社団法人　電子情報通信学会
THE INSTITUTE OF ELECTRONICS,
INFORMATION AND COMMUNICATION ENGINEERS

信学技報
TECHNICAL REPORT OF IEICE.

FoodCam: スマートフォン上での リアルタイム食事画像認識による食事記録アプリケーション

河野　憲之†　　柳井　啓司†

† 電気通信大学大学院　〒 182–8285 東京都調布市調布ヶ丘 1–5–1
E-mail: †kawano-y@mm.cs.uec.ac.jp, ††yanai@cs.uec.ac.jp

あらまし　近年、スマートフォンが普及しその性能も向上している。従来のスマートフォンアプリケーションはデータをサーバに送り、サーバ上で画像認識をしていた。だが、通信コストがかかる。また、ユーザの増加により計算資源が多数必要になるという問題点がある。そこで、スマートフォン上で画像処理することが望まれる。本論文では、計算資源の限られたスマートフォン上でより高速、高精度に一般物体認識を行う手法を提案する。そして、従来の画像認識を用いた食事記録アプリケーションの画像認識の面において改良を行った。従来手法よりも大幅に認識性能が向上し、かつ高速であることを実験により確認した。さらに、サーバ上でのコストが大きい認識手法と比較しより高い認識性能を示し、有効性を確認した。

キーワード　食事画像認識，食事記録、スマートフォン，リアルタイム，スマートフォンアプリケーション

1. ま え が き

近年、Android フォンや iPhone などのスマートフォンが広く普及している。それに伴い、スマートフォンの性能が大きく向上し、スマートフォン上での画像認識が可能になった。現在、4 コアの CPU を搭載したスマートフォンが一般になり、一世代以前の PC の CPU と同等のパフォーマンスにまでなっている。従来の画像認識を行うシステムは、高性能なサーバに画像を送り、サーバ上で画像処理を行っていた。そのため、通信コストがかかる。また、ネットワーク環境にも依存する。さらに、システムを使用するユーザ数の増加に伴い、サーバの計算資源も増加させなければ、パフォーマンスが低下する。特に、食事記録アプリケーションは辞書アプリケーションと異なり、毎日多数のユーザが使用することが想定される。従って、コストのかかる画像認識処理は、ユーザのスマートフォン上で行うことがよいと考える。スマートフォン上で画像認識を行うことにより、サーバ上で画像認識を行う問題点として挙げた、ネットワーク環境に依存しない、通信コストがかからない、計算資源が分散されるという大きな利点がある。そのため、複数コアあるスマートフォンの計算資源のみを活用し、高速かつ高精度に画像認識を行う手法が望まれている。

我々は以前、スマートフォン上で画像認識を行い、ユーザの食事記録をとる補助をするアプリケーション [1,2] を提案した。図 1 は、我々の提案した食事記録アプリケーションのイメージである。本論文では、局所特徴量と画像表現の面において、画像認識部分の改良を行う。

だが、高速に画像認識、特にスマートフォン上で画像認識を行うことは、実用性が高いものの依然として困難なタスクでもある。ここでは、Android フォンを例に挙げる。CPU 周波数は 1.5GHz 前後と PC と比較すると大幅に低い。RAM も同様に、1 つのアプリケーションが使用できるメモリ量は現在の仕様では最大 256MB(ネイティブヒープと Java ヒープ (Dalvik ヒープ) の合計) しかない。そのため、低メモリかつ複雑な処理をしないことが求められる。スマートフォン上で高速かつ高精度に認識を行うためには、認識をする際の各モジュールについて計算コストが非常に大きいものがあると難しい。スマートフォンの計算資源を有効に活用しなければならない。

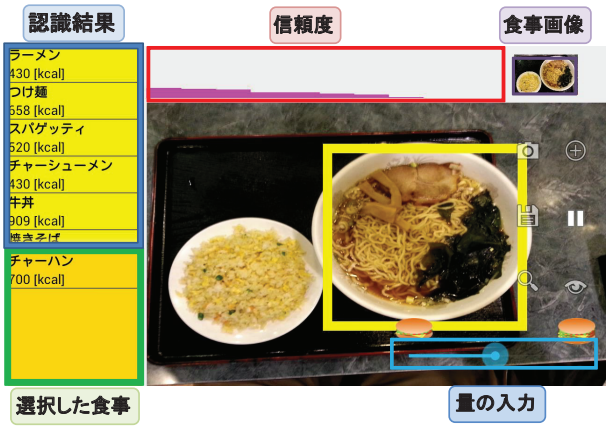


図 1　食事記録アプリケーションのイメージ

また、近年、高速な線形識別器に適した画像表現が複数提案されている。従来の一般物体認識は、画像表現に Bag-of-Features(BoF) を用い、識別に非線形識別器を用いることが一般的であった。非線形識別器は、学習は学習サンプル数 N に対し $\mathcal{O}(N^2) \sim \mathcal{O}(N^3)$、評価はサポートベクトル数 M に対し線形識別器よりも M 倍遅い。だが、画像表現を改良することにより、線形識別器の適用が可能になった。特に、BoF を改良した画像表現で Fisher Vector [3,4] が認識性能が最も高い結果になっている [5]。Fisher Vector は線形識別器に適していて、従来の BoF+非線形識別器と比較して大幅に性能が向上することが示されている。さらに、BoF の場合は、認識性能を高めるために大きなコードブックが必要であるが、それに伴い近傍探索の計算コストが増大する。Fisher Vector は、小さいコードブッ

クでも識別性能が高い。そして、計算コストも小さい。Fisher Vector を用いることで、スマートフォン上で高速、高精度に食事画像認識を行う。

実験では 100 種類の食事に対して提案手法と従来手法との認識性能の比較を行う。そして、従来のスマートフォン上で認識するシステムよりも大幅に性能が高く、さらに高速であることを実験により示す。また、BoF や大域特徴の多数特徴量+非線形 SVM によるコストが非常に大きい従来のサーバ側で画像認識を行うシステムと比較し、提案手法の有効性を示す。

2. 関 連 研 究

2.1 食事画像認識

まず、食事画像認識の研究を紹介する。松田ら [6] は、複数検出手法により食事領域を推定し、推定された領域を複数特徴量を用いて分類を行う食事画像認識エンジンを提案した。特徴量は、SIFT、CSIFT、色ヒストグラム、Gabor、HOG の 5 種類を用い、SIFT、CSIFT、色ヒストグラムは BoF に表現して使用している。そして、識別に非線形識別器を用いているため、非常にコストが高く、認識はクラスタマシンを用いて行っている。そのため、スマートフォン上で高速に画像認識をすることはできない。実験では、サーバサイドの認識手法として認識性能の比較を行う。また、食事画像からバランス推定をし、その結果を返す FoodLog [7] や、チェッカーボードとともに食事を撮影し、食事の分類と量の認識を行う TADAproject [8] がある。しかし、いずれもサーバに画像データを送り、画像処理しているため通信コストが高く、認識を誤った場合は、ユーザが後から手動で直すことになる。

本研究では、スマートフォン上で高速に食事を認識し、記録をとる。なお、量はユーザに入力してもらい、食事の種類の認識のみになっている。

2.2 モバイルデバイスと画像認識

次に、スマートフォンから利用できる画像認識アプリケーションを紹介する。Google Googles(注1) は物体認識システムとして有名なアプリケーションである。しかし、認識対象はロゴや芸術品など視覚的変化のない特定物体であり、本研究では、一般物体である食事を認識対象にしている。Kumar ら [9] は、スマートフォンで葉を撮影すると、葉独自の湾曲具合から形状特徴を抽出し葉を認識するアプリケーションを提案した (Leaf Snap)。だが、認識はサーバ上で行っている。そのため、通信コストがかかり、ネットワークにも依存する。本研究では、サーバ側で画像認識を行う従来のスタイルでなく、スマートフォン上で画像認識を行う新しいスタイルを目的としている。

最後に、スマートフォン上で画像認識をするアプリケーションを紹介する。Lee ら [10] は、テンプレートを学習し、方向と強度に関する記述子に分解し、複数スケールでのテンプレートマッチングにより、ユーザが登録した物体の検出や追跡をリアルタイムに実現している。Maruyama ら [11] は、視覚的変化の小さい食材の認識に対して、特徴量は 144 個のみ局所色ヒストグラムの BoF 一つであり、直接線形識別器に適用している。そのため、視覚的変化が大きい一般物体認識では認識精度が極めて低い。

本論文では、スマートフォン上でより高速かつ高精度に画像認識を行う手法として、局所特徴量に計算コストの小さい HOG Patch と Color Patch 特徴を用い、画像表現に Fisher Vector を用いる。

(注1)：http://www.google.com/mobile/goggles/

図 2　アプリケーション使用例

3. 従来アプリケーション

3.1 概　　要

以前我々が提案した食事記録アプリケーション [1] について、記録をとるまでの使用の流れを以下に、また例を図 2 に示す。

（1） スマートフォンを食事にかざす。
（2） ユーザは料理領域を入力する。
（3） 食事画像認識を行う。
（4） 一定時間後、認識結果上位を提示する。
（5） ユーザは認識結果上位から料理を選択する。
（6） 未選択の料理があれば、2 もしくは 3 に戻る。
（7） 食事画像を保存する。
（8） 食事の栄養を確認する。

2 では、ユーザによる食事領域の入力は対角線を引くことで矩形領域とし、バックグラウンドで領域推定による料理領域の補正が行われる。食事領域は 4 つまで入力可能であり、食事領域を入力しなければフレーム全体から食事画像認識を行う。3、4 では、一定時間食事画像認識を繰り返し、各識別器の最終的な出力値はその間の出力値の平均とする。そして、最終的な出力値を降順にソートして、上位をユーザに表示する。候補リストの更新の間隔はユーザ調査により決定している。5 では、料理があると考えられる方向を提示することにより、認識結果上位に目的の料理が現れなかった場合の対処をする。記録にはメモや位置情報も登録でき、サーバにアップロードすることにより食事記録をユーザ間で共有可能である。

食事記録の閲覧では以下が可能でる。

（1） 日ごとに食事記録を閲覧
（2） Google Maps 上で閲覧
（3） 最近の食の傾向の確認
（4） アップロードされた食事記録の閲覧

3.2 認識手法

次に、従来手法で用いられた画像認識手法について説明する。

3.2.1 認識領域の修正

ユーザは食事領域を入力する際、実際の領域よりも大きく入力する傾向がある。そこで、ユーザが食事領域を入力すると、バックグラウンドで GrabCut を用い背景を除外し、前景を全て囲む最小の矩形に認識領域を修正する。

3.2.2 認識領域に対する種類分類

画像特徴量は、色ヒストグラムと SURF-BoF を用いていた。色ヒストグラムは、画像を 3×3 に分割しそれぞれ 64 次元色ヒストグラムを抽出し、結合したベクトルを使用した。SURF-BoF は、画像から SURF [12] をデンスサンプリングで抽出し、500 次元の BoF 表現で特徴ベクトルにした。このとき、識別性能を高めるために、soft-assignment [13] を適用した。識別器は、線形 SVM を用いた。L1 正規化された特徴ベクトル (色ヒストグラム、SURF-BoF) は、線形識別器での識別性能が悪いため、ともに χ^2 kernel feature map [14] を適用し、特徴ベクトルをあらかじめ少し高次元に射影することによって、線形識別器への適用を可能にした。これにより、線形識別器でカーネル関数が χ^2 kernel である非線形識別器と同等の認識精度をだすことが可能になっている。

しかし、SURF [12] は高速な記述子として物体認識で広く用いられるが、計算資源の限られたスマートフォン上ではコストが高い。カラーヒストグラムは単純で高速であるが、より高精度な認識をするために認識に適した色特徴を用いることが考えられる。

3.2.3 食事性の高い方向提示

正しく認識できないとき、ユーザはどの方向を写せば認識できるかわからない。そこで、食事性の高い領域を提示し、その方向を写すことで認識性能を高める。手法は、領域内に多数矩形領域を考え、各矩形での BoF を直接線形 SVM にかけ、最もスコアが高くなった矩形の重心の方向を提示する。BoF と線形 SVM のスコアは積分画像を作成しておくことで $\mathcal{O}(1)$ で得ることができる。

しかし、食事性の高い方向提示はユーザからの評価が低かった [1]。その後の調査においても高い評価を得ることができなかったため、本論文では使用していない。

3.3 認識時間

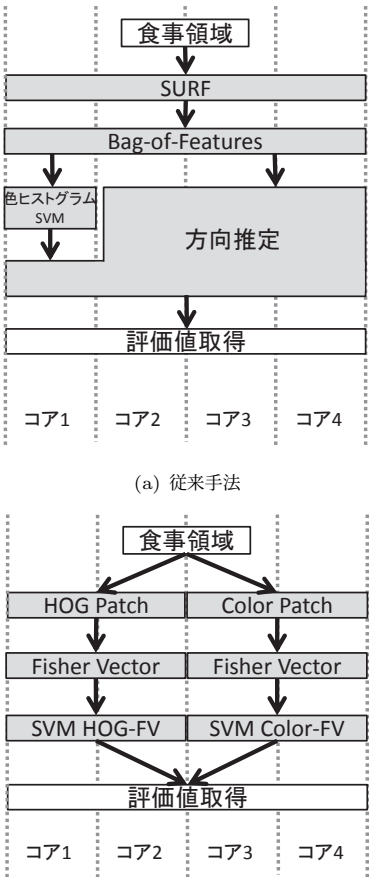
画像データを受け取り各識別器の評価値を得るまでの時間は 1 回あたり平均 0.26 秒であった。食事性の高い方向も推定したため、その際は平均 0.34 秒であった。本論文では食事性の高い方向は推定しないため、従来の認識時間は 0.26 秒として比較を行う。

4. 提案手法

スマートフォン上で高速かつ高精度に画像認識を行う手法として、特徴量は、HOG Patch と Color Patch の 2 種類の局所特徴量を用いる。そして、各局所特徴量は Fisher Vector で表現する。識別器は線形 SVM を用い、one-vs-rest 法により分類を行う。従来手法と提案手法の認識の流れを図 3 に示した。

4.1 局所特徴量

本論文では、HOG Patch と Color Patch の局所特徴量を用いる。画像は、大きい場合には、計算コストを抑えるためにアスペクト比を保ったまま 200 × 150 ピクセルにリサイズした。



(a) 従来手法

(b) 提案手法

図 3 従来手法と提案手法の認識の流れ

局所特徴量は、ともにスケール 16 と 24 ピクセルとし、6 ピクセルごとのデンスサンプリングで抽出した。局所特徴は多数抽出するため、中間データが多くなる。メモリ面を考慮し低次元に抑える。本論文では、パッチを SIFT [15] や SURF [12] に見られる 4 × 4 でなく、2 × 2 に分割し特徴量を抽出することで局所特徴量の次元数を抑えた。

4.1.1 HOG Patch

Histogram of Oriented Gradients(HOG) [16] は、N.Dalal らによって提案された。勾配ヒストグラムに基づいた記述子であるため SIFT [15] に類似している。だが、記述が単純なため SIFT [15] や SURF [12] よりも高速に記述することができる。これは、スマートフォン上でより高速に認識を行うために、重要な要素である。また、その分特徴点を密にとることが可能であり、認識性能が向上する。

本論文では、HOG を局所記述子として局所パッチから抽出する。局所パッチを 2 × 2 に分割し、それぞれ 8 方向、合計 32 次元の HOG を抽出した。そして、PCA を適用し 24 次元に圧縮した。

4.1.2 Color Patch

色特徴は画素の RGB 値の平均と分散を用いる。局所パッチを 2 × 2 に分割し、それぞれ RGB の平均と分散 6 次元、合計 24 次元の色特徴を抽出した。そして、PCA を適用したが、次元数は 24 次元を保った。

4.2 Fisher Vector

Fisher Vector [3,4] は、BoF [17] と異なり高次統計量を含め

ることにより量子化誤差を軽減している。さらに、BoF を拡張した手法で最も性能が高いとされる [5]。局所特徴量の総数 T、局所特徴群 $X = \{x_t, t = 1, ..., T\}$ に対する Fisher Vector $\mathcal{G}_\theta^X$ は以下で表される。

$$\mathcal{G}_\theta^X = L_\theta(\frac{1}{T}\nabla_\theta \log p(X|\theta)) = L_\theta G_\theta^X \quad (1)$$

ここで、$p(X|\theta)$ は確率密度関数、$\nabla_\theta \log p(X|\theta)$ は対数尤度の勾配、F_θ をフィッシャー情報行列とすると、F_θ は $F_\theta^{-1} = L'_\theta L_\theta$ にコレスキー分解される。Fisher Kernel $K(X,Y) = G_\theta^{X'} F_\theta^{-1} G_\theta^Y$ は Fisher Vector の内積で表されるため、線形識別器で効率よく識別可能になる。

Perronnin ら [4] に従い、確率密度関数に Gaussian Mixture Model(GMM) を仮定した。そのとき、確率密度関数は以下で与えられる。

$$p(x|\theta) = \sum_{i=1}^{K} \pi_i \mathcal{N}(x|\mu_i, \Sigma_i) \quad (2)$$

x は局所記述子、K はガウシアンのコンポーネントの数、$\theta = \{\pi_i, \mu_i, \Sigma_i, i = 1, ..., K\}$ は GMM のパラメータである。π_i は混合係数、μ_i は平均のベクトル、Σ_i は共分散行列を表す。共分散行列は対角行列とし対角成分を σ^2 と表し分散のベクトルで表す。

$\gamma_t(i)$ をガウシアン i への x_t の soft assignment とする。平均と分散に関する勾配 $\mathcal{G}_{\mu,i}^X$ と $\mathcal{G}_{\sigma,i}^X$ は、以下で表される。

$$\mathcal{G}_{\mu,i}^X = \frac{1}{T\sqrt{\pi_i}} \sum_{t=1}^{T} \gamma_t(i) \left(\frac{x_t - \mu_i}{\sigma_i} \right) \quad (3)$$

$$\mathcal{G}_{\sigma,i}^X = \frac{1}{T\sqrt{2\pi_i}} \sum_{t=1}^{T} \gamma_t(i) \left[\frac{(x_t - \mu_i)^2}{\sigma_i^2} - 1 \right] \quad (4)$$

最終的に Fisher Vector $\mathcal{G}_\theta^X$ は勾配 $\mathcal{G}_{\mu,i}^X$ と $\mathcal{G}_{\sigma,i}^X$ を全てのコンポーネントに対して求め、連結することで $2KD$ 次元のベクトルとなる。

本論文では、GMM のコンポーネントの数を 32、局所特徴はともに PCA で 24 次元にしたため、特徴ベクトルはともに 1536 次元になった。従来手法の特徴ベクトルの次元数と同程度になるように設定した。そして、識別性能を高めるために、パワー正規化 ($\alpha = 0.5$) と L2 正規化を適用した [4]。

4.3 識 別 器

識別器に線形 SVM を用い、one-vs-rest 法によりマルチクラス識別を行う。

スマートフォン上で認識を行うために、低コスト、低メモリで評価値を得られることが要求される。線形 SVM は、あらかじめサポートベクトルとサポートベクトルの重みを計算しておくことにより、データベクタ $\mathbf{x}$ と重みベクタ $\mathbf{w}$ の内積 $f(\mathbf{x}) = \langle \mathbf{w}, \mathbf{x} \rangle$ で与えられる。つまり、特徴ベクトルの次元数 N に対し、計算コスト $\mathcal{O}(N)$ のオペレーションで評価値を得ることができ、必要なメモリ量は $\mathcal{O}(N)$ である。SVM は、特徴量ごとに学習を行い、late fusion で結合した。SVM は LIBLINEAR [18] を使用し、オフラインで学習した。

4.4 実　　装

実装は、一般的な 4 コア 4 スレッドデバイスを想定し、マルチスレッドによる並列処理を行った。HOG Patch、Color Patch それぞれ 2 コアずつ用い並列処理した。特徴量の抽出、PCA による次元圧縮、Fisher Vector にエンコーディング、パワー正規化、L2 正規化、SVMs による識別を特徴量ごとにそれぞれ 2 並列、合計 4 並列により処理した。

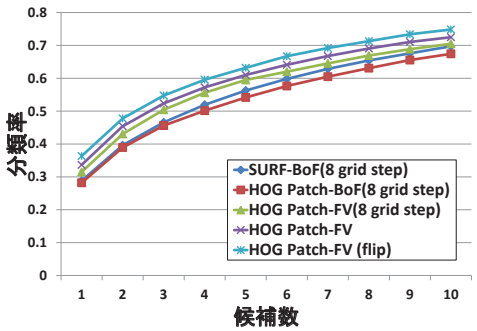


図 5　SURF-BoF、HOG Patch-BoF、HOG Patch-FV の分類率

HOG Patch は最初に画像全体から勾配方向と勾配強度を算出しておくことで高速に局所特徴を抽出した。Fisher Vector の平均に関する勾配 Eq.(3) は以下のようにすることで演算回数を減らし、高速化した。

$$\mathcal{G}_{\mu,i}^X = \frac{1}{\sqrt{\pi_i}\sigma_i} \frac{1}{T} \sum_{t=1}^{T} \gamma_t(i)(x_t - \mu_i) \quad (5)$$

GMM により事後確率を求める際や、平均や分散に関する勾配を求める際に必要になる項で、オフラインに計算可能な部分はあらかじめ計算しておき、ルックアップテーブルを作成し高速化した ($r_t(i)$ を求める際に用いる $\log \pi_i - 0.5 \times \log |\Sigma_i|$、Eq.(4) の $1/\sqrt{2\pi_i}$、$1/\sigma^2$、Eq.(5) の $1/\sqrt{\pi_i}\sigma_i$)。SVM はオフラインで学習しておいた。そして、認識時に用いる値は全てメモリにロードしておいた (PCA のための固有値、固有ベクトル、作成したルックアップテーブル、GMM の平均、SVMs の重みベクタ)。多数メモリにロードしているが、Fisher Vector は BoF よりも小さいコードブックで高い性能を示すことで可能であり、局所特徴量の次元も小さく抑えている。Fisher Vector 生成のためにメモリにロードしておいた値は BoF のコードブックよりも小さく、メモリ面においても優れている。

5. 実　　験

5.1 セットアップ

データセットは我々が構築した食事画像データセットを用いる。それは、サーバサイドの認識手法 [6] とクライアントサイドの認識手法 [1] で用いられ、比較に適していると考えられる。図 4 は、本論文の認識対象である 100 種類の食事のサンプルである。データセットは、各 100 枚以上ある食事 100 種類、合計 12,905 領域が人手でバウンディングボックスを付与されている。本論文では、特徴量を抽出する領域は、人手で与えられた正しい食事領域とした。検証、テストに各 20 枚、残りを学習に使用し評価することを、ランダムにデータを入れ換え 5 回繰り返しその平均値で評価した。次に、認識時間を計測するために画像データが与えられ全ての識別器の評価値を得るまでの時間 (認識時間) を計測した。比較とした従来手法での認識時間は、本論文で用いていない食事性の高い方向推定に要した時間は含めていない。今回実験に使用したデバイスは、Galaxy NoteII(1.6GHz 4 コア 4 スレッド Android4.1) であり、従来 [1] と同様である。

図 4　認識対象の 100 種類の食事のサンプル

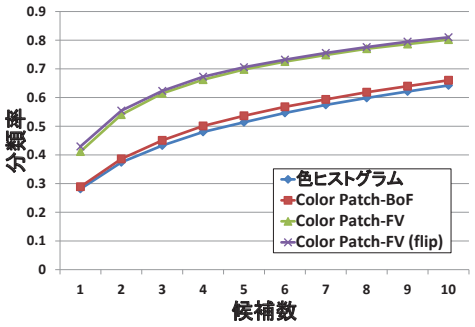


図 6　色ヒストグラム、Color Patch-BoF、Color Patch-FV の分類率

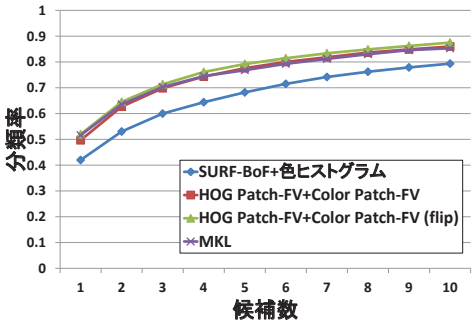


図 7　提案手法とクライアントサイドの従来手法 [1]、サーバサイドの手法 [6] での分類率

表 1　提案手法とクライアントサイドの従来手法 [1]、サーバサイドの手法 [6] での 1 位と 5 位以内の分類率

認識手法	1 位	5 位以内
SURF-BoF+色ヒストグラム [1]	42.0	68.2
HOG Patch-FV+Color Patch-FV	49.7	77.6
HOG Patch-FV+Color Patch-FV(flip)	51.9	79.2
MKL [6]	51.6	76.8

5.2 認識精度

最初に、SURF-BoF と HOG Patch-BoF、HOG Patch-FV を比較し HOG Patch の性能評価を行う。結果を図 5 に示した。SURF-BoF と同様にデンスサンプリングの間隔を 8 ピクセルにし、BoF は 500 次元とし soft assignment [13] を適用、そして χ^2 kernel feature map [14] を適用した。そのため、1500 次元の特徴ベクトルになった。BoF 生成の際には、局所特徴群に対し PCA を適用していない。SURF-BoF と HOG-BoF の認識精度の差は 1 位で 0.62%、5 位以内で 2.1%の性能低下のみであった。FV にすることで SURF-BoF より 1 位で 2.70%、5 位以内で 3.22%性能が向上した。そして、HOG は SURF と比較して高速なため本論文ではサンプリングの間隔を 6 ピクセルにした。認識精度が上がり、さらに、左右反転した画像を学習データに加えた (HOG-FV(flip)) 場合、SURF-BoF より 1 位で 7.52%、5 位以内で 6.9%高い 36.3%、63.2%を達成した。左右反転した画像を加えることで、HOG の回転不変でない性質を補うことができたと考える。

表 2　認識時間

	time[sec]	認識対象数
SURF-BoF+色ヒストグラム [1]	0.26	50
HOG Patch-FV+Color Patch-FV	0.065	100

次に、Color Patch の性能評価を行う。結果を図 6 に示した。Color Patch-BoF は色ヒストグラムより分類率が 1 位で 0.76%、5 位以内で 2.28%のみ高い。だが、FV にすることで性能が劇的に上昇し、色ヒストグラムより 1 位で 13.0%、5 位以内で 18.4%高くなった。左右反転を加えることでわずかに性能が向上し、1 位で 43.0%、5 位以内で 70.6%を達成した。

最後にアプリケーションとしての認識性能の評価を行う。提案手法 (HOG Patch-FV+Color Patch-FV)、(HOG Patch-FV+Color Patch-FV(flip)) と、従来の手法 (SURF-BoF+色ヒストグラム)、またサーバサイドの手法 [6] と比較を行う。[6] は複数品の食事の認識が主であるが、100 種類の単品の食事においても分類率を評価している。そのため、[6] で示されている単品の食事において食事領域が与えられた際の食事 100 種類の分類率を示した (MKL)。結果を図 7、また 1 位と 5 位以内の結果を表 1 に示した。

実験結果は、HOG-FV+Color-FV の場合 1 位で 49.7%、5 位以内で 77.6%を達成し、反転画像を加えた場合 1 位で 51.9%、5 位以内で 79.2%を達成した。色特徴のみで従来の手法 [1] よりも高い認識性能となった。近年の食事画像認識の研究 [6,19,20] で Fisher Vector を用いた食事画像認識は行われていないが、食事画像認識において色特徴を Fisher Vector で表現すること

により認識性能が大きく向上することが示された。そして、コストが非常に高い [6] の結果と同等以上の性能であった。従って、提案手法の有効性が示され、スマートフォン上で高精度に画像認識可能であることが示された。

5.3 認識時間

次に、認識時間を計測するために、20 回実験を行った。平均認識時間について実験結果を表 2 に示した。比較として、従来の認識時間の実験値 [1] を示した。従来は認識対象 50 種類に対して 0.26 秒であった。提案手法では 100 種類の食事に対して 0.065 秒とより高速に認識可能であった。従って、高速な認識にも適していることが示された。

5.4 使用メモリ量

メモリ面に関して、従来の手法では、SURF-BoF の次元数が 1500 次元、色ヒストグラムの次元数が 1728 次元に対し、本論文では HOG Patch-FV、Color Patch-FV ともに 1536 次元である。特徴ベクトルは密であるが、次元数はほぼ同じである。そして、高速に Fisher Vector にエンコードするため多数メモリにロードしているが、それらの合計は BoF のコードブックと比較し HOG Patch の場合 5 分の 1 以下、Color Patch の場合で 4 分の 1 以下のメモリ量である。実装したアプリケーションの Java ヒープ (主に画像処理系以外) は約 16MB、ネイティブヒープ (主に画像処理系) は約 3MB であった。低メモリを実現している。さらに特徴ベクトルの高次元化や認識対象数を増加可能であることがわかる。

6. まとめ

スマートフォン上でより高速かつ高精度に画像認識を行う手法として、局所特徴量に HOG Patch と Color Patch 特徴を用い、Fisher Vector で表現し、線形識別器で高速に分類する手法を提案した。そして、従来アプリケーションの画像認識の面において改良をした。100 種類の食事に対して正しい食事領域が与えられた時、候補を 5 つ提示した際に 79.2%の認識精度であった。従来手法と比較して上位 5 位以内で 11.0%の精度向上である。また、サーバサイドのコストが非常に高い認識手法 [6] 以上の性能が示された。そして、認識時間は、100 種類の認識対象に対して 0.065 秒であった。それは、認識対象が 50 種類であった従来の 0.26 秒と比較して 75.0%高速である。実験により提案手法の有効性と、スマートフォン上で高速かつ高精度物体認識が可能であることを示した。

今後は以下のことを考えている。クラウドソーシングを利用し画像にアノテーションをすることで認識対象を増やし、スマートフォン上で大規模画像認識を行う。また、認識する領域は手動で与えているため、高速な物体検出を行い、認識する領域を自動で決定する。

我々が提案した手法を実装した画像認識を用いた食事記録システムは以下で公開している。http://foodcam.jp.

文献

[1] Y. Kawano and K. Yanai. Real-time mobile food recognition system. In *Proc. of IEEE CVPR International Workshop on Mobile Vision (IWMV)*, 2013.

[2] 河野憲之, 柳井啓司. 食事認識を用いたモバイル食事管理システム. 第 5 回データ工学と情報マネジメントに関するフォーラム (DEIM), 2012.

[3] F. Perronnin and C. Dance. Fisher kernels on visual vocabularies for image categorization. In *Proc. of IEEE Computer Vision and Pattern Recognition*, 2007.

[4] F. Perronnin, J. Sánchez, and T. Mensink. Improving the fisher kernel for large-scale image classification. In *Proc. of European Conference on Computer Vision*, 2010.

[5] K. Chatfield, V. Lempitsky, A. Vedaldi, and A. Zisserman. The devil is in the details: an evaluation of recent feature encoding methods. In *Proc. of British Machine Vision Conference*, 2011.

[6] Y. Matsuda, H. Hoashi, and K. Yanai. Recognition of multiple-food images by detecting candidate regions. In *Proc. of IEEE International Conference on Multimedia and Expo*, 2012.

[7] K. Kitamura, T. Yamasaki, and K. Aizawa. Foodlog: Capture, analysis and retrieval of personal food images via web. In *Proc. of ACM Multimedia Workshop on Multimedia for Cooking and Eating Activities*, pp. 23–30, 2009.

[8] A. Mariappan, M. Bosch, F. Zhu, C.J. Boushey, D.A. Kerr, D.S̃. Ebert, and E.J. Delp. Personal dietary assessment using mobile devices. In *Proc. of the IS&T/SPIE Conference on Computational Imaging VII*, Vol. 7246, pp. 72460Z–1–72460Z–12, 2009.

[9] N. Kumar, P. Belhumeur, A. Biswas, D. Jacobs, W. Kress, I. Lopez, and J. Soares. Leafsnap: A computer vision system for automatic plant species identification. In *Proc. of European Conference on Computer Vision*, 2012.

[10] T. Lee and S. Soatto. Learning and matching multiscale template descriptors for real-time detection, localization and tracking. In *Proc. of IEEE Computer Vision and Pattern Recognition*, 2011.

[11] T. Maruyama, Y. Kawano, and K. Yanai. Real-time mobile recipe recommendation system using food ingredient recognition. In *Proc. of ACM MM Workshop on Interactive Multimedia on Mobile and Portable Devices(IMMPD)*, 2012.

[12] H. Bay, A. Ess, T. Tuytelaars, and L.V. Gool. Speeded-up robust features (SURF). *Computer Vision and Image Understanding*, Vol. 110, No. 3, pp. 346–359, 2008.

[13] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman. Lost in quantization: Improving particular object retrieval in large scale image databases. In *Proc. of IEEE Computer Vision and Pattern Recognition*, 2008.

[14] A. Vedaldi and A. Zisserman. Efficient additive kernels via explicit feature maps. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2012.

[15] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, Vol. 60, No. 2, pp. 91–110, 2004.

[16] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *Proc. of IEEE Computer Vision and Pattern Recognition*, 2005.

[17] G. Csurka, C. Bray, C. Dance, and L. Fan. Visual categorization with bags of keypoints. In *Proc.of ECCV Workshop on Statistical Learning in Computer Vision (SLCV)*, pp. 59–74, 2004.

[18] R. E. Fan, K. W. Chang, C. J. Hsieh, X. R. Wang, and C. J. Lin. LIBLINEAR: A library for large linear classification. *The Journal of Machine Learning Research*, Vol. 9, pp. 1871–1874, 2008.

[19] M. Chen, Y. Yang, C. Ho, S. Wang, S. Liu, E. Chang, C. Yeh, and M. Ouhyoung. Automatic chinese food identification and quantity estimation. In *SIGGRAPH Asia 2012 Technical Briefs*, 2012.

[20] M. Bosch, F. Zhu, N. Khanna, C. J Boushey, and E. J Delp. Combining global and local features for food identification in dietary assessment. In *Proc. of IEEE International Conference on Image Processing*, 2011.