CVPR, 2005.
- HOG特徴+SVM (Support Vector Machine)
- HOG (Histograms of Oriented Gradients)
  - SIFTやShape Contextに類似
  - 等間隔かつ密なグリッド上での輝度勾配計算
  - オーバーラップした局所コントラスト正規化
- OpenCV2.0に実装済み

# Dalalらの人検出の流れ



# 人の検出 (sliding window法)

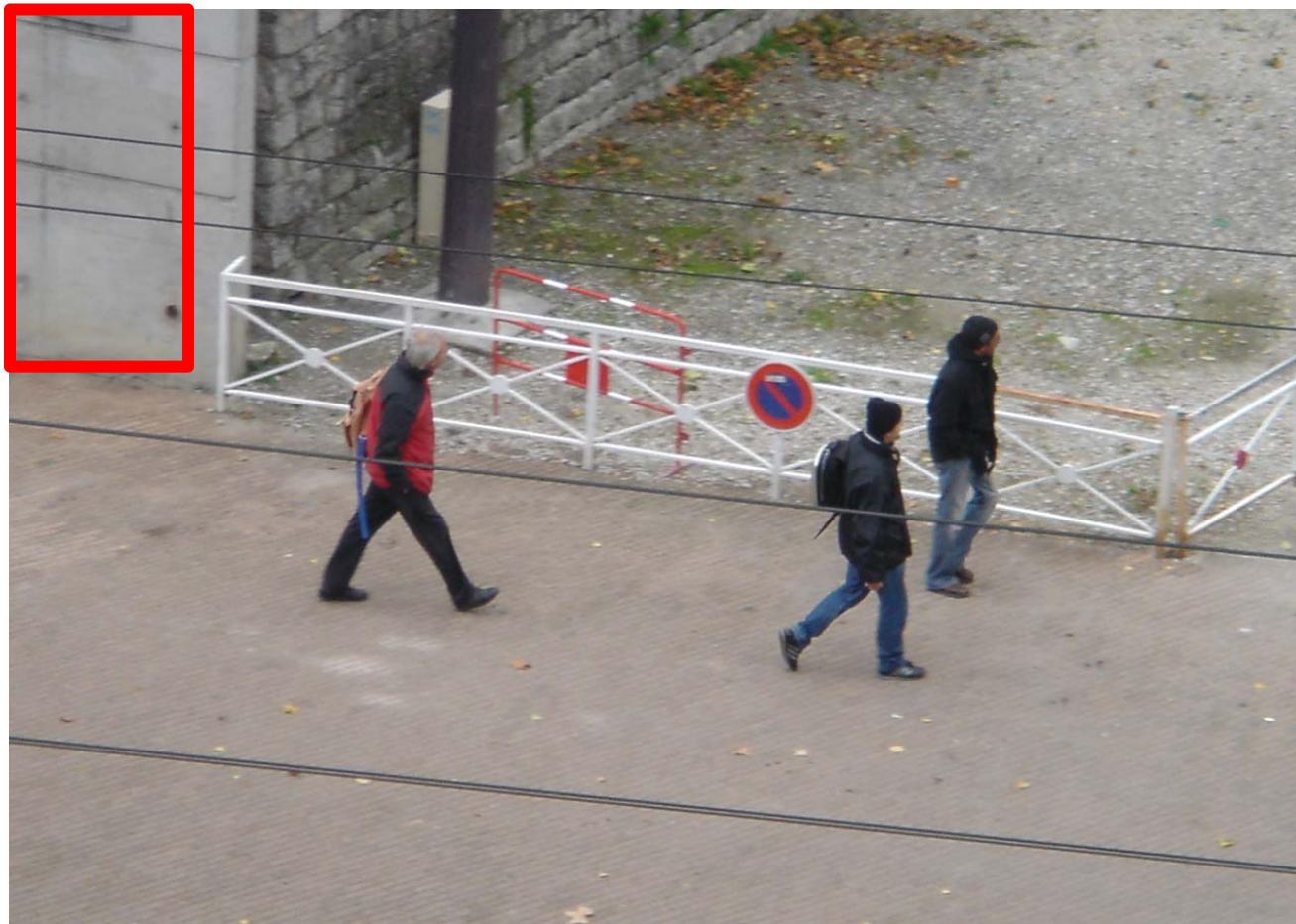
位置 $x$ における  
特徴量

$$\mathbf{f}(\mathbf{x})$$

↓  
特徴量を識別機  
 $y=g(\cdot)$ に入力

$$y = g(\mathbf{f}(\mathbf{x}))$$

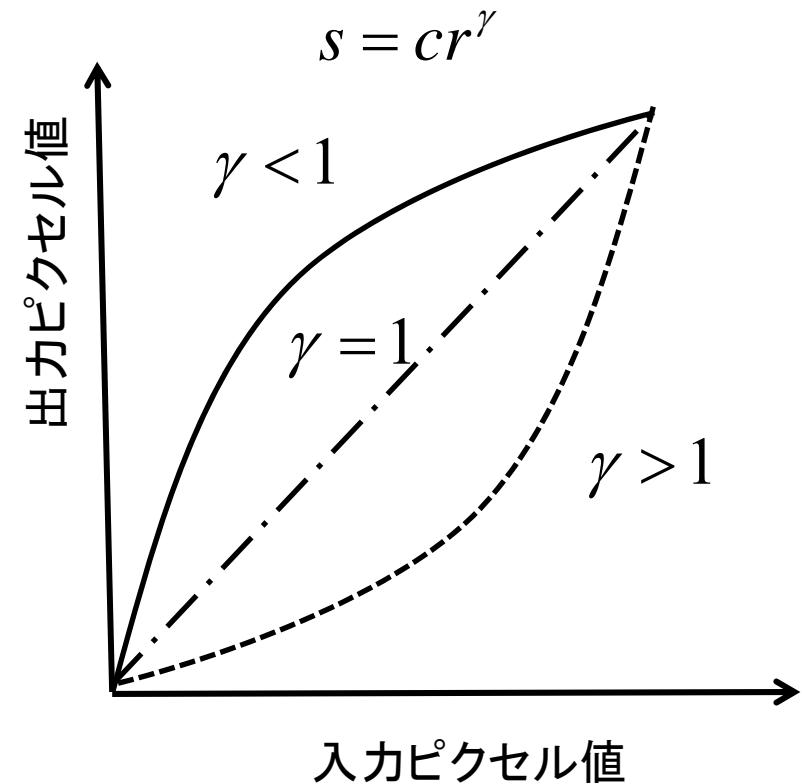
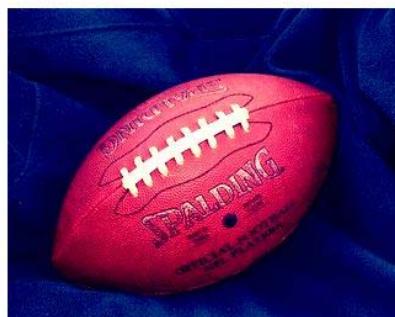
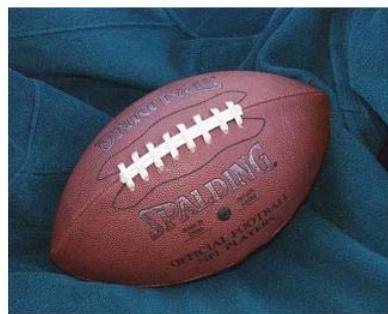
↓  
人 or not 人



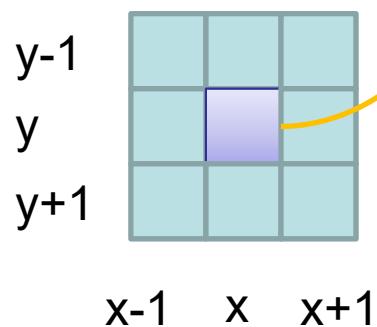
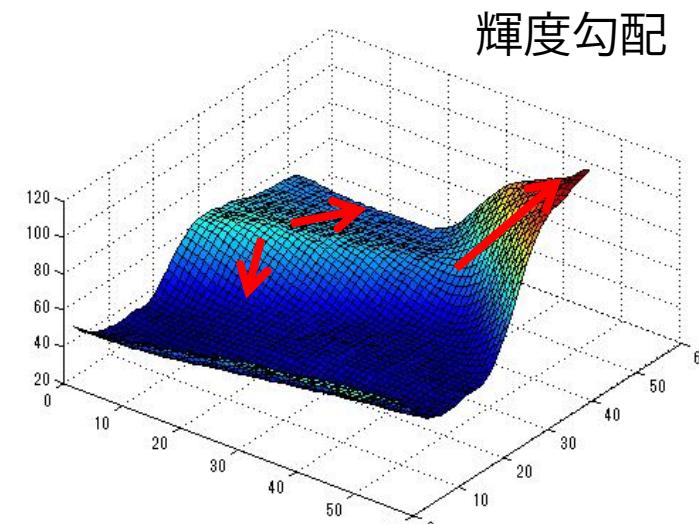
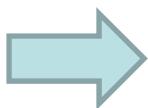
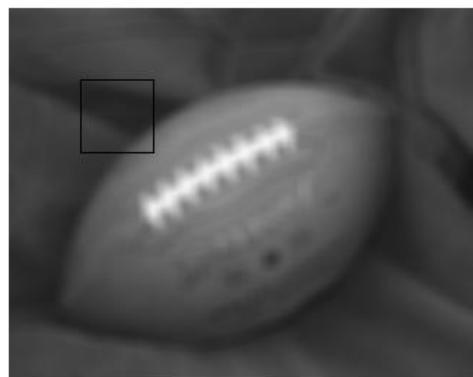
- 線形SVM
  - ガウシアンカーネルのSVMはほんの少し性能が高  
いが、計算コストが非常に高い

# ガンマ補正, カラー正規化

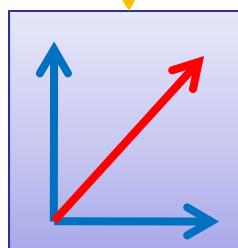
- 輝度勾配計算前のガンマ補正, カラー正規化は人検出性能にあまり影響を与えない。
  - 後で行う局所輝度勾配ヒストグラムの正規化が利いている



# 勾配の計算



$f_y$



輝度勾配

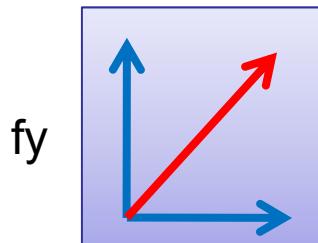
局所輝度勾配の計算

$$f_x = I(x+1, y) - I(x-1, y)$$
$$f_y = I(x, y+1) - I(x, y-1)$$

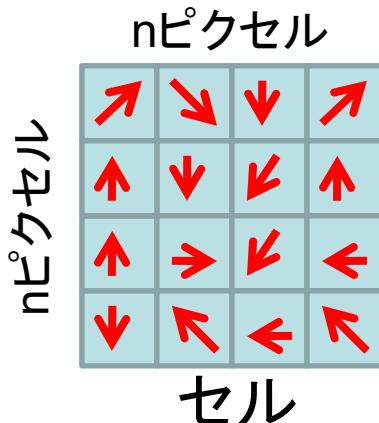
- 各ピクセル毎に勾配を計算する。
- 勾配計算前にガウシアンカーネルの平滑化を試したが性能は良くなかった。
- カラー画像の場合は各カラーチャンネルで勾配を計算する。

# 勾配のヒストグラム化

1. 全てのピクセルで輝度勾配を計算

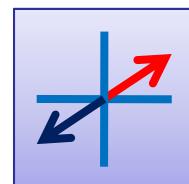


3.  $n \times n$  ピクセルをひとまとまりとする → セル

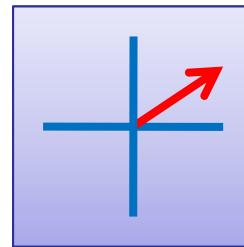


セル：1次元勾配ヒストグラムを計算する単位

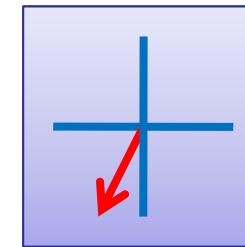
- 符号無し ( $0^\circ \sim 180^\circ$ ) ,  $20^\circ$  每のヒストグラムが性能が高い。
- つまり1セルあたり9次元のベクトルで表現される。
- Dalalらの論文では $8 \times 8$  ピクセルを1セルとしている。



2. 勾配方向の量子化

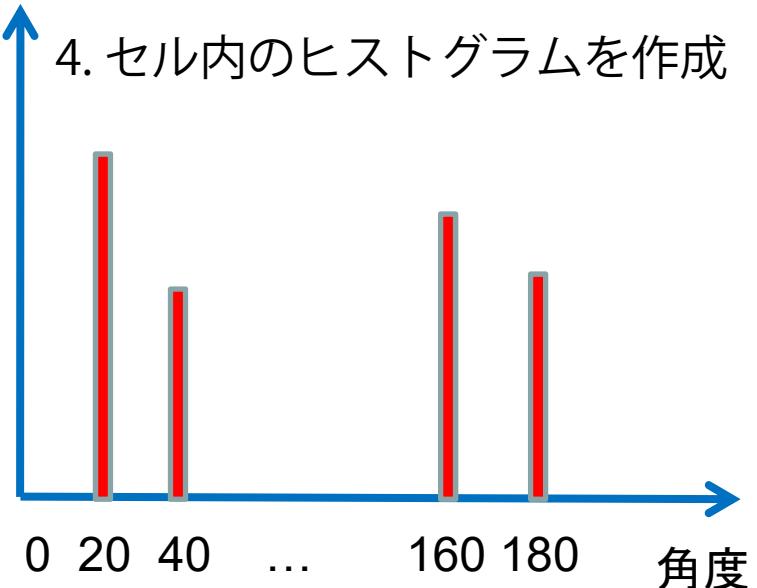


$37^\circ \rightarrow 40^\circ$



$256^\circ \rightarrow 260^\circ$

4. セル内のヒストグラムを作成



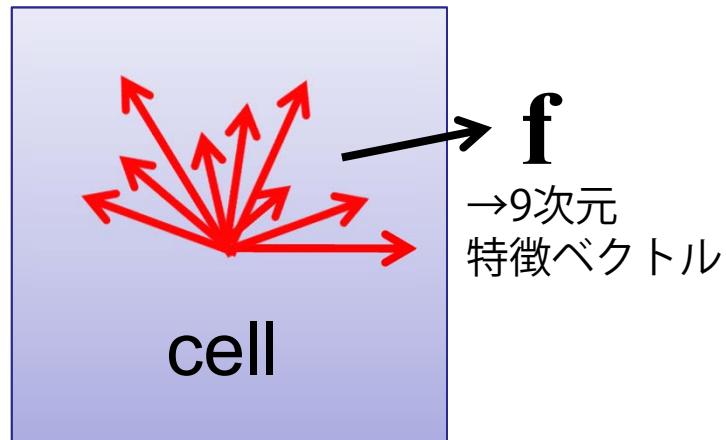
5. 特徴ベクトル

$\mathbf{f} \rightarrow 9$  次元

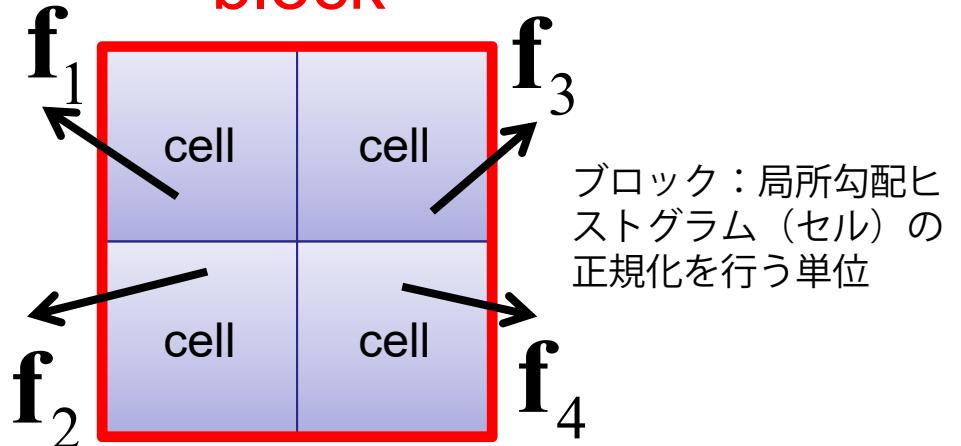
勾配を同じとみなす：符号なし  
勾配を異なるとみなす：符号あり

# 正規化と記述子ブロック

1. 全てのセルの局所勾配ヒストグラムを作成



2. 複数の隣接するセルをまとめてブロック化  
**block**



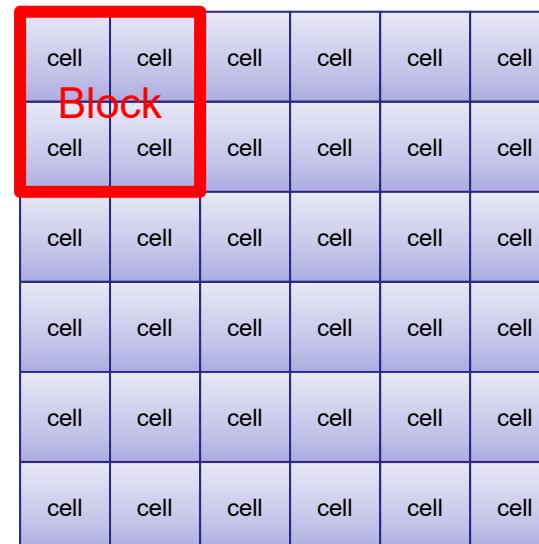
3. ブロックの特徴ベクトル

$$\mathbf{v} = (\mathbf{f}_1^T \ \mathbf{f}_2^T \ \mathbf{f}_3^T \ \mathbf{f}_4^T)^T$$

4. ブロック内での特徴ベクトルの正規化

$$\mathbf{v} = \frac{\mathbf{v}}{\sqrt{\|\mathbf{v}\|_2^2 + \epsilon^2}}$$

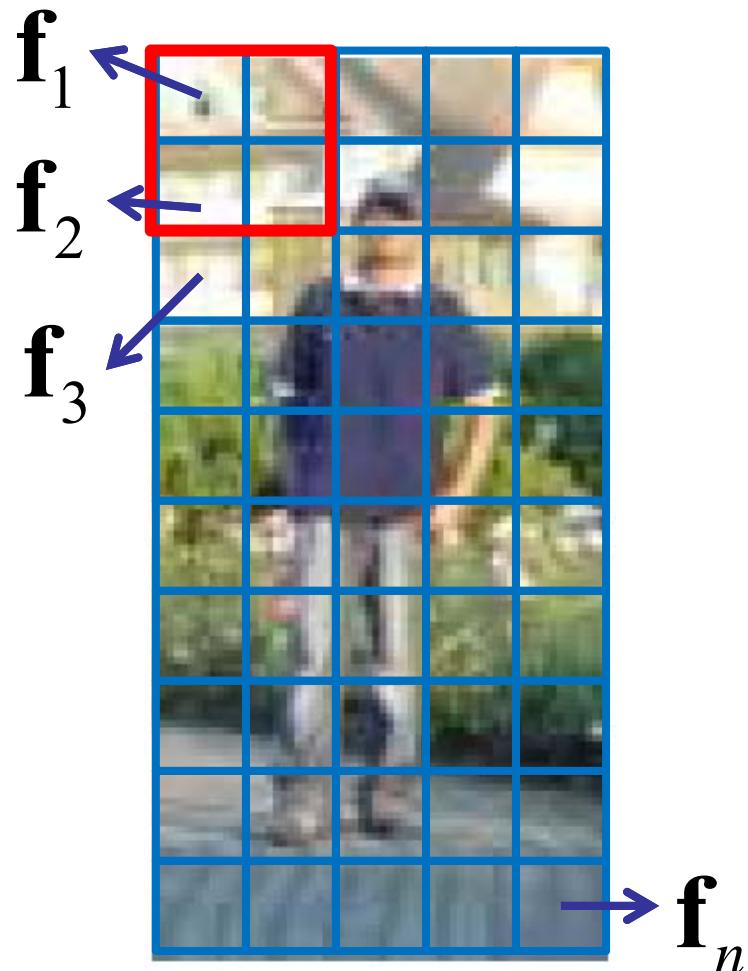
5. ブロックを移動させて正規化



つまり各セルは何度も正規化されることになる。

冗長だが性能が良かった。

# 検出器のウィンドウ



- ・ ウィンドウ内をセルに分割
- ・ 1セルは9次元のベクトル
  - 局所輝度勾配ヒストグラム
- ・ ブロック毎の正規化
- ・ セルから得られるベクトルを全て連結して一つの特徴ベクトルとする

$$\mathbf{f} = (\mathbf{f}_1^T \ \mathbf{f}_2^T \ \mathbf{f}_3^T \cdots \ \mathbf{f}_n^T)^T$$

# Dalalらの人検出の流れ

入力画像



↓  
ガンマ補正,  
カラー正規化

Histograms of Oriented Gradients (HOG)

輝度勾配計算

ヒストグラム

局所コントラスト  
の正規化

検出ウィンドウ内  
の特徴量計算

線形SVM

人 or not 人

# 人検出 SVMの学習結果

- SVMの学習結果

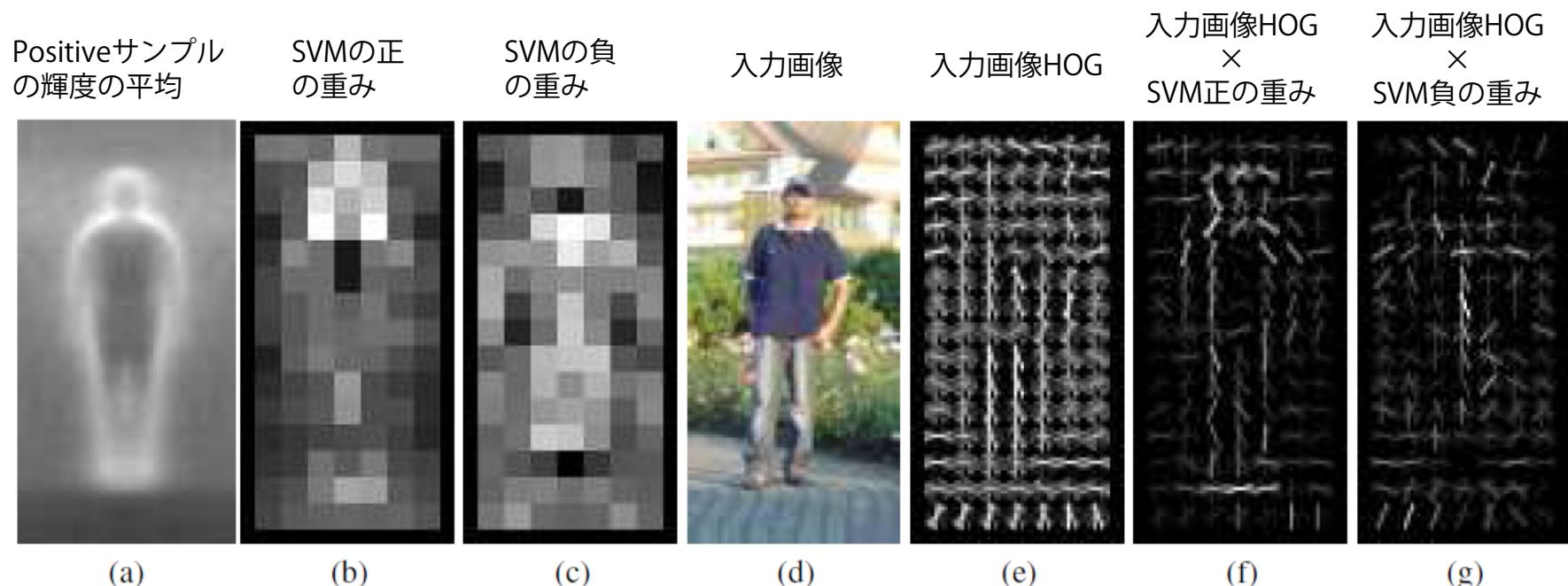
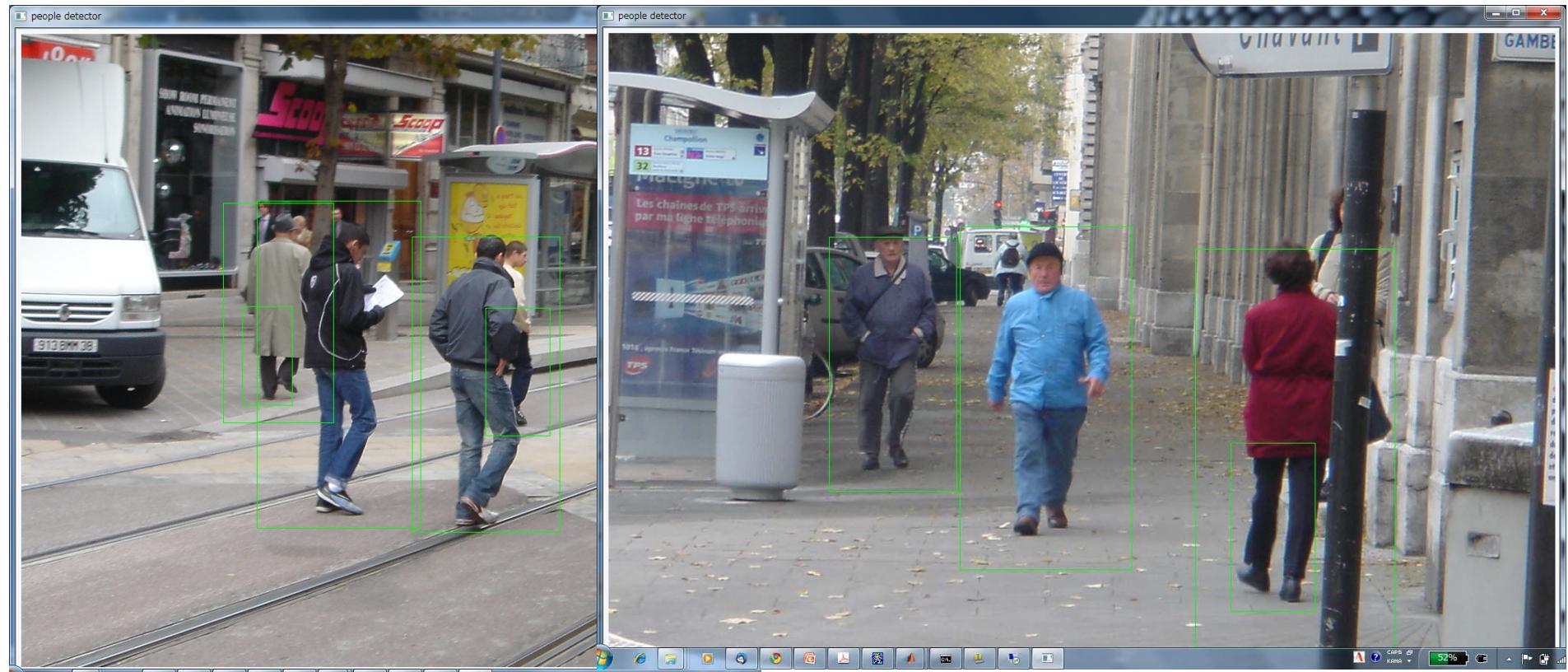


Figure 6. Our HOG detectors cue mainly on silhouette contours (especially the head, shoulders and feet). The most active blocks are centred on the image background just *outside* the contour. (a) The average gradient image over the training examples. (b) Each “pixel” shows the maximum positive SVM weight in the block centred on the pixel. (c) Likewise for the negative SVM weights. (d) A test image. (e) It’s computed R-HOG descriptor. (f,g) The R-HOG descriptor weighted by respectively the positive and the negative SVM weights.

# Dalalらの手法による人検出結果



# Dalalらの人検出まとめ

- Navneet Dalal and Bill Triggs. Histograms of Oriented Gradients for Human Detection. In CVPR, 2005.
- HOG特徴+SVM (Support Vector Machine)
- HOG (Histograms of Oriented Gradients)
  - SIFTやShape Contextに類似
  - 等間隔かつ密なグリッド上での輝度勾配計算
  - オーバーラップした局所コントラスト正規化
- OpenCV2.0に実装済み
- False Positiveが多い！

# Deformable Parts Models (DPM)

- 一般的物体検出のデファクトスタンダード
- P. Felzenszwalb, D. McAllester, D. Ramaman.  
A Discriminatively Trained, Multiscale,  
Deformable Part Model. CVPR 2008.
- P. Felzenszwalb, R. Girshick, D. McAllester, D.  
Ramanan. Object Detection with  
Discriminatively Trained Part Based Models.  
IEEE Transactions on Pattern Analysis and  
Machine Intelligence, Vol. 32, No. 9, 2010.

# DPMの概要

- Dalalらの手法に近い.
  - HOG+SVM
- 異なる点
  - 変形可能な物体モデルを扱える.
  - 隠れ変数を扱えるSVM手法の提案.
    - 物体の変形具合を隠れ変数と見なす.
  - 識別が困難な負例を積極的に利用して学習する手法の提案.
    - 負例はほぼ無限に存在するので、困難な負例のみを利用する.

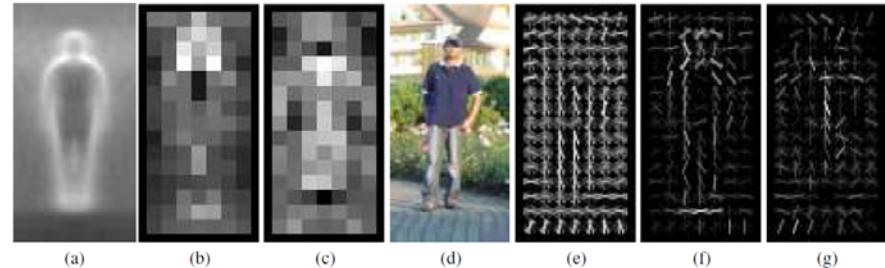
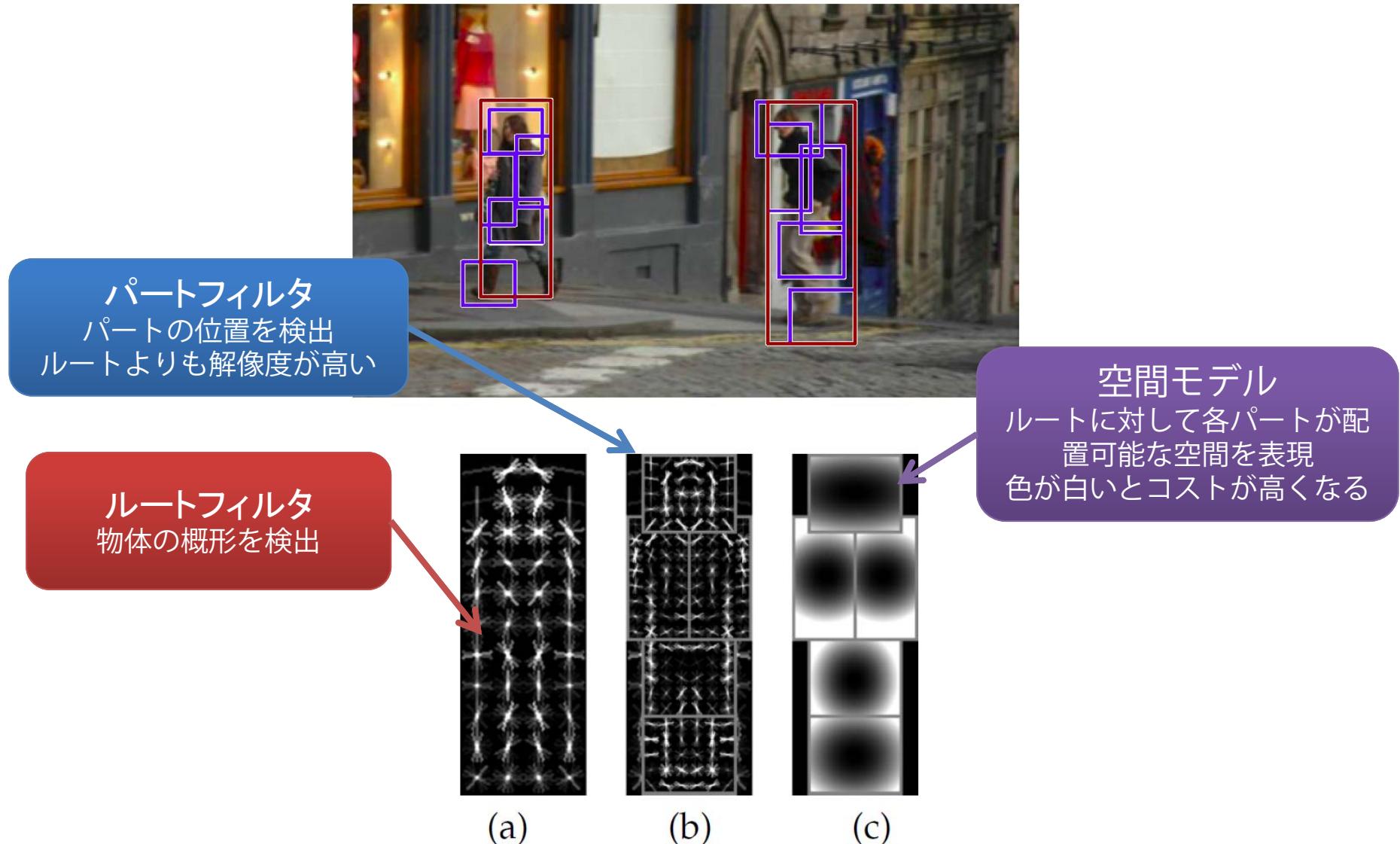


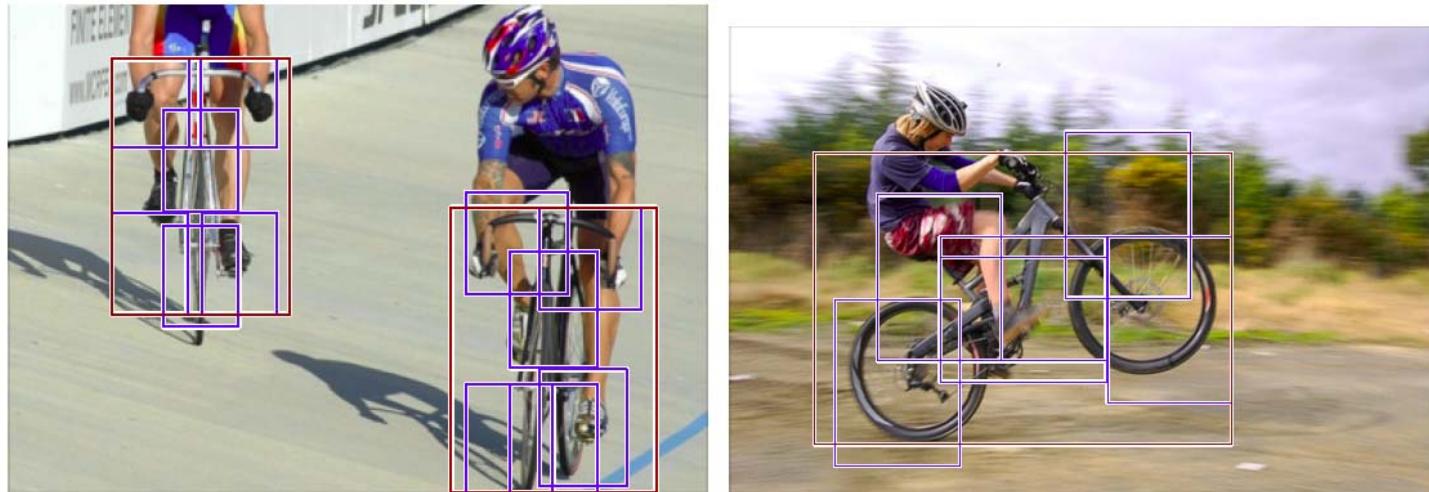
Figure 6. Our HOG detectors cue mainly on silhouette contours (especially the head, shoulders and feet). The most active blocks are centred on the image background just *outside* the contour. (a) The average gradient image over the training examples. (b) Each “pixel” shows the maximum positive SVM weight in the block centred on the pixel. (c) Likewise for the negative SVM weights. (d) A test image. (e) It’s computed R-HOG descriptor. (f,g) The R-HOG descriptor weighted by respectively the positive and the negative SVM weights.

# モデル

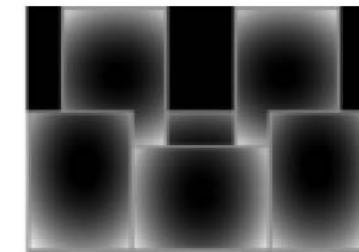
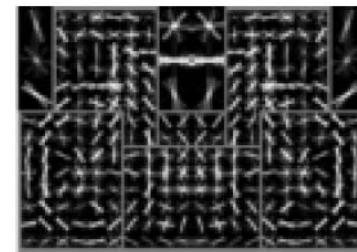
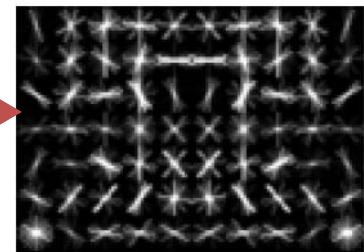
- ・ 変形を許容する柔軟なモデル
- ・ モデル=ルート+パート+変形コストのマップ



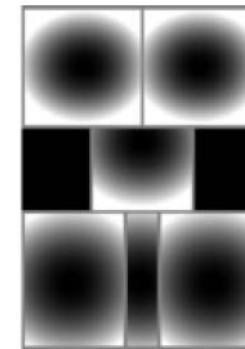
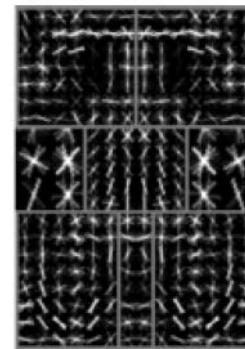
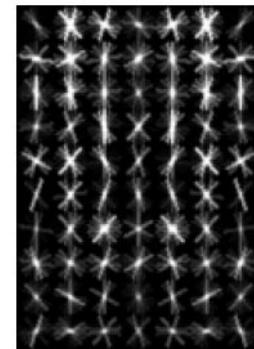
# 混合モデル



自転車を横から  
見たモデル

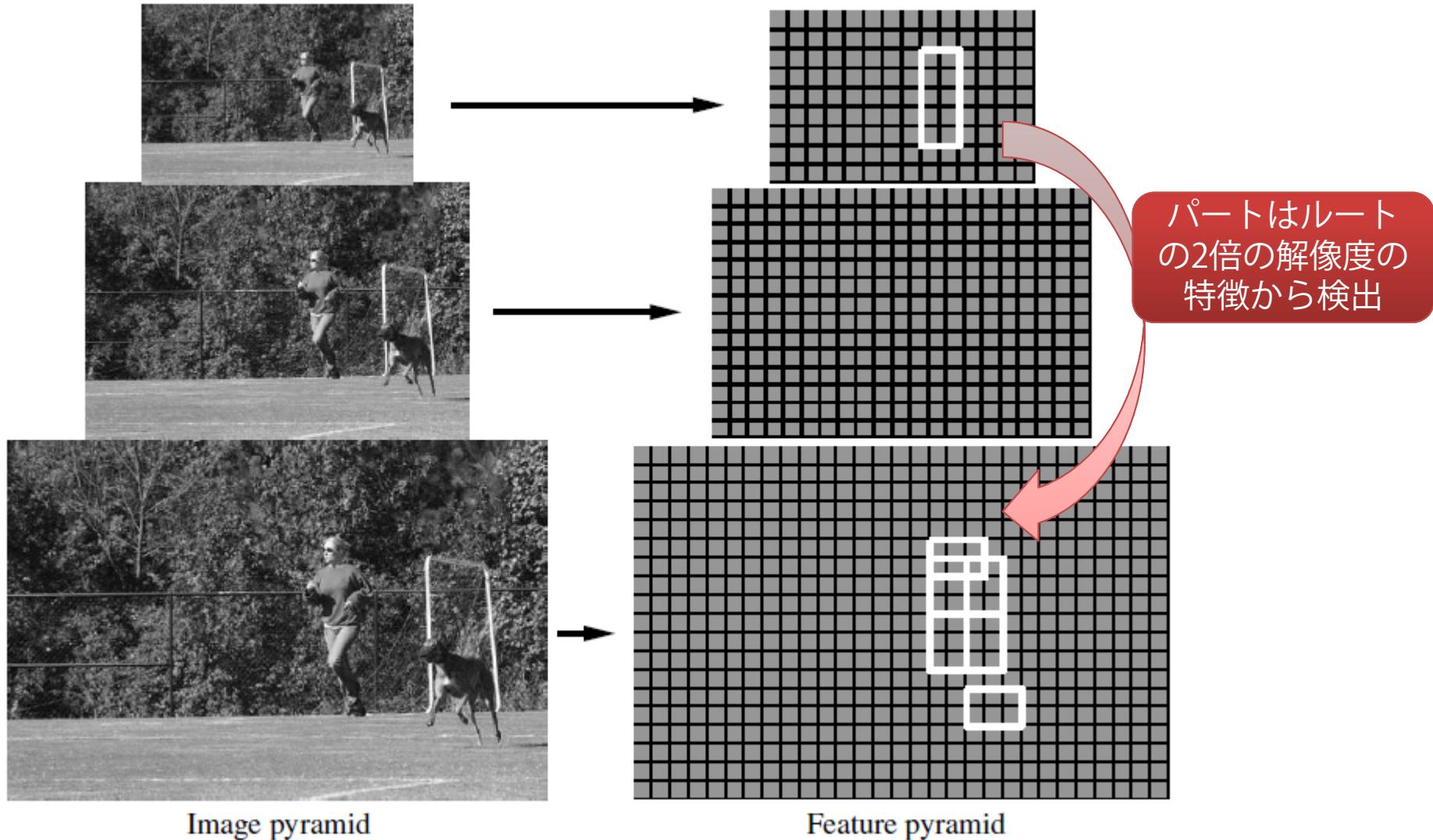


自転車を正面から  
見たモデル



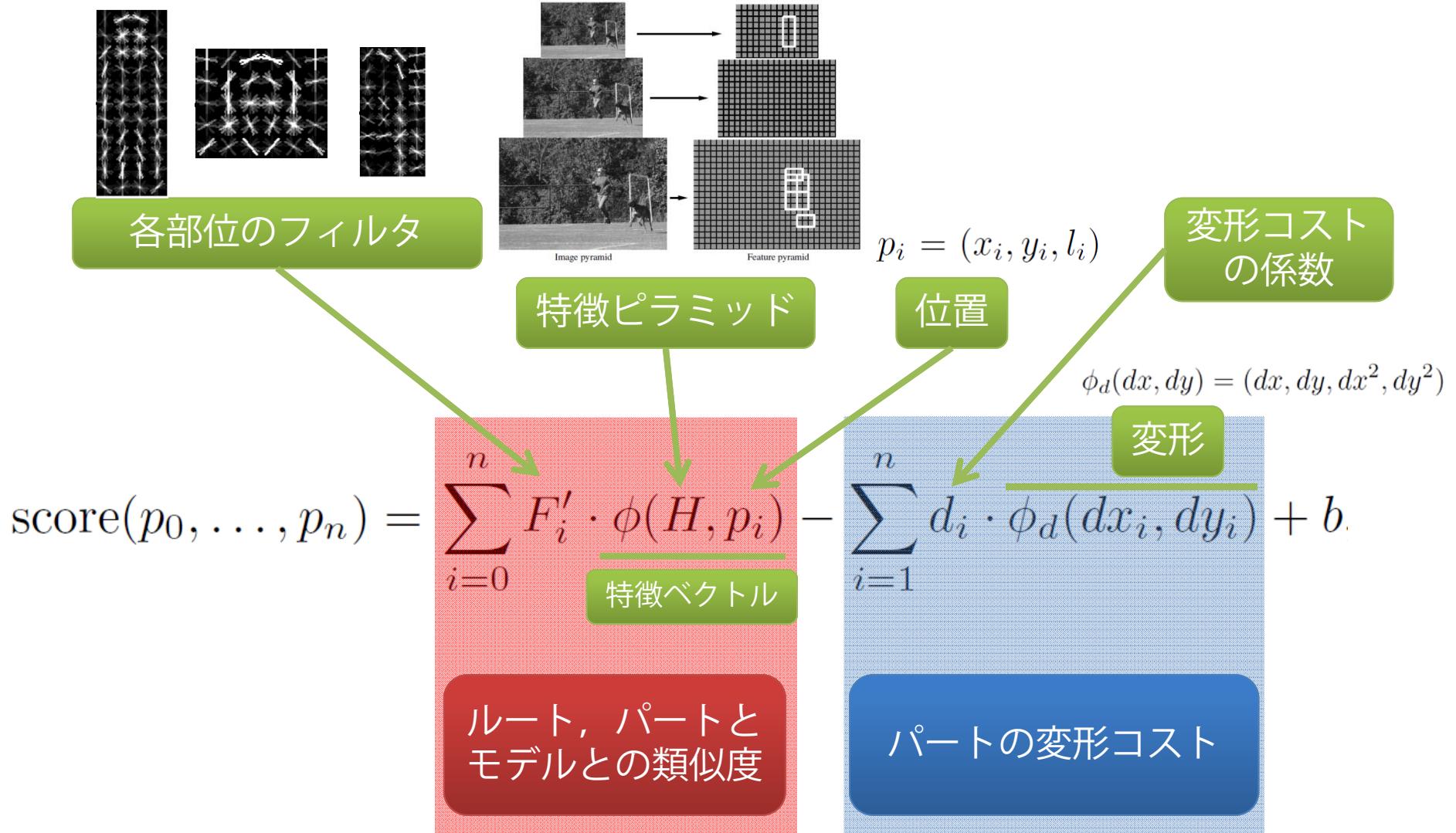
# 特徴ピラミッド

- 様々な位置とスケールに対応するために特徴ピラミッドを利用



# スコア

- スコア=各パートの類似度-変形コスト



# スコア

- スコア関数は要するに線形モデル

$$\text{score}(p_0, \dots, p_n) = \sum_{i=0}^n F'_i \cdot \phi(H, p_i) - \sum_{i=1}^n d_i \cdot \phi_d(dx_i, dy_i) + b$$



$$\boxed{\beta \cdot \psi(H, z)}$$

$$\beta = (F'_0, \dots, F'_n, d_1, \dots, d_n, b)$$

$$\begin{aligned}\psi(H, z) = & (\phi(H, p_0), \dots, \phi(H, p_n), \\ & -\phi_d(dx_1, dy_1), \dots, -\phi_d(dx_n, dy_n), 1)\end{aligned}$$

# マッチング

- 可能性のあるパートの位置を動かしてスコアが最大となるルート位置を求める

$$\text{score}(p_0) = \max_{p_1, \dots, p_n} \text{score}(p_0, \dots, p_n)$$

- パート*i*のレベル*l*における類似度を2次元のマトリクス*R*に保持する

$$R_{i,l}(x, y) = F'_i \cdot \phi(H, (x, y, l))$$

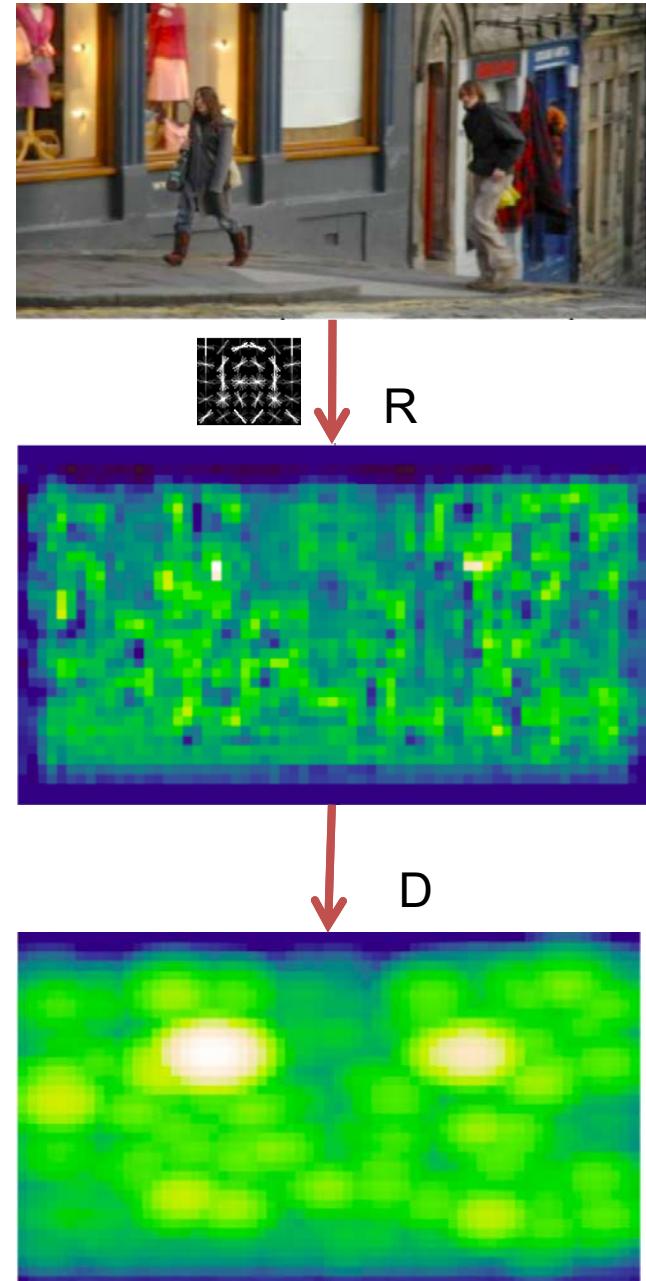
- R*に変形コストを考慮する。これは高いフィルタの値を近傍の領域に伝播させる効果がある。

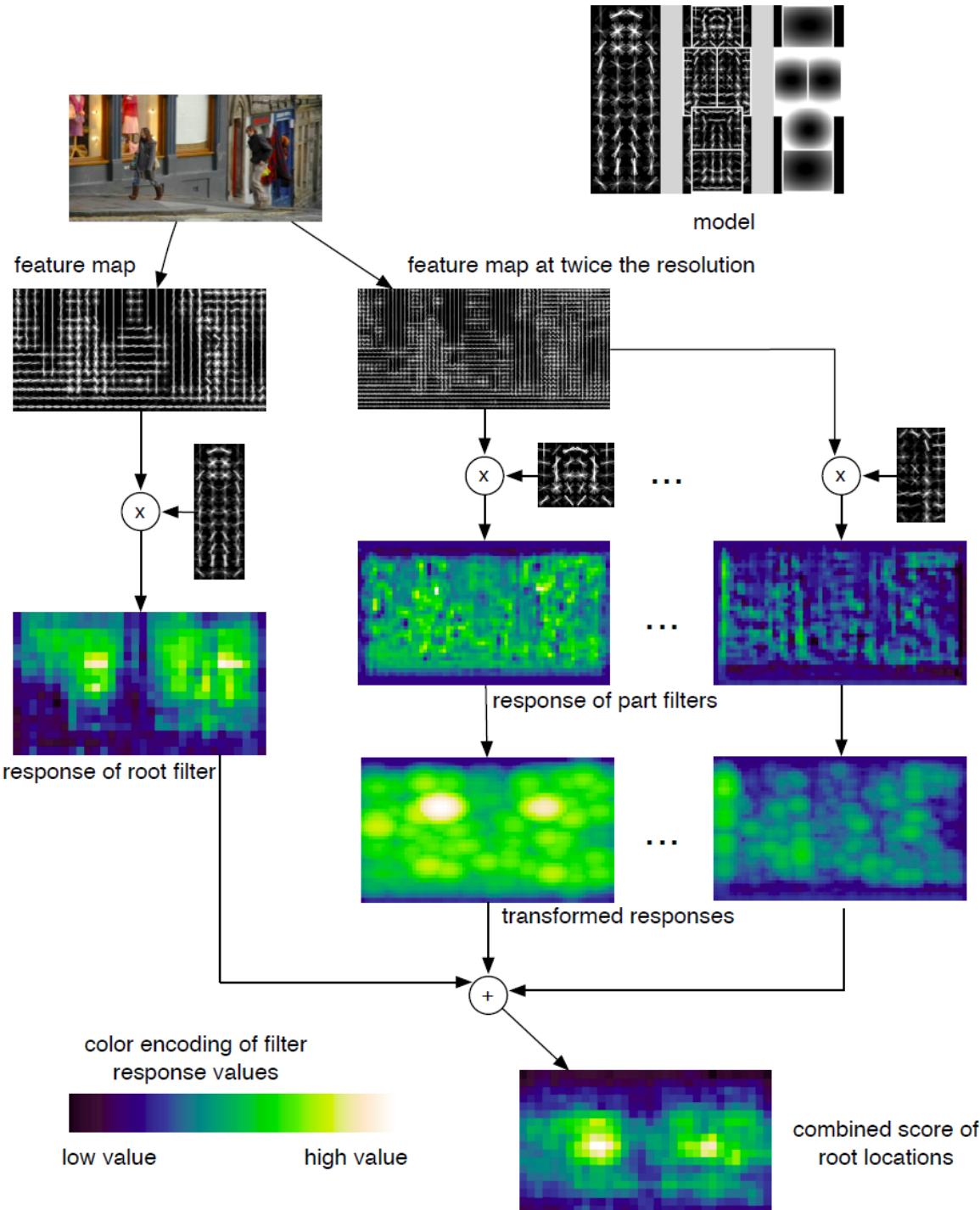
$$D_{i,l}(x, y) = \max_{dx, dy} (R_{i,l}(x + dx, y + dy) - d_i \cdot \phi_d(dx, dy))$$

- 各部位の和を計算してスコアとする。

$$\text{score}(x_0, y_0, l_0) =$$

$$R_{0,l_0}(x_0, y_0) + \sum_{i=1}^n D_{i,l_0-\lambda}(2(x_0, y_0) + v_i) + b$$





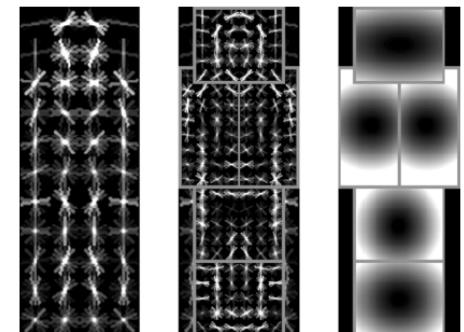
# 学習

- トレーニングデータ
  - ラベルとバウンディングボックスが与えられた画像
  - モデルの構造, フィルタ, 変形コストを学習する必要がある.



バウンディングボックス内では、各パーツの位置は訓練データで明示的に与えられていない

Training →



# Latent SVM

- 識別機

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

- 目的関数

訓練データ

$$D = (\langle x_1, y_1 \rangle, \dots, \langle x_n, y_n \rangle), \text{ where } y_i \in \{-1, 1\}$$

$$L_D(\beta) = \frac{1}{2} \|\beta\|^2 + C \sum_{i=1}^n \max(0, 1 - y_i f_{\beta}(x_i))$$

- 線形SVMはlatent SVMの特殊形

# Latent SVMの学習

- 目的関数の最小化問題

$$L_D(\beta) = \frac{1}{2} \|\beta\|^2 + C \sum_{i=1}^n \max(0, 1 - y_i f_\beta(x_i))$$

- 正例の隠れ変数を固定すると凸関数

$$L_D(\beta) = \min_{Z_p} L_D(\beta, Z_p)$$

- アルゴリズム

- 各正例に対してスコアを最大とする隠れ変数を選択する.
- 隠れ変数を固定して、 $\beta$ を最適化する.

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


# $\beta$ の学習

- 目的関数

$$L_D(\beta) = \frac{1}{2} \|\beta\|^2 + C \sum_{i=1}^n \max(0, 1 - y_i f_\beta(x_i))$$

- 目的関数の微分

$$\nabla L_D(\beta) = \beta + C \sum_{i=1}^n h(\beta, x_i, y_i)$$

$$h(\beta, x_i, y_i) = \begin{cases} 0 & \text{if } y_i f_\beta(x_i) \geq 1 \\ -y_i \Phi(x_i, z_i(\beta)) & \text{otherwise} \end{cases}$$

- Stochastic Gradient Decent

- 1) Let  $\alpha_t$  be the learning rate for iteration  $t$ .
- 2) Let  $i$  be a random example.
- 3) Let  $z_i = \operatorname{argmax}_{z \in Z(x_i)} \beta \cdot \Phi(x_i, z)$ .
- 4) If  $y_i f_\beta(x_i) = y_i (\beta \cdot \Phi(x_i, z_i)) \geq 1$  set  $\beta := \beta - \alpha_t \beta$ .
- 5) Else set  $\beta := \beta - \alpha_t (\beta - C n y_i \Phi(x_i, z_i))$ .

# Hard Negative

- 物体検出ではほとんどの訓練データが負例
- 全ての負例を考えるのではなく、正例 + 困難な負例を考える。
  - $F_t = \text{正例} + \text{困難な負例}$

マージン外

- アルゴリズム

$$E(\beta, F) = \{(i, v) \in F \mid y_i(\beta \cdot v) > 1\}$$

- 1) Let  $\beta_t := \beta^*(F_t)$  (train a model).
- 2) If  $H(\beta, D(Z_p)) \subseteq F_t$  stop and return  $\beta_t$ .
- 3) Let  $F'_t := F_t \setminus X$  for any  $X$  such that  $X \subseteq E(\beta_t, F_t)$  (shrink the cache).
- 4) Let  $F_{t+1} := F'_t \cup X$  for any  $X$  such that  $X \cap H(\beta_t, D(Z_p)) \setminus F_t \neq \emptyset$  (grow the cache).

簡単な集合



困難な集合

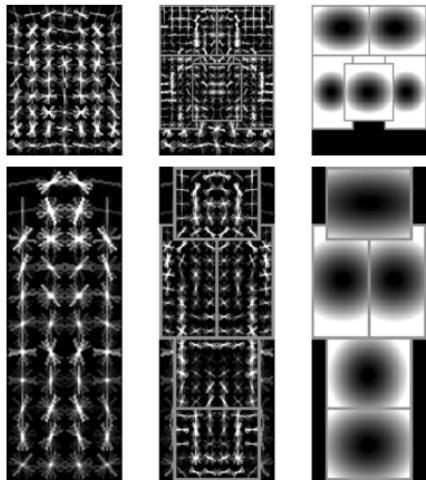
$$H(\beta, D) = \{(i, \Phi(x_i, z_i)) \mid$$

マージン内

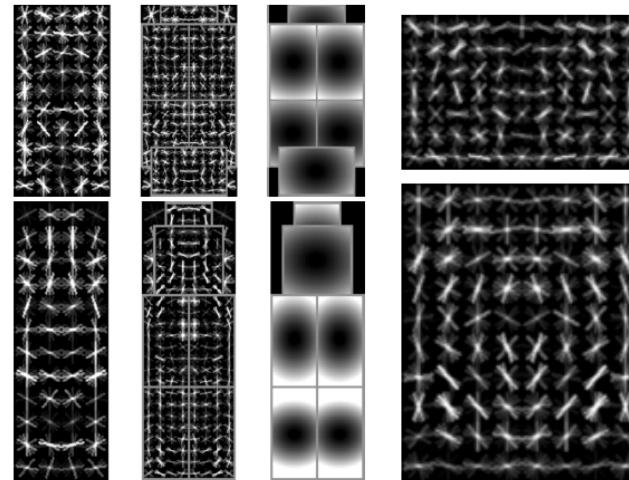
$$z_i = \underset{z \in Z(x_i)}{\operatorname{argmax}} \beta \cdot \Phi(x_i, z) \text{ and } y_i(\beta \cdot \Phi(x_i, z_i)) < 1\}$$

# 学習されたモデル

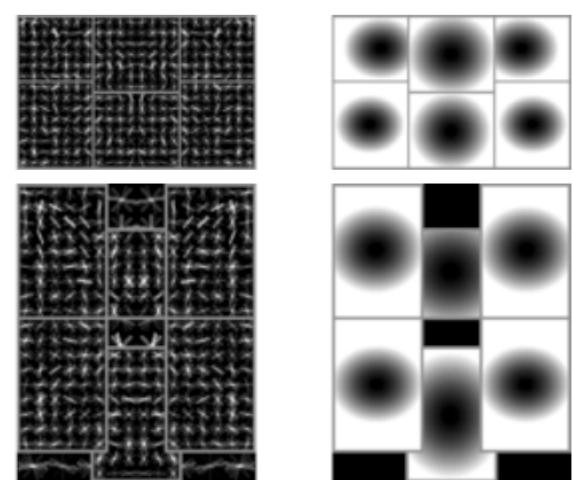
person



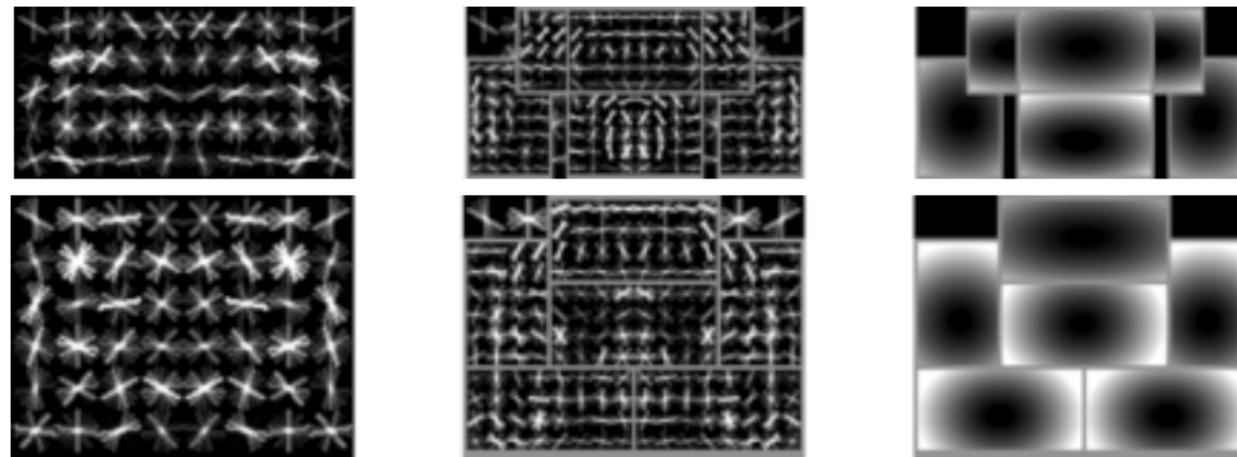
bottle



cat

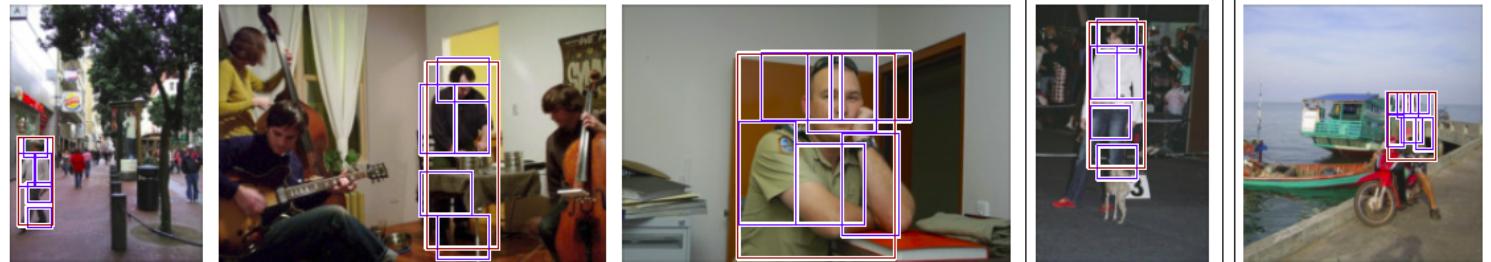


car

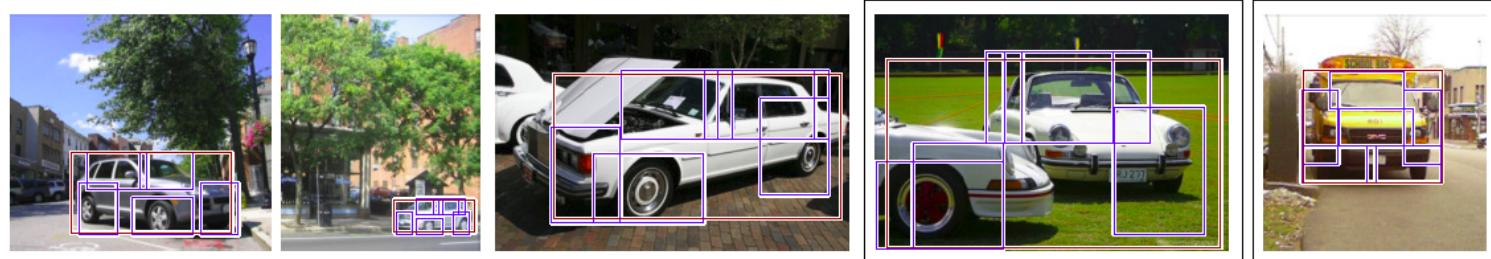


# 検出結果

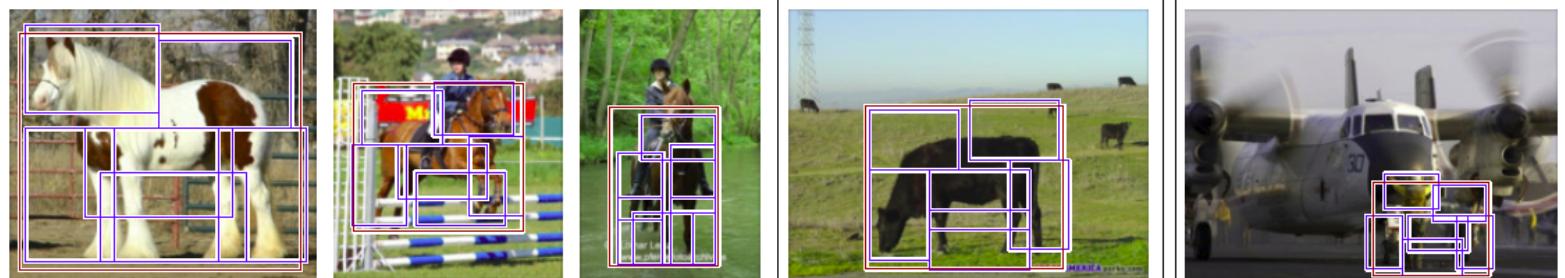
person



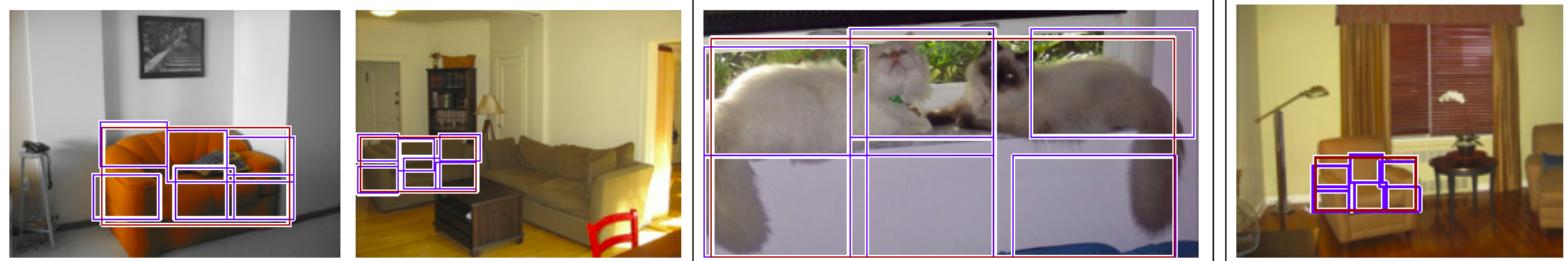
car



horse

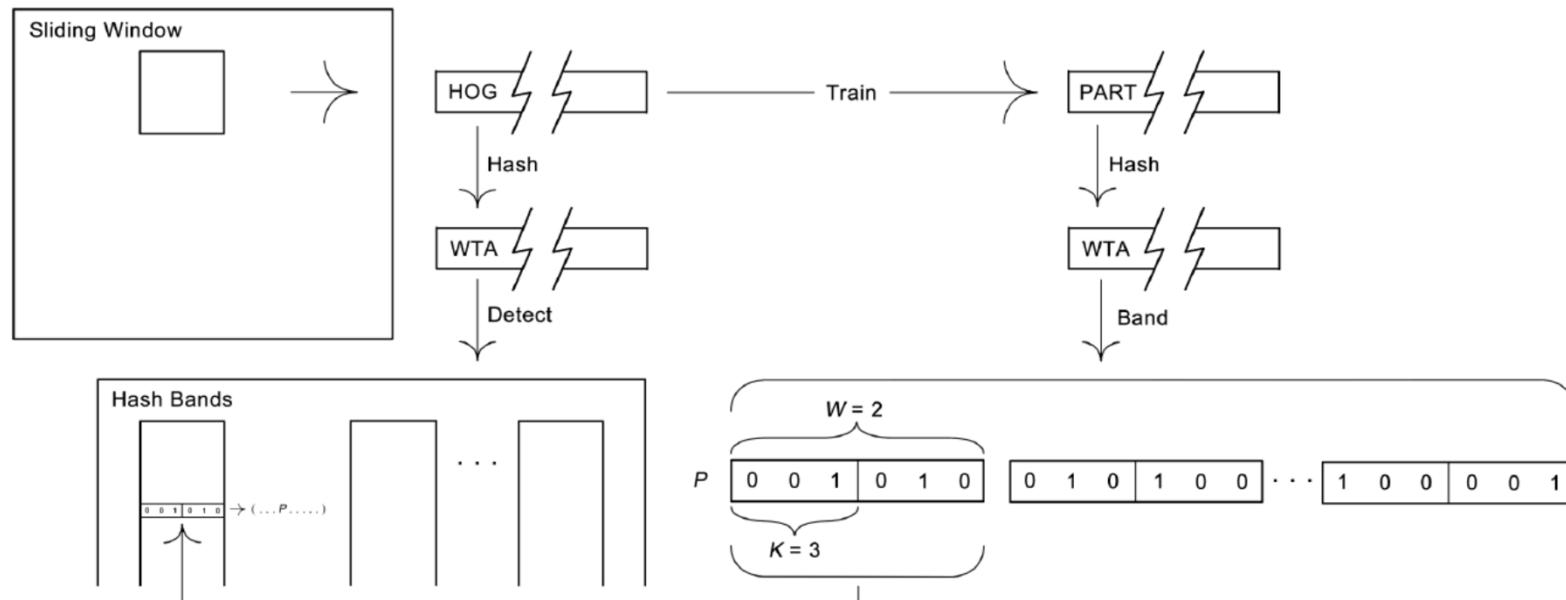


sofa



# 100,000 objects detection on a single machine

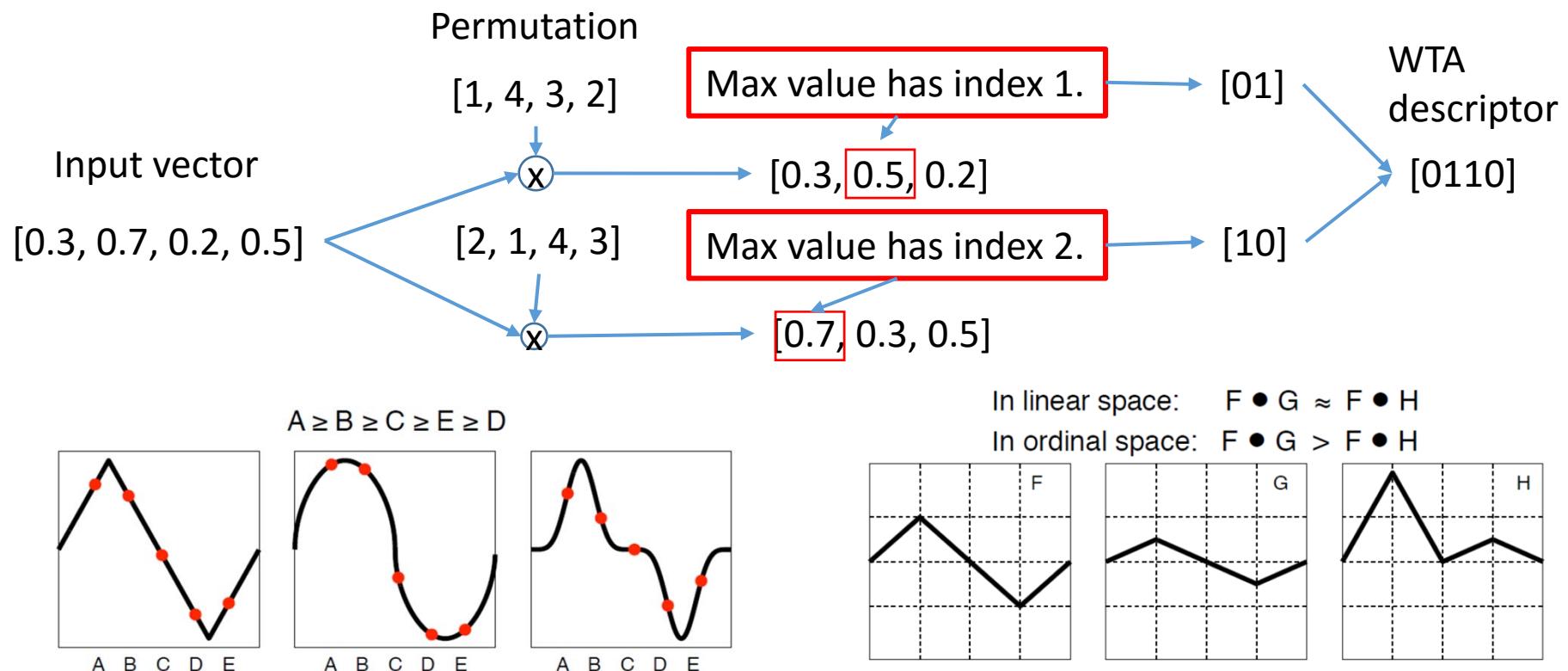
- Thomas Dean, Mark A. Ruzon, Mark Segal, Jonathon Shlens, Sudheendra Vijayanarasimhan, and Jay Yagnik. Fast, Accurate Detection of 100,000 Object Classes on a Single Machine. CVPR, 2013.
- Deformable Parts Models (DPM)を用いて膨大なクラス数のディテクションシステムを作る。
- 膨大なクラスのpart filter群との内積を計算するのは大変。
- Locality sensitive hashingを用いてR個のクラス候補を絞り込み, R個のクラスに対してのみまともなpart filterとの内積計算を行う。
- Locality sensitive hashing: Winner-Take-All Hashingを利用
- 100,000個のクラス検出がシングルマシンで20秒で実行可能。



WTA hashingを利用してR個のクラスを絞り込み, このクラス群との内積計算を行う.

# Winner-Take-All (WTA) Hashing

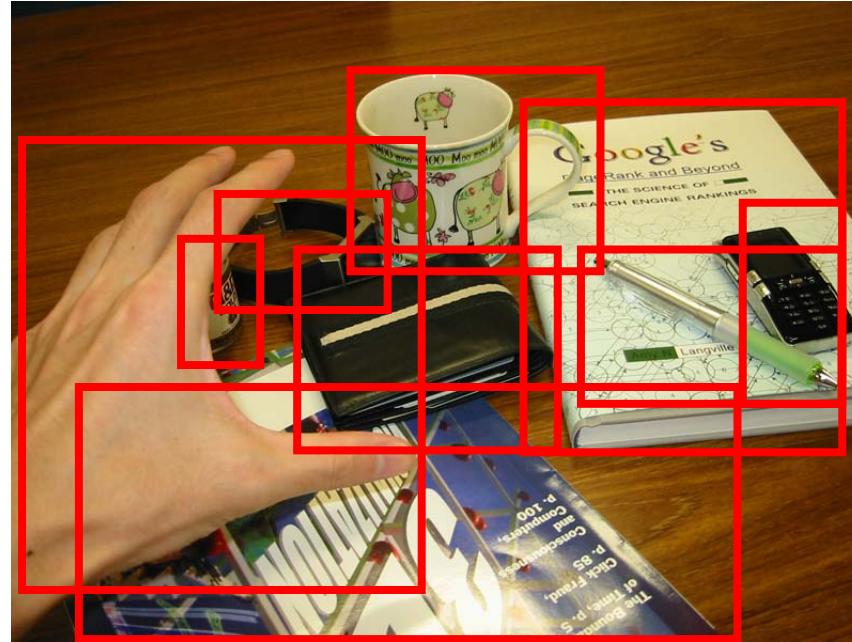
- WTAハッシュ関数
  - 順序埋め込みとランク相関類似度
  - 数値の外乱に関してある程度の不変性を提供



フィルタの変形と外乱に不変

Linear measureではordinal measureで  
捉えられる質的差異を捉えられない

# 物体候補領域群の抽出



- 画像中から物体候補の領域群を抽出するモジュールは、物体検出の精度と速度を決める重要な部分。
- 物体候補領域群を抽出した後は一般的なカテゴリ認識にパイプラインを適用することが多いために、物体候補領域群の抽出は物体検出の本質であるとも言える。
- 物体候補領域群のことを **region proposals** という。

# スライディングウィンドウ法

- ある決まった大きさの小領域を一定のピクセル毎にずらしながら候補領域を抽出する手法
- 幅 $W$ , 高さ $H$ の入力画像
- スライディングウィンドウのグリッド幅:  $d$
- スケールの種類:  $N_s$ , アスペクト比の種類:  $N_a$
- Region proposal数:  $\left\lfloor \frac{W}{d} \right\rfloor \times \left\lfloor \frac{H}{d} \right\rfloor \times N_s \times N_a$
- 確実だが、大量のregion proposalが得られるので、検出時の計算コストが高くなる。



# 分枝限定法 (branch and bound)

Christoph H. Lampert, Matthew B. Blaschko, and Thomas Hofmann. 2009. Efficient Subwindow Search: A Branch and Bound Framework for Object Localization. IEEE Trans. Pattern Anal. Mach. Intell. 31, 12 (December 2009), 2129-2142.

- 画像の中から物体を検出する問題は、物体識別器 $f(\theta)$ のスコアを最大とするサブウィンドウのパラメータ（矩形の上下左右の座標）をパラメータ空間 $\Theta = [[1, H], [1, H], [1, W], [1, W]]$ の中から探し出す問題に帰着される。
- 画像の大きさが $W \times H$ であれば、 $\Theta$ は $O(W^2H^2)$ のオーダーの要素を持つために、全てをしらみつぶしに探索することは難しい。
- 分枝限定法を利用したサブウィンドウ探索手法
  - パラメータ空間を共通する要素のない部分集合に階層的に分割。
  - 各部分集合のスコアの最大値の下限は計算し保持しておく。
  - 有望なパラメータを含みそうな（つまり最大値の下限が大きい）部分集合を優先的に探索し、最大値を含みそうにない他の大部分の部分集合は処理せずに置いておく。
  - この操作を繰り返すことで空間を分割していく、部分集合の要素が一つになれば、その要素が最大値をとるパラメータとなる。

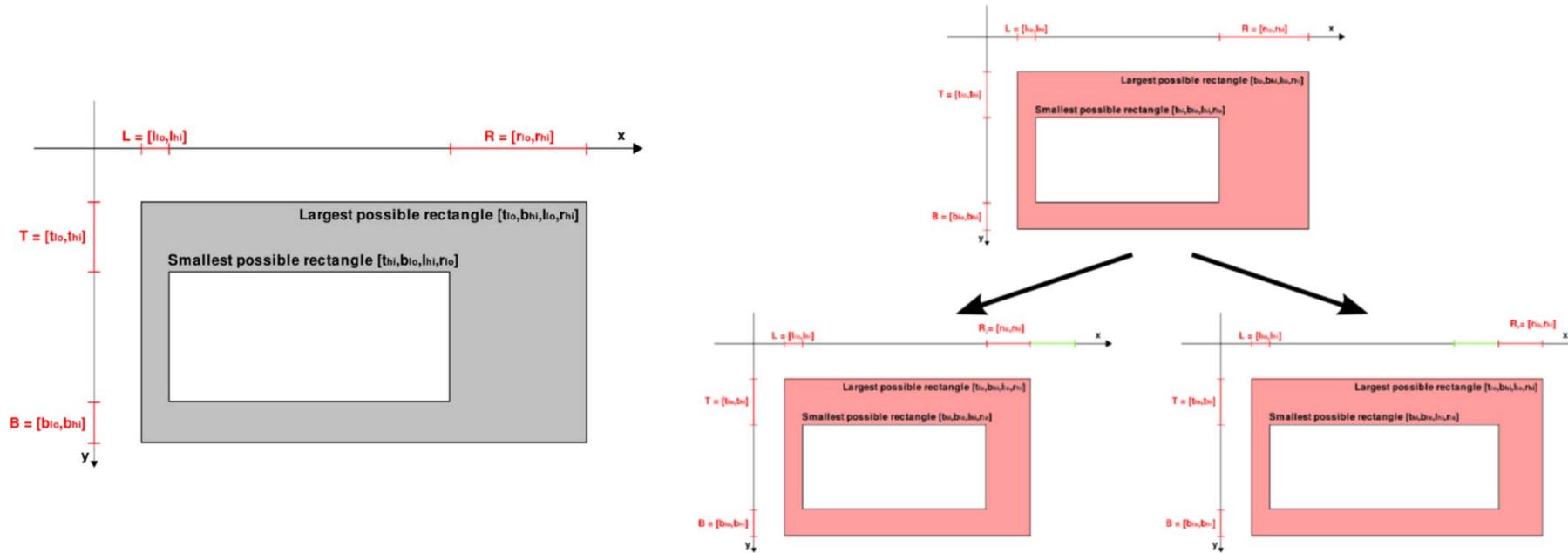


Fig. 2. Splitting rectangle sets is done by dividing one of the intervals in two. In this case,  $[T, B, L, R] \rightarrow [T, B, L, R_1] \dot{\cup} [T, B, L, R_2]$ , where  $R_1 := [r_{lo}, \lfloor \frac{r_{lo}+r_{hi}}{2} \rfloor]$  and  $R_2 := [\lfloor \frac{r_{lo}+r_{hi}}{2} \rfloor + 1, r_{hi}]$ .

# 分枝限定法 (branch and bound)

- 効率的なサブウィンドウ探索

```
input 画像  $I$ , バウンド関数  $\hat{f}$ 
output  $(t_{opt}, b_{opt}, l_{opt}, r_{opt}) = \operatorname{argmax}_{\theta \in \Theta} f(\theta)$ 
initialization
 $P = \emptyset, \Theta = [[1, H], [1, H], [1, W], [1, W]]$ 
repeat
    パラメータ空間を分割  $\Theta \rightarrow \Theta_1 \dot{\cup} \Theta_2$ 
     $P$  に  $(\Theta_1; \hat{f}(\Theta_1))$  と  $(\Theta_2; \hat{f}(\Theta_2))$  をプッシュ
     $P$  から値の最も高いパラメータ集合を  $\Theta$  として取り出す.
until  $\Theta$  の構成要素が 1 つ
return  $(t_{opt}, b_{opt}, l_{opt}, r_{opt}) = \theta \in \Theta$ 
```

関数  $\hat{f}$  は矩形領域の集合に対する識別器  $f$  の出力値の上界の値を計算する関数で、以下の条件を満たす必要がある。

$$\hat{f}(\Theta) \geq \max_{\theta \in \Theta} f(\theta)$$

$$\hat{f}(\Theta) = f(\theta), \text{if } \theta \in \Theta \text{ and } |\Theta| = 1$$

# Selective Search for Object Recognition

- Jasper R. R. Uijlings, Koen E. A. van de Sande, Theo Gevers, Arnold W. M. Smeulders. Selective Search for Object Recognition. International Journal of Computer Vision, Volume 104 (2), page 154-171, 2013

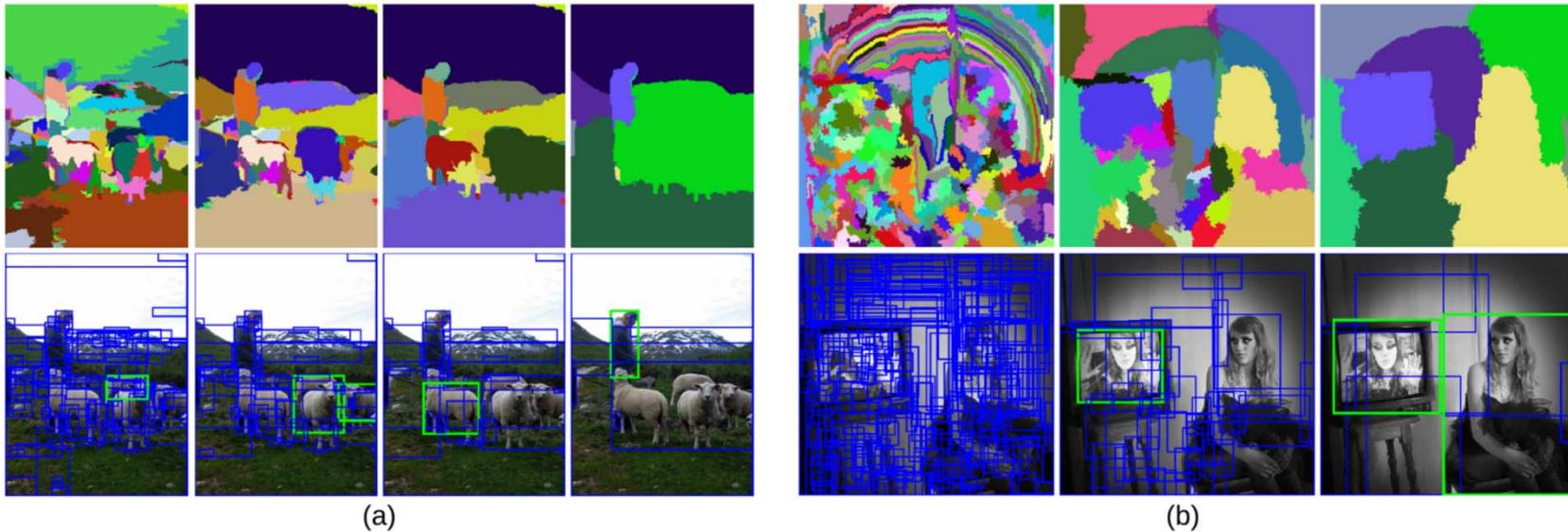


Figure 2: Two examples of our selective search showing the necessity of different scales. On the left we find many objects at different scales. On the right we necessarily find the objects at different scales as the girl is contained by the tv.

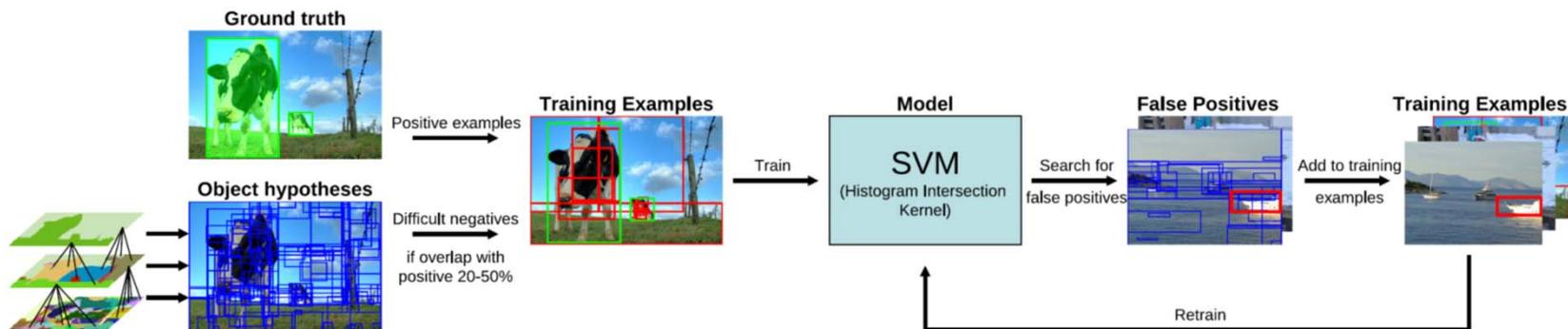


Figure 3: The training procedure of our object recognition pipeline. As positive learning examples we use the ground truth. As negatives we use examples that have a 20-50% overlap with the positive examples. We iteratively add hard negatives using a retraining phase.

# Selective Search for Object Recognition

Jasper R. R. Uijlings, Koen E. A. van de Sande, Theo Gevers, Arnold W. M. Smeulders. Selective Search for Object Recognition. International Journal of Computer Vision, Volume 104 (2), page 154-171, 2013

## 画像セグメンテーション

- 画像を頂点 $V$ とエッジ $E$ で構成されるグラフ構造 $G = (V, E)$ で表現
  - 頂点 $v_i \in V$  は、各画像のピクセル $p_i$
  - エッジ集合はピクセルとその8近傍のピクセル
  - 頂点間のエッジの重み関数はピクセルの輝度の差の絶対値
- エッジの重みの中で最も小さい重みを持つ頂点のペア $p_a$ と $p_b$ を選択。このペアを統合し、新たなセグメンテーション $S^1$ を得る。セグメンテーションされた各領域をコンポーネント $C$ と呼ぶ。頂点 $v_i$ を含むコンポーネントを $C_i$ とする
- 2番目に小さな重みを持つ頂点のペア $p_c$ と $p_d$ を選択。 $w(p_c, p_d)$ が、各コンポーネント $C_c$ と $C_d$ の内部の重みの最大値のうち、小さい重みの値よりも小さければ、 $C_c$ と $C_d$ を統合し、セグメンテーション $S^2$ を得る。もしこの条件を満たさなければ統合せずにセグメンテーション $S^2$ とする。
- この操作を全てのエッジに対して行うことで最終的なセグメンテーション $S = S^m$ を得る。

# Selective Search for Object Recognition

Jasper R. R. Uijlings, Koen E. A. van de Sande, Theo Gevers, Arnold W. M. Smeulders. Selective Search for Object Recognition. International Journal of Computer Vision, Volume 104 (2), page 154-171, 2013

## 階層的グルーピングアルゴリズム

```
input 初期セグメンテーション  $R = (C_1, \dots, C_r)$ 
output 物体候補領域  $L$ 
initialization 初期類似度集合  $S = \emptyset$ 
foreach 隣接する領域ペア  $(C_i, C_j)$  do
    類似度計算  $s(C_i, C_j)$ ,  $S = S \cup s(C_i, C_j)$ 
end foreach
while  $S \neq \emptyset$  do
    最も類似度の高い領域ペアを選択  $s(C_i, C_j) = \max(S)$ 
    領域を統合  $C_t = C_i \cup C_j$ 
     $C_i$  に関する類似度を削除  $S = S \setminus s(C_i, C_*)$ 
     $C_j$  に関する類似度を削除  $S = S \setminus s(C_j, C_*)$ 
     $C_t$  とその近傍領域の類似度集合  $S_t$  を計算
     $S = S \cup S_t$ ,  $R = R \cup r_t$ 
end while
return 全ての領域  $R$  から物体候補領域  $L$  を抽出
```

# Edge Boxes

C. Lawrence Zitnick and Piotr Dollar. Edge Boxes: Locating Object Proposals from Edges. ECCV, 2014.

- ・バウンディングボックスに完全に含まれる輪郭の数が、物体を含むボックスの尤度の指標となる。
- ・物体らしさのスコア
  - ・ボックス内に含まれるエッジの数から、ボックスの境界をまたぐ輪郭のメンバーとなるエッジ数を引いた値。

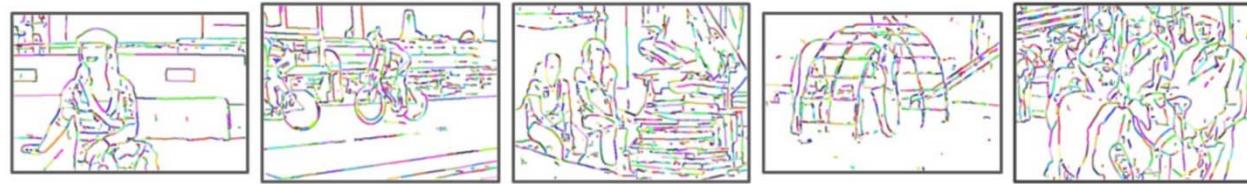
original image



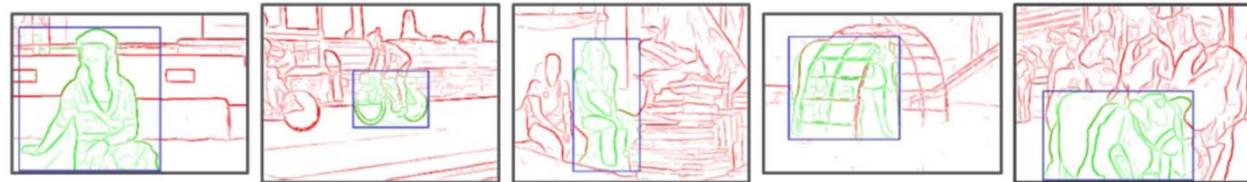
Structured Edges



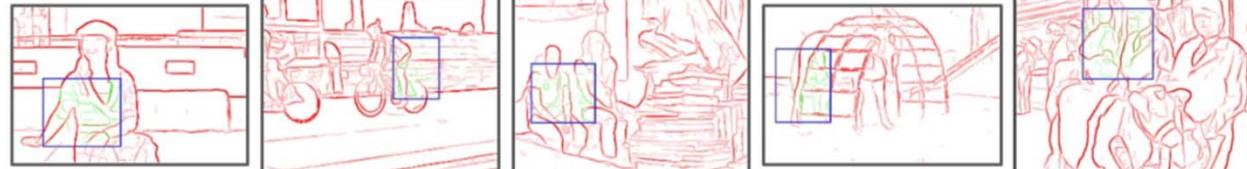
edge groups



Example correct bounding box and edge labeling



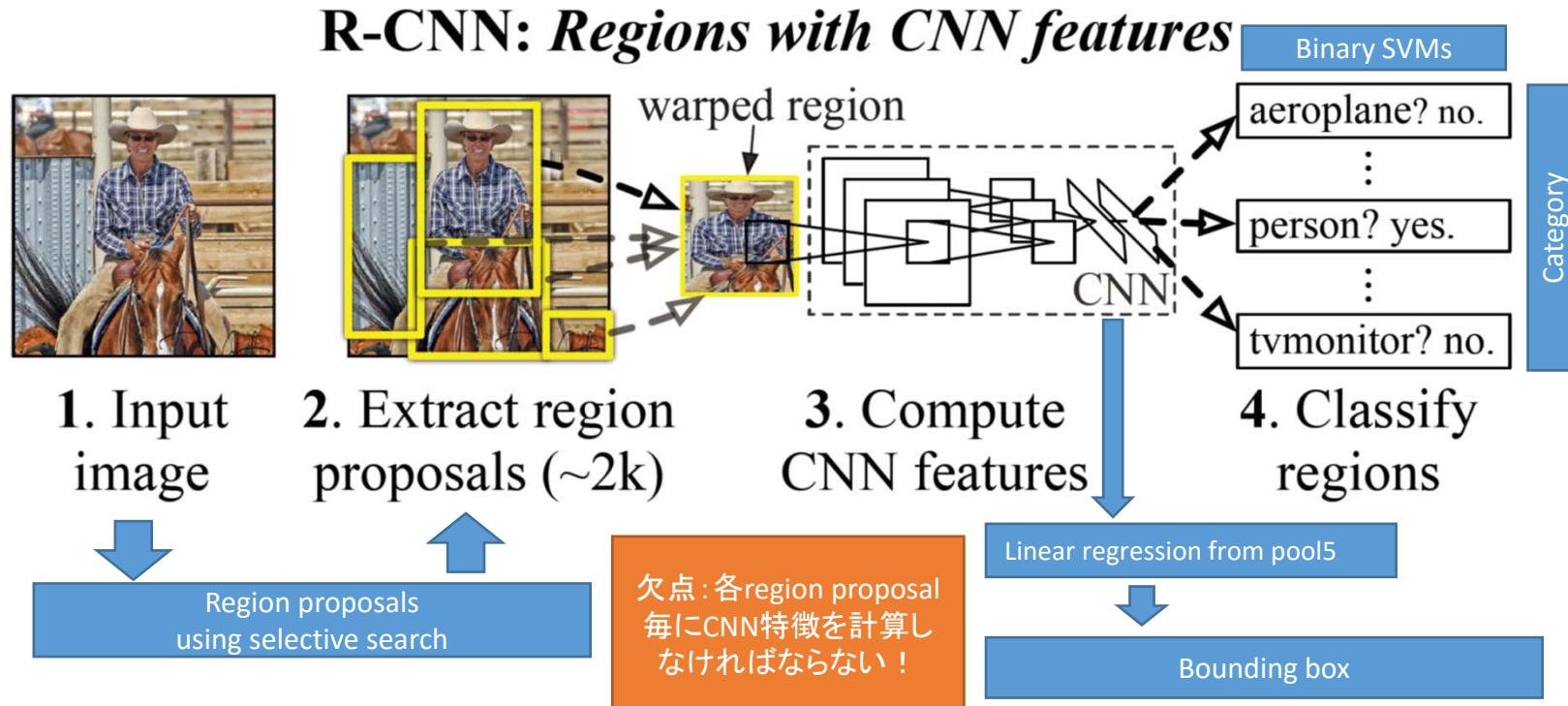
Example incorrect boxes and edge labeling



# Regions with CNN features (R-CNN)

Ross Girshick Jeff Donahue Trevor Darrell Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. CVPR, 2014.

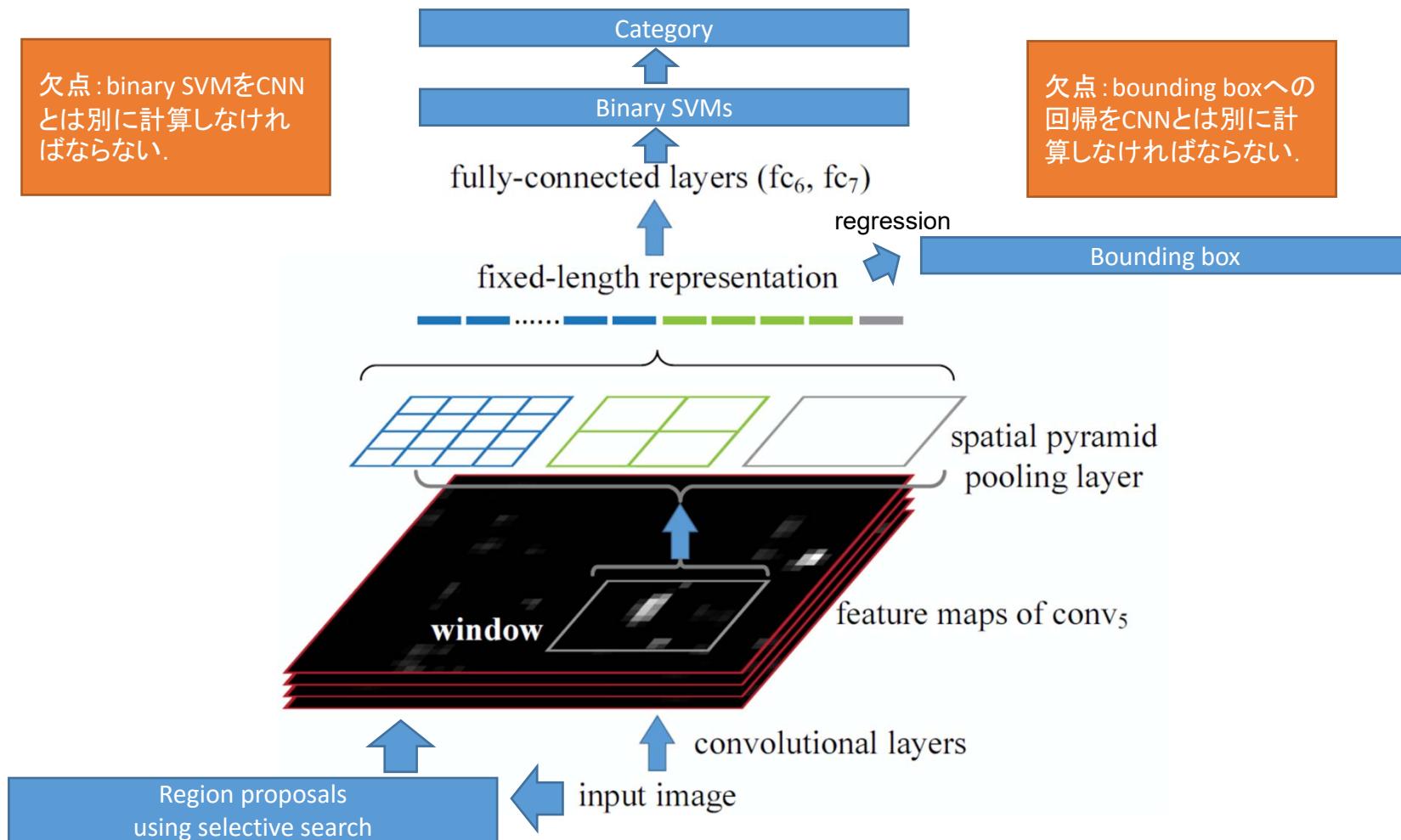
- 各region proposal内の画像を一定の大きさにリサイズした後にCNNに入力する。
- CNNは大規模データでpre-trainしておいて、対象に合わせてfine-tuneする。
- 得られた各region proposal内CNN特徴をbinary SVMに入力してクラス識別を行う。SVMは各クラスごとに準備して、hard negative miningを用いて学習する。
- Region proposal内のpool5特徴から、bounding boxを回帰する。



# SPP-net for object detection

Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun. Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition.  
arXiv:1406.4729, 2014.

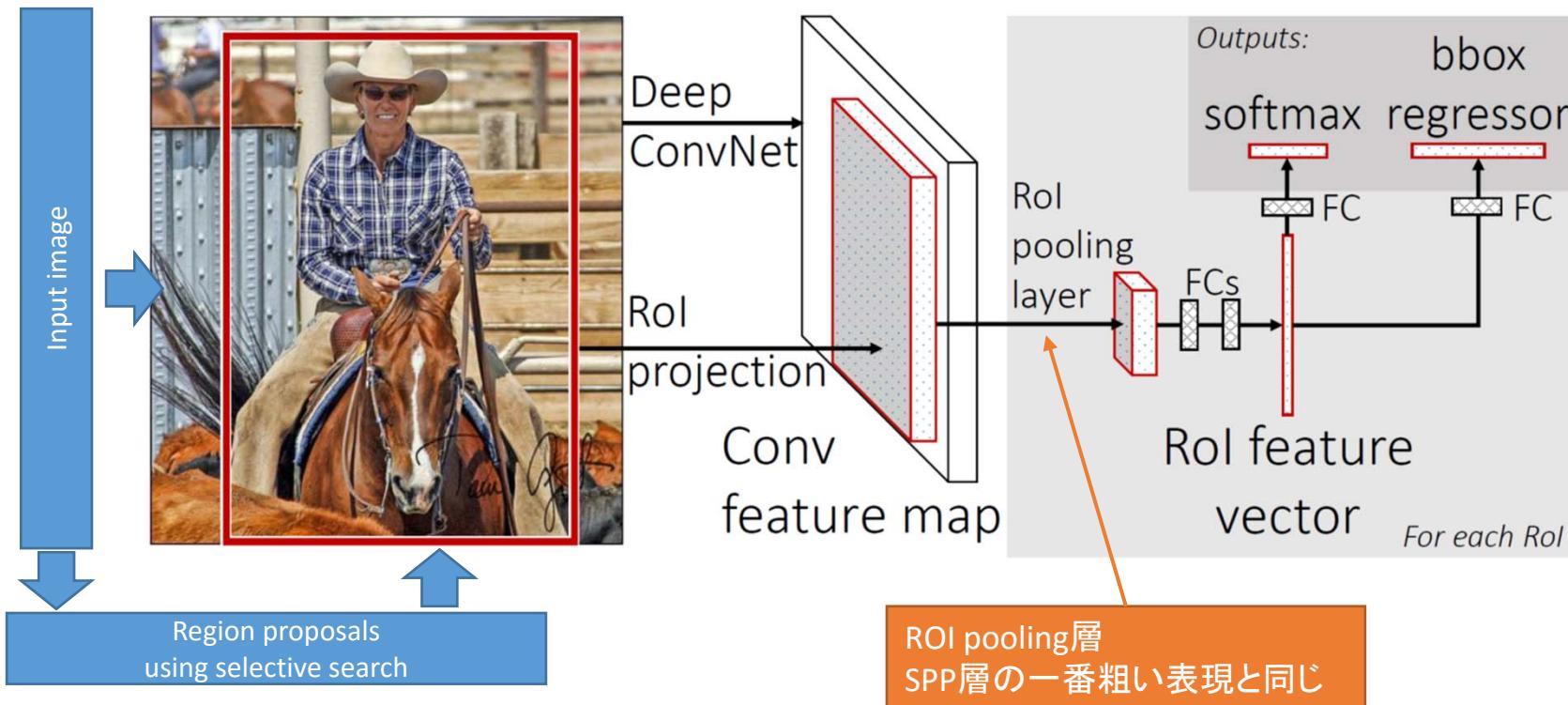
- Region proposal内のfeature mapsをSPP層を利用して一定次元の特徴ベクトルに変換する。
- 変換後の特徴をbinary SVMで識別する。
- Feature mapは一度計算するだけでよいので、高速化が図れる。
- Conv5のプール後の特徴を利用して、バウンディングボックスへ回帰する。



# Fast R-CNN

Ross Girshick. Fast R-CNN. arXiv:1504.08083.

- SPP-netと同様に, feature mapsは一度計算するだけでよい.
- Region proposal内のfeature mapsをregion of interest (RoI) pooling層を利用して一定次元の特徴ベクトルに変換する. Region proposalsはselective searchを利用してあらかじめ計算しておく.
- RoI pooling層の後に全結合層に入力する. その後クラス識別の層とbounding boxへ回帰する層へ分岐する.
- マルチタスク損失を最適化することでクラス識別とBbox回帰の学習を同時に行う.



マルチタスク損失

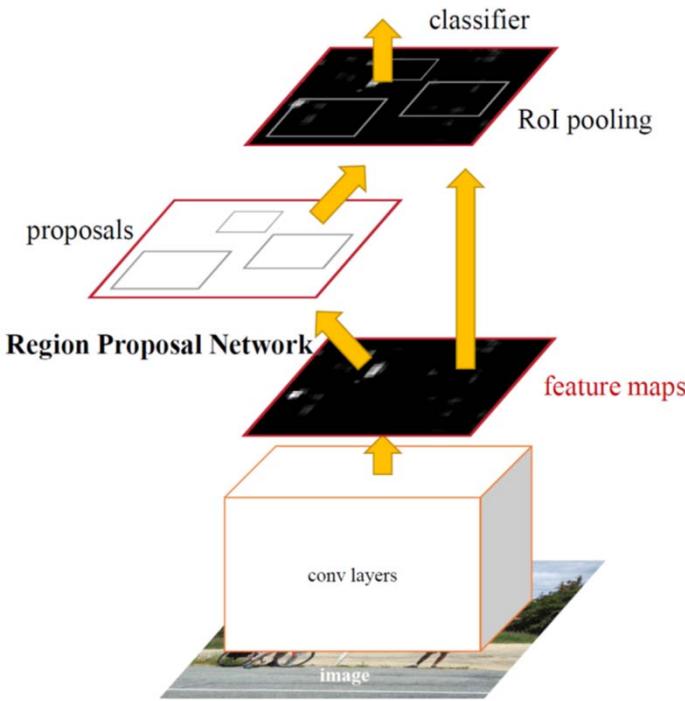
$$L(p, u, t^u, v) = L_{\text{cls}}(p, u) + \lambda[u \geq 1]L_{\text{loc}}(t^u, v)$$

$$L_{\text{cls}}(p, u) = -\log p_u \quad L_{\text{loc}}(t^u, v) = \sum_{i \in \{\text{x}, \text{y}, \text{w}, \text{h}\}} \text{smooth}_{L_1}(t_i^u - v_i)$$

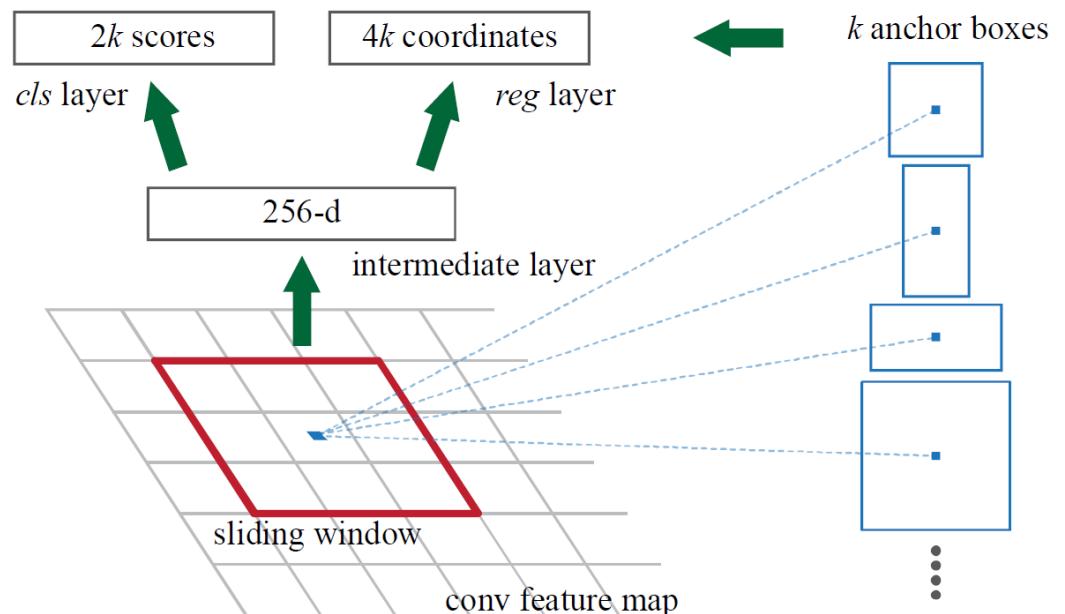
欠点: region proposalの計算コストが高い.

# Faster R-CNN

- Region Proposal Network (RPN)
  - Feature mapsからregion proposalを推定する.
- Faster R-CNN
  - RPNとFast R-CNNの組み合わせ.



Shaoqing Ren, Kaiming He, Ross Girshick, Jian Sun. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks.  
arXiv:1506.01497.



- Conv feature map上に、スライディングウインドウにより、ウインドウ内のfeature mapを小さなネットワークを通じて一定次元(256 dim, 512 dimなど)の特徴に変換.
- スライディングウインドウでの各サンプリング点において $k$ 個(3 scales, 3 aspect ratios,  $k=9$ など)のアンカーボックスを準備する.
- スライディングウインドウのサンプリング数を $WH$ とすると、 $WHk$ のアンカーを一枚の画像から得られる.
- マルチタスク損失を最小化するようにRPNを学習する.

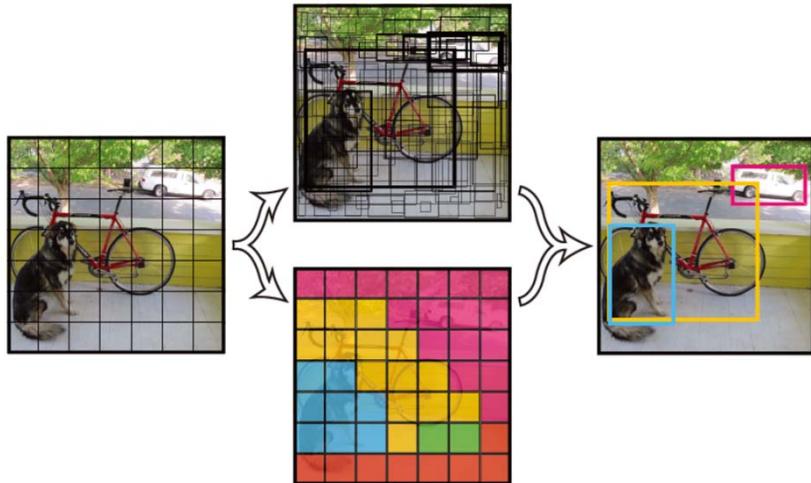
## Faster R-CNNの学習

1. RPNの学習, imagenet pre-train, fine-tune
2. 1.のRPNから得られるregion proposalsでfast R-CNNを学習
3. RPNとfast R-CNNの統合モデルを利用して、RPN独自のネットワークパラメータを最適化
4. Fast R-CNN独自のパラメータを最適化

# YOLO

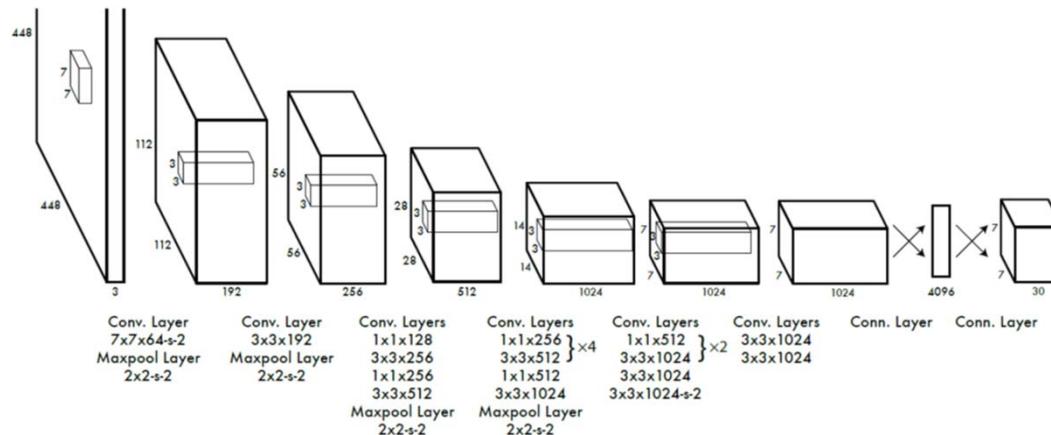
Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You Only Look Once: Unified, Real-Time Object Detection. CVPR, 2016.

<http://pjreddie.com/darknet/yolo/>



**Figure 2: The Model.** Our system models detection as a regression problem. It divides the image into an even grid and simultaneously predicts bounding boxes, confidence in those boxes, and class probabilities. These predictions are encoded as an  $S \times S \times (B * 5 + C)$  tensor.

- Each bounding box consists of 5 predictions:  $x$ ,  $y$ ,  $w$ ,  $h$ , and confidence.
- Each grid cell predicts  $B$  bounding boxes and confidence scores for those boxes.



# YOLO

Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You Only Look Once: Unified, Real-Time Object Detection. CVPR, 2016.

<http://pjreddie.com/darknet/yolo/>

Real-Time Detectors	Train	mAP	FPS
100Hz DPM [30]	2007	16.0	100
30Hz DPM [30]	2007	26.1	30
Fast YOLO	2007+2012	52.7	<b>155</b>
YOLO	2007+2012	<b>63.4</b>	45
Less Than Real-Time			
Fastest DPM [37]	2007	30.4	15
R-CNN Minus R [20]	2007	53.5	6
Fast R-CNN [14]	2007+2012	70.0	0.5
Faster R-CNN VGG-16[27]	2007+2012	73.2	7
Faster R-CNN ZF [27]	2007+2012	62.1	18

**Table 1: Real-Time Systems on PASCAL VOC 2007.** Comparing the performance and speed of fast detectors. Fast YOLO is the fastest detector on record for PASCAL VOC detection and is still twice as accurate as any other real-time detector. YOLO is 10 mAP more accurate than the fast version while still well above real-time in speed.