1P-08

畳み込みニューラルネットワークの特徴マップ選択によるトラッキング

山田 真生 † 渡辺 崇 †

† 名古屋大学大学院 情報科学研究科

1. はじめに

VGG Net [1] に代表される一般物体認識で高い認識精度を達成した畳み込みニューラルネットワーク (Convolutional Neural Networks，以下 CNN) は，一般物体認識以外の問題にも数多くの応用が行われている．

こうした CNN を利用する既存のトラッキング手法では，オフライン学習やオンライン学習を行うために多くの計算量を必要とする．そこで本研究では，学習済みの CNN からトラッキングに有効な特徴マップを選択することで，新たなモデルの学習を行うことなく対象の追跡を行う高速なトラッキング手法を提案する．

2. 関連研究

Wang ら [2] は CNN 後段の全結合層を取り除いて畳み込み層を追加し，追跡対象を中心とする 2 次元ガウス分布のマップを出力するようネットワークのオンライン学習を行った．過学習を避けるために学習に必要のない特徴マップを除外する，特徴マップ選択と呼ばれる処理を同時に提案している．Ma ら [3] は CNN の畳み込み層から計 3 層を選び，各層の特徴量から 2 次元ガウス分布のマップを予測する相関フィルタのオンライン学習を行った．Tao ら [4] は CNN の複数の層から得られる特徴を統合して，2 枚の画像に対して同一の物体が映っているかを判定する関数のオフライン学習を行った．

3. 提案手法

3.1 候補領域の生成

矩形 $\mathbf{x}_t = \{x_t, y_t, w_t, h_t\}$ により，時刻 t における追跡対象の領域を定める．x, y, w, h はそれぞれ矩形の中心座標，幅，高さを表す．Wang ら [2] の手法に基いて領域のランダムサンプリングを行い，候補領域を得る．x_t，y_t はそれぞれ平均 x_{t-1}，y_{t-1}，標準偏差 $\sigma_{x,y} \max(w_{t-1}, h_{t-1})$ の正規分布に，w_t，h_t については前時刻との比を平均 1，標準偏差 $\sigma_{w,h}$ の正規分布に従いサンプリングする．

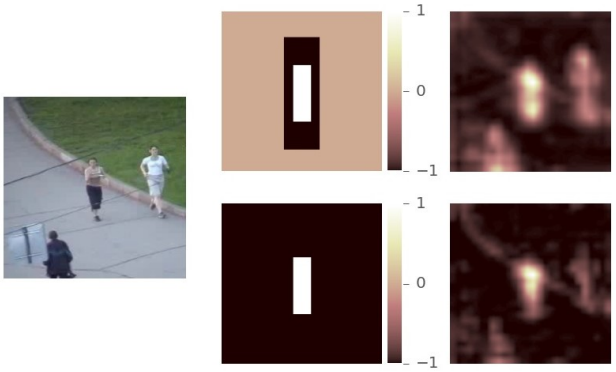

図 1: 特徴マップ選択の流れ．左から順にそれぞれ入力画像，目標マップ，予測マップを表す．異なる 2 枚の目標マップを用いた結果をそれぞれ上段と下段に示す．

3.2 特徴マップ選択

提案する特徴マップ選択の流れを図 1 に示す．$\mathbf{x}$ に対して中心座標は同一で長辺の 3 倍を一辺とする正方領域 ROI を画像から切り抜く．ROI と同じサイズで $\mathbf{x}$ に対応する要素を 1，周囲を -1 とする行列を目標マップ $\mathbf{M}^l$ とする，セグメンテーションと，他の物体との識別に有効な特徴を得るために，$\mathbf{M}^l$ として -1 とする領域を $\mathbf{x}$ の 2 倍，または ROI 全体とした 2 種類の目標マップを用意する．ROI を CNN に入力して得た第 l 層の特徴マップの c チャネル目を $\mathbf{F}^l_c$ として，$\mathbf{F}^l_c$ の重要度 s^l_c を次式で定義する．

$$s^l_c = \mathrm{sum}(\mathbf{F}^l_c \odot \mathbf{M}^l) \tag{1}$$

$\odot$ は行列の要素毎の積，sum は行列の全要素の和を表す．第 l 層について重要度 s^l_c が上位であるチャネルの

Selecting Feature Maps of Convolutional Neural Network for Visual Tracking
Masaki YAMADA† and Takashi WATANABE†
†Graduate School of Information Science, Nagoya University

集合 C^l について特徴マップの和

$$\hat{\mathbf{M}}^l = \sum_{c \in C^l} \mathbf{F}_c^l \tag{2}$$

をとり，予測マップ $\hat{\mathbf{M}}^l$ を得る．

3.3 候補領域の評価

予測マップ上での候補領域の評価の際には，Wang ら [2] の手法に加えて前時刻と現時刻の中心座標間のユークリッド距離 d を考慮した．予測マップは最大値で割りレンジを $[0,1]$ にした上で，値の小さい領域を除外するため b ずらす．候補領域 R_i，$i \in \{1,2,...,N\}$ のスコア

$$score_i = \sum_{(x,y) \in R_i} (\hat{\mathbf{M}}(x,y) - b) \tag{3}$$

を求め，このスコアを基に候補領域の確信度を

$$conf_i = (1 - \frac{d_i}{D})(score_i - \min_j score_j) \tag{4}$$

で定義する．D は ROI の一辺の $\frac{1}{2}$ 倍とする．このとき $conf_i$ が最大の領域 R_i を予測として用いる．

3.4 提案手法の全体

提案手法の概略をアルゴリズム 1 に示す．CNN は 16 層の VGG Net [1] とした．特徴マップには conv4-3, conv5-3 の計 2 層，$l \in \{4,5\}$ を用いる．元の conv5-3 層は解像度が低いため pool4 層を外している．異なる層から得た予測マップは，最大値で割ることでレンジを $[0,1]$ に揃える．2 種類の目標マップと $l \in \{4,5\}$ についてそれぞれ特徴マップ選択の手法を適用し，得られた予測マップ 4 枚の和を全体の予測マップ $\hat{\mathbf{M}}$ とする．式 (2) の予測マップの算出には移動平均により更新を行う平均重要度 $\bar{\mathbf{s}}^l$ を用いる．

Algorithm 1 提案トラッキング手法

Input: $\mathbf{x}_0$，画像 $\mathbf{I}_t$，$t \in \{0,1,...,T\}$
Output: $\mathbf{x}_t$, $t \in \{1,2,...,T\}$

1: $\mathbf{I}_0$ から $\mathbf{x}_0$ 中心の ROI を切り抜き $\mathbf{F}^l$ を得る;
2: $\mathbf{x}_0$ と $\mathbf{F}^l$ から式 (1) に従い $\mathbf{s}^l$ を得て $\bar{\mathbf{s}}^l \leftarrow \mathbf{s}^l$ とする;
3: **for** $t = 1$ to T **do**
4: 　$\mathbf{I}_t$ から $\mathbf{x}_{t-1}$ 中心の ROI を切り抜き $\mathbf{F}^l$ を得る;
5: 　$\mathbf{F}^l$ と $\bar{\mathbf{s}}^l$ から式 (2) に従い $\hat{\mathbf{M}}^l$ を得る;
6: 　$\hat{\mathbf{M}}$ 上で式 (4) を最大にする領域を $\mathbf{x}_t$ とする;
7: 　$\mathbf{x}_t$ と $\mathbf{F}^l$ から式 (1) に従い $\mathbf{s}^l$ を得る;
8: 　$\bar{\mathbf{s}}^l \leftarrow \eta\bar{\mathbf{s}}^l + (1-\eta)\mathbf{s}^l$ とする;
9: **end for**

4. 評価実験

Wu ら [5] によるデータセットで提案手法の評価を行った．Intel i7-4790 3.60GHz CPU と 16GB のメモリ，NVIDIA GeForce TITAN X GPU を搭載した PC で実験を行った．これは処理速度を比較する 2 手法 [2, 3] とほぼ同様の構成である．ROI は 224×224, $\hat{\mathbf{M}}$ は 112×112 にリサイズする．C^l は重要度 s_c^l が上位 $\frac{1}{10}$ のチャネルの集合とした．候補領域の生成では $\sigma_{x,y} = \frac{1}{3}$, $\sigma_{w,h} = 0.01$，$N = 600$ とし，評価の際には $b = 0.2$ を用いた．$\bar{\mathbf{s}}^l$ の更新では $\eta = 0.99$ を用いた．

Wu ら [5] により提案されている評価基準値と処理速度を表 1 に示す．提案手法は処理速度で 2 手法を上回り，予測中心座標の精度 (precision) と予測領域の精度 (success) では，FCNT を 3, 4% ほど下回った．

表 1: 他の CNN を利用したトラッキング手法との比較．Wu ら [5] のデータセットにおける閾値 20 ピクセルでの precision rate と success rate 曲線の AUC，加えて処理速度を示す．

	Ours	FCNT [2]	HCF [3]	SINT [4]
precision (%)	81.3	85.6	89.1	88.2
success (%)	56.9	59.9	60.5	65.5
speed (fps)	24.3	~ 3	11.0	-

5. おわりに

一般物体認識を学習済みの CNN を用いて，オフライン学習やオンライン学習を行うことなく高速にトラッキングを実現する手法を提案した．オンライン学習やオフライン学習を行う既存手法と比較すると予測精度はやや下回るが，24.3 fps の高速な処理を実現した．

参考文献

[1] Simonyan, K. and Zisserman, A.: Very deep convolutional networks for large-scale image recognition, arXiv preprint arXiv:1409.1556 (2014).
[2] Wang, L., Ouyang, W., Wang, X. and Lu, H.: Visual tracking with fully convolutional networks, *Proc. ICCV*, pp.3119-3127 (2015).
[3] Ma, C., Huang, J. B., Yang, X. and Yang, M. H.: Hierarchical convolutional features for visual tracking, *Proc. ICCV*, pp.3074-3082 (2015).
[4] Tao, R., Gavves, E. and Smeulders, A. W. M.; Siamese instance search for tracking, *Proc. CVPR*, pp.1420-1429 (2016).
[5] Wu, Y., Lim, J. and Yang, M. H.: Online object tracking: A benchmark, *Proc. CVPR*, pp.2411-2418 (2013).