

# 物体認識技術を用いたモバイル物品管理システム

望月 宏史<sup>†</sup> 柳井 啓司<sup>†</sup>

<sup>†</sup> 電気通信大学 電気通信学部 情報工学科 〒182-8585 東京都調布市調布ヶ丘 1-5-1

E-mail: †mochizu-h@mm.cs.uec.ac.jp, ††yanai@cs.uec.ac.jp

あらまし 本論文では、物品管理サービスにおいて、特定物体認識技術を利用することにより簡単に物品を検索する手法を提案する。そのためにスマートフォン上で動作するモバイル物品管理システムを作成した。このシステムは、ユーザが所持品をスマートフォンのカメラで撮影し、その物体についてのテキスト情報を付加して登録する。その後同じ物体を再度撮影すると、登録された物体と撮影中の物体が同一であるかを判定する。その結果が同一物体であれば、付加したテキスト情報をリアルタイムで表示することが出来るというシステムである。また、ネットワークを通じて他のユーザが登録した物品の情報を共有することが出来る。これにより、モノを介した新しいコミュニケーションを提案する。従来の所持品管理サービスでは、テキストやカテゴリによる検索、あるいはバーコードを読み取って登録するものが主流であった。それらに対して、本手法ではカメラに写すだけで所持品を検索することが出来る、バーコードの無い所持品も登録することが出来るという利点がある。本研究による利便性の向上を調べるために、物品認識を用いずに検索を行うベースライン機能を作成し、認識を用いた場合との比較評価を行った。

キーワード [モバイル],[特定物体認識]

Hirofumi MOCHIZUKI<sup>†</sup> and Keiji YANAI<sup>†</sup>

<sup>†</sup> Department of Information, The University of Electro-Communications

1-5-1 Chofugaoka, Chofu, Tokyo 182-8585 Japan

E-mail: †mochizu-h@mm.cs.uec.ac.jp, ††yanai@cs.uec.ac.jp

## 1. 背景と目的

近年、スマートフォンが普及し多くの人々が利用するようになった。また、それに伴いスマートフォンの性能も年々向上している。これにより多くのスマートフォン向けアプリケーション開発が行われるようになり、これまでの従来型の携帯電話では難しかった複雑で時間のかかる処理も可能となった。さらに、画像処理ライブラリである OpenCV がバージョン 2.3 より大きな変更があり、スマートフォン OS のひとつである Android での利用がしやすいものとなった。これにより、Android のカメラを使った物体認識アプリケーションの作成が可能になった。

また、身の回りに多くのものが溢れるようになった現代では同じ商品を 2 度購入してしまうという問題や、それをいつどうして買ったのか、使ってみて評価がどうであったかといった情報を忘れてしまうという問題が発生している。こうした問題を解決するために、所持品を管理するサービスが存在している。たとえばモノをコレクションするはてなモノリス [1] や、蔵書管理サービスであるブックログ [2] などである。しかし、これら従来の所持品管理サービスはテキストやカテゴリによる検索、あるいはバーコードを読み取って登録するものが主流である。

そこで本論文では、物品管理サービスにおいて、特定物体認識技術を利用することにより簡単に物品を検索する手法を提案する。物品に対してカメラを向けるだけで簡単に物品情報を検索することが出来るモバイル物品管理システムを作成した。このシステムはユーザが所持品をスマートフォンのカメラで撮影し、物品名・日時・位置情報・物品についてのコメントなどのテキスト情報を付加して登録する。その後、同じ物品を再度撮影すると、登録された物品と撮影中の物品が同一であるかを判定し、同一であれば付加したテキスト情報をリアルタイムで表示することが出来るというシステムである。また、ネットワークを通じて他のユーザが登録した物品の情報を共有することが出来る。これにより、モノを介した新しいコミュニケーションが可能となった。スクリーンショットを図 1 に示す。従来の物品管理サービスにおける物品検索に対して、本手法は

- カメラに写すだけで所持品をリアルタイムで高速に検索可能
  - 検索時にテキスト入力が必要ない
  - バーコードの無い所持品も登録可能
- という利点がある。

また、本研究による利便性の向上を調べるために、物品認識

を用いずに検索を行うベースライン機能を作成し、認識を用いた場合との比較評価を行った。

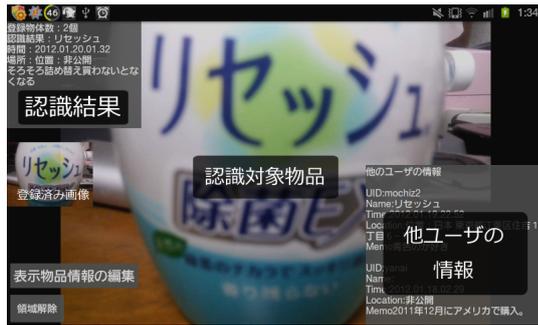


図 1 提案システム

## 2. 関連研究

### 2.1 スマートフォンと画像認識

スマートフォン上での特定物体認識アプリケーションでは、Google の Google Goggle [3] がある。このアプリでは、カメラで撮影した文字情報、著名な建造物や美術作品、本や DVD、バーコードなどの情報を認識して提供された情報を得ることが出来る。また、これら以外のもでも Google の画像による類似画像検索機能により商品の名前を取得することが出来る。これは Google の持つ巨大なデータベースとサーバを用いているので、非常に高い精度で物体を特定することが出来る。たとえば図 2 に示すように有名な絵画の認識を行うことが出来る。しかし、自分で物体を新規に登録することや、認識した結果を他のことに利用することは出来ない。

その他にスマートフォンでの画像認識の研究では、Takacs らによるスマートフォンの GPS による位置情報を用いて撮影した建物画像をデータベース中から検索する方法を提案した研究 [4] がある。位置情報を用いることによって比較する対象の画像を少なくすることが可能となり高速化を実現した。

これらの研究では物体認識にネットワークの利用が不可欠であるが、本研究では他ユーザとの情報同期のみにネットワークを利用し、物体認識処理はスマートフォンのみで行うことが出来る。

また、スマートフォンでの画像認識を行う場合では認識処理の軽さとメモリ消費量を抑えることが重要となる。近年のスマートフォンは CPU の高速化およびメモリの大容量化が進んでいるが、従来の画像認識処理で使われる計算機に比べればハードウェア資源に乏しい。そのため、従来手法よりも速度を重視する必要がある。たとえば Ta らは携帯電話で AR を行うために、物体追跡手法である SURFTrac [5] を提案した。これは、連続した画像の認識やビデオのトラッキングのための効率的なアルゴリズムであり、SURF の特徴点を検出する空間を注目空間のみに減らすことで高速化を実現した。これにより、実験で NOKIA の携帯電話 (CPU:330Mhz) において 6-7FPS で物体を追跡することを示した。

さらに近年では、SIFT や SURF よりも高速かつ低消費メ

モリな局所特徴として Leutenegger らの提案した BRISK [6]、Rublee らの提案した ORB [7] がある。これらは特徴量の記述がバイナリになっており、AR やリアルタイム処理などでの利用を想定している。本研究では特定物体認識に OpenCV に実装されている ORB を用いている。

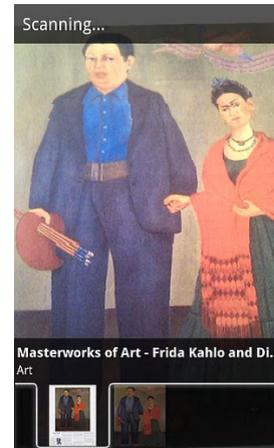


図 2 Google Goggle による絵画の認識

### 2.2 物品管理サービス

物品管理サービスとして有名なものには、はてなモノリス [1] がある。はてなモノリスは、スマートフォンを利用してバーコードをスキャンし物品名を調べ、その物品を自分のモノとしてコレクション出来るサービスである。また、モノについてコメントを投稿することが出来る。登録した物品は図 3 に示したように表示され、モノを持っているユーザのコメントを見ることが出来る。はてなモノリスでの登録作業の流れは以下のようになる。

- (1) バーコードを撮る
- (2) バーコードから物品情報が取得される
- (3) コメントを入力
- (4) コレクションに登録

バーコードを用いているので物品の認識は正確であるが、バーコードがついていない物品を登録することは出来ない。また、登録したモノを探そうとする場合は、コレクション一覧の中から探して選ぶことになる。そのため、登録した物品の数が多くなった場合に探し出すのが難しくなると考えられる。しかし、提案システムではバーコードではなく物品の画像特徴を利用するので、バーコードがない物品でも自由に登録することが出来る。また、提案システムでは物品にカメラをかざすだけで検索することが出来るので、リストから選ぶよりも早く見つけ出すことが期待出来る。

## 3. 提案システム詳細

提案システムの全体図を図 4 に示した。システムのおおまかな流れは、まず画像描画スレッドからカメラから取得した画像と選択領域を取得する。物品登録については、特定物体認識により登録済みかを確認した上で登録画面を表示する。認識スレッドについては、取得した画像と領域から特定物体認識を行



図 3 はてなモノリス

う．その結果をテキスト描画スレッドで画面に表示する．この章では，各処理について詳細を記述する．

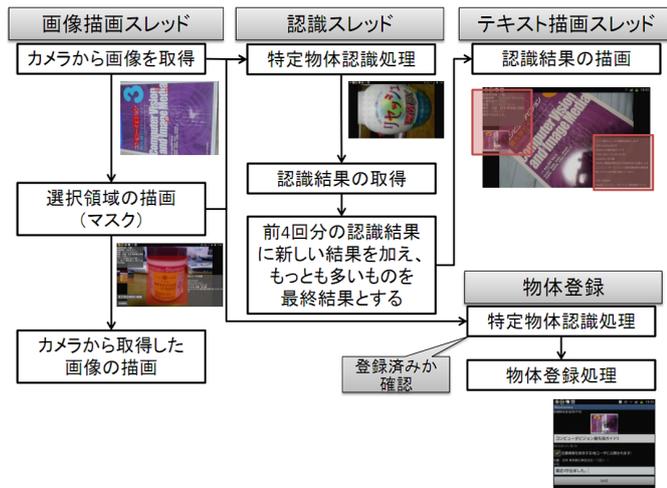


図 4 提案システム全体の流れ

### 3.1 特定物体認識

特定物体認識とは，画像から抽出する局所特徴量を比較することで，画像中に同一の物体が写っているかを判定する手法である．特定物体認識は，認識する対象に多くの形状がある一般物体認識と違い，形が定まっているので認識しやすいとされている．例として“携帯電話”の認識を考えると，図 5 に示すように，一般物体認識では携帯電話全般の認識を目的とするのに対し，特定物体認識では“iPhone”のような特定の機種を認識することを目的とする．本論文では“登録した物品”を特定物体として特定物体認識技術を利用する．



図 5 特定物体認識と一般物体認識の例

### 3.1.1 局所特徴量

局所特徴とは，画像の特徴的な点に注目し，注目点周辺の情報を特徴ベクトルとして表現したものである．有名なものとして，SIFT(Scale Invariant Feature Transform) [8] や，SIFTを高速化した SURF(Speeded-Up Robust Features) [9] などがある．しかし，これらの特徴量をハードウェア資源の貧しいスマートフォン上で計算するのは時間がかかる．そこで，より動作の早い特徴量として ORB [7] を用いた．

### 3.1.2 ORB

ORB(Oriented FAST and Rotated Brief) [7] とは FAST [10] による特徴点検出と BRIEF 記述子 [11] をもとに提案された 256bit のバイナリ特徴量である．BRIEF に比べ弱点であった回転不変に対する不変性を改善している．また特徴点の検出と記述子の計算はともに高速であり，SURF よりも 10 倍以上 SIFT よりも 100 倍以上の速度での計算を可能にしている．

### 3.1.3 本実験で用いる特徴点の検出

本研究では，特徴点の検出に ORB を用いて空間ピラミッドから検出する方法と一定間隔ごとに特徴点をとる DENSE を用いた．空間ピラミッドは 3 段階分を利用している．DENSE については，画像サイズが機種によって異なるので，固定された幅を使うと解像度の高い機種で点が多くなり，低い機種で少なくなってしまう．そこで，画像の横幅を  $x$  ピクセルとして  $\frac{x}{10}$  ピクセルごとに特徴点をとることで機種ごとの点の数の大きな変化をなくす．各格子点から得た点は，2 つのスケールを特徴点として利用する．DENSE を加えることで PyramidORB による特徴点検出で特徴点の出にくい物品も登録できるようになるが，背景の特徴の影響を受けやすくなってしまふ．それを防ぐために DENSE による特徴点検出の割合は PyramidORB よりも低く抑えている．

また，背景による影響を受けにくくするために，タッチパネルをドラッグすることで四角形の領域を描けるようにし，その領域からのみ特徴点を検出することが出来るようにした．図 6 に示すように，背景に物体が多く特徴が対象物体以外からも検出されてしまいそうな場合でも，対象物体からのみ特徴点を抽出することが出来る．



図 6 特徴点抽出領域の選択

### 3.1.4 特徴点の投票による認識

特定物体認識は，与えられた画像から検出した特徴点について投票を行い，その結果から認識結果を決める．投票方法は，

図 7 に示したように登録済み物体の全特徴点とマッチングを行い、最も近い点を持つ登録済み物体に対して投票をする。このとき、最も近い点との距離が閾値以上であった場合は似た点かどの登録済み物体にも存在しなかったとみなし、投票は行わない。これを全特徴点に対して行くと、与えられた画像の持つ特徴と似た特徴を持つ物体に投票が集まる。この投票が閾値以上であれば、与えられた画像に写っている物体と、投票数の最も多かった登録済み物体を同一物体として認識する。閾値以下だった場合は、同じ物体が存在しなかったとみなす。閾値については、値を大きくした方が物体の認識率は上がるが、そのかわりに登録した物品が写っていない状態でも何か物品があると誤認識してしまう率が高まってしまう。本システムでは実際に物品を撮影した時のデータを見て、正しい結果となるように閾値を定めた。

本論文で行う特定物体認識処理の流れを図 8 に示す。物品への投票は自分の登録した物品だけではなく、他ユーザの登録した物品に対しても行う。

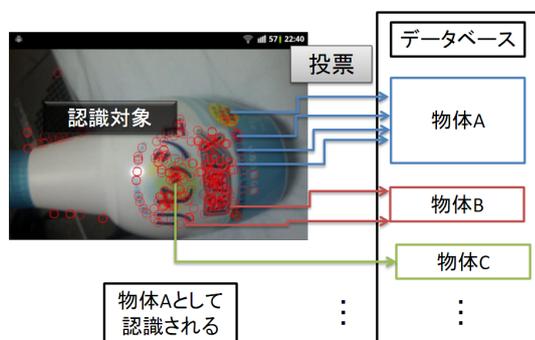


図 7 特徴点の物体への投票

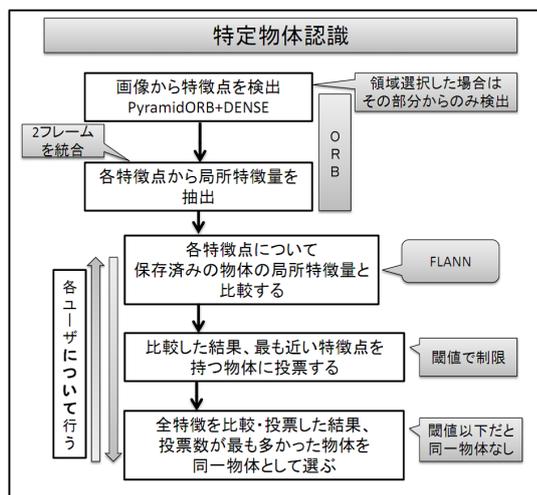


図 8 本システムでの特定物体認識

### 3.1.5 複数フレームの利用

当初は、2つのフレームから特徴を抽出し両フレームから同じように検出される点のみを利用することを考えていた。これによってより検出されやすい点を選ぶことが出来ると考えたからである。しかし実際には両フレームの点のマッチング処理に

余分な時間がかかるうえに、認識結果もあまり良くなかった。そこで各フレームから ORB を 180, DENSE を 20 ずつランダムで選び、これを 2 フレーム分の 400 点を用いて認識を行うこととした。2 フレーム分の点を検出すると、同じ物品であっても、角度や大きさの違いによって 2 フレーム目では 1 フレーム目と異なる点が発見されると期待される。これにより処理速度と認識結果の改善が見られたため、ランダムで選ぶこととした。

また、認識結果をそのまま用いるのではなく、5 回分の認識の投票を 3 位まで考慮する。5 回分のそれぞれの物品の認識投票数を足し合わせ、その中で最も投票数が多かったものを認識物品として選ぶ。たとえば投票数が

- (1) 物品 B(40) 物品 A(35) 物品 C(0)
- (2) 物品 B(33) 物品 A(30) 物品 C(10)
- (3) 物品 A(34) 物品 C(20) 物品 B(19)
- (4) 物品 C(20) 物品 A(18) 物品 B(16)
- (5) 物品 A(40) 物品 B(20) 物品 C(18)

となった場合は、

- 物品 A(35+30+34+18+40=157)
- 物品 B(40+33+19+16+20=128)
- 物品 C(0+10+20+20+18=68)

となり、物品 A が選ばれることとなる。こうすることにより、1 位だけの多数決をとる場合に比べ、わずかな投票差で 2 位になった場合と多くの投票差で 2 位になった場合との違いを考慮することが出来る。これにより認識精度が 20%ほど向上した。

## 3.2 物品管理

### 3.2.1 物品への情報付加

物品の管理は、写真(サムネイル)・テキストデータ・局所特徴量を登録する。物品へ付加するテキスト情報を以下に示す。

- 物品名
- 物品登録日
- 物品撮影場所
- 登録者の ID
- 物品に対するコメント・情報など自由記入欄

物品登録日・物品撮影場所については、システム側で自動で入力する。撮影場所については、他のユーザと共有するにあたってプライバシーの問題があるので、保存するかどうかは任意で選べるようにした。位置情報は Android の GPS が基地局情報による位置情報から緯度経度を取得し、逆ジオコーディングで地域名を取得する。これらの情報をテキストとして Android のストレージに保存し、認識時に表示させる。

### 3.2.2 ユーザ間での物品情報共有

自分が保存した物品情報を記述したファイルをサーバに送信して Android のストレージとサーバのファイルを同期させる。また、サーバに保存されている他ユーザの物品情報を記述したファイルも同期させる。同期については更新されるたびに自動で行われるようにするのが望ましいが、通信するファイル量が多いので、ユーザのボタン操作によって同期を開始するようにした。サーバと Android 間のファイルのやりとりについては、HTTP 通信の POST と GET で行う。他ユーザの物品についても認識スレッドで別途認識を行っており、その結果を図 1 に

示したように、画面中の右下に表示する。

#### 4. システムの使用例

実際にどのようにシステムを使うのか例を図9から図12で示す。ここでは例として、Twitterプログラミング本の登録と認識を行う。登録対象の本を図9に示す。

まず、図10に示したように登録ボタンを押して物品を撮影する。その後図11のようにテキスト情報の入力を行う。ここでは“TwitterAPI”を物品名とし、物品の情報として“Twitterプログラミングの参考に”というコメントを付与している。認識モードに切り替え、登録済み物品を写すと物体認識が行われ、図12に示すように左上に認識結果のテキスト情報が表示されるという仕組みである。ここでは先程入力した“TwitterAPI”が表示されているのがわかる。



図9 システムの動作例 登録対象



図10 システムの動作例 登録ボタン



図11 システムの動作例 テキスト入力



図12 システムの動作例 認識の様子

#### 5. 評価

持ち物管理サービスに特定物体認識を用いたことによる利便性が向上したかを調べるために、評価実験を行った。評価実験を行うにあたって、比較として認識を用いずに物品を探すベースライン機能を追加した。スクリーンショットを図13に示す。

- 年月
- 登録者のID
- 位置情報
- 物品名
- 物品に対するコメント・情報など自由記入欄から絞り込み検索を行うことが出来る。



図13 テキスト検索による登録物品検索システム

##### 5.1 評価項目

あらかじめ30物品をテストデータとして登録した。テストデータの画像一覧を図14に示す。さらに追加で各自10物品を追加してもらい、計40物品の状態での評価を行った。評価内容は、5人に以下の項目について評価してもらった。物品認識とベースラインとで、

- 検索にかかる時間
- 使いやすさの5段階評価

を比較し評価する。

##### 5.2 評価結果

検索にかかる時間の結果を表1に示した。認識を用いた場合に13.6秒、ベースラインで10.6秒となりベースラインのほうがやや早い結果となった。

また、検索の使いやすさを、1が使いにくい、5が使いやすいとした5段階で評価もらった結果を表2に示した。認識の使いやすさの平均が3.4、ベースラインの使いやすさの平均が3.2となり、認識のほうがやや良い結果となった。



図 14 あらかじめ登録したテストデータの画像一覧



図 15 登録した 5 物品

表 1 検索にかかった時間 (秒)

被験者 A (HTC DesireHD)						平均
認識	9	14	22	13	24	16.4
ベースライン	11	8	7	7	14	9.4
被験者 B (GALAPAGOS 003SH)						平均
認識	16	7	8	47	50	25.6
ベースライン	10	10	13	11	11	11
被験者 C (IS11T)						平均
認識	7	10	12	12	17	11.6
ベースライン	8	21	9	11	10	11.8
被験者 D (GalaxySII)						平均
認識	3	4	4	5	5	4.2
ベースライン	19	8	12	11	7	11.4
被験者 E (HTC EVO)						平均
認識	21	12	16	14	11	14.8
ベースライン	18	7	13	7	7	10.4
認識平均時間						14.52
ベースライン平均時間						10.8

表 2 使いやすさの 5 段階評価

	A	B	C	D	E	平均
認識	2	3	4	4	4	3.4
ベースライン	3	4	2	2	5	3.2

## 6. 認識精度の計測

### 6.1 認識精度の評価実験

実際にどの程度の割合で正しく認識が行えるかを調べるために、評価に用いたテストデータ 30 物品が登録されている状態で 5 物品を追加した。その 5 物品について認識を 50 フレーム行い、正しい認識結果が得られた割合を調べた。5 物品は

- (1) 雑誌
- (2) コーヒー
- (3) 筆箱
- (4) 消しゴム
- (5) 携帯

を登録した。画像を図 15 に示す。3.1.5 節で述べた複数フレームの結果から多数決をとった場合と、1 枚ずつ認識した結果の場合との両方を調べた。また、特徴点の投票において 1 位ではなかったが 2 位、3 位以内に正しい結果が出る場合についても調べた。

## 6.2 結果

認識精度の計測結果を図 16 に示した。複数フレームの結果から多数決をとった場合とそのまの場合を見ると、複数フレームの多数決をとったほうが精度が大きく向上していることがわかった。

物品については、認識しやすい物品とまったく認識できない物品とがあった。雑誌、コーヒー、筆箱、消しゴムはロゴや絵などで特徴的な部分が多いため、特徴点の検出がしやすく認識の精度も良かったと考えられる。一方、携帯電話は色が単一で特徴的な部分が少なく、光沢を持つため光の影響を受けやすい。こうした理由から、認識がうまくいかなかったのではないかと考えられる。

次に、複数フレームを利用した場合に、1 位に選ばれた物品だけでなく 3 位までを正解として含めた場合の結果を図 17 に示した。3 位まで含めたほうが認識精度があがっているのが、1 位が不正解であった場合でも、正解の物品にある程度の投票が行われていることがわかる。特に携帯について 1 位にはまったく現れなかったが、3 位まで含めると 40 % は認識に成功していることがわかった。

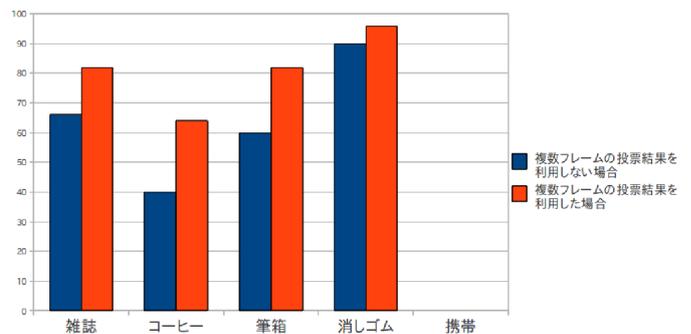


図 16 認識に成功した割合 (フレーム利用による違い)

## 7. 考察

### 7.1 評価実験について

評価結果は認識を用いるよりもベースラインのほうが早く検索出来たという結果になった。特定物体認識が遅くなってしまった原因は、特定物体認識によって認識しにくい物品があり、それらを検索する場合に時間がかかってしまい平均を下げてあげられる。認識しにくい物品が出てしまうのは、特徴点のとりにくい物品であったことや、背景の影響を受けてし

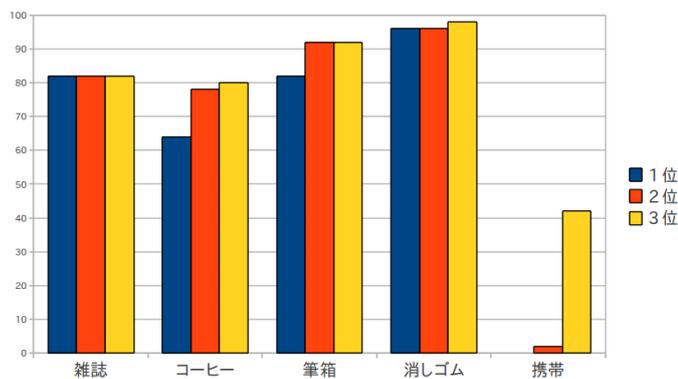


図 17 認識に成功した割合 (3位まで考慮した場合)

まったことが考えられる。また、登録物品が少なかったため、ベースラインの手動選択でも十分に選ぶことが出来てしまったことが考えられる。

検索しやすさの5段階評価については、認識を用いるほうが良いという結果となった。リストから選ぶ場合やテキストを入力するよりもカメラで撮るといった動作の方が楽である、さらに物品数が増えてきたときにリストから選んで探すのは難しくなるだろうなどの意見があった。

## 7.2 認識について

他人のスマートフォンで登録したものを自分のスマートフォンでも同様に認識させることはうまくいかない場合が多かった。これはユーザによって物品を撮影する撮り方や室内環境などの条件が異なることや、Androidのカメラ性能の違いによるところが大きいと思われる。図18に示したように、同じ環境下で撮影しても写真の撮れ方が異なっている。



図 18 Androidのカメラによる違い

自分で登録した物品については、「携帯」のように投票の結

果の1位だけを表示する場合にはまったく認識出来ない物品もあったが、3位まで考慮すればその中に含まれていた。35物品が登録されている状態であるので、3位以内に入っていればある程度は同じ点が検出され投票されていると考えられる。よって、1位だけでなく上位の物品をいくつか表示し、その中から選ぶ方式にすることで検索の簡便化という目的を果たせるのではないかと考えられる。たとえば100個の物品が登録されている場合を考えると、単純な登録順のリストから選択するよりも、認識結果の上位いくつかを提示し、その中から選択してもらう方がより短い時間で登録物品を探すことが出来るだろう。

## 8. おわりに

本研究では、特定物体認識を用いた物品管理サービスを作成した。また、ハードウェア資源の貧しいスマートフォンでも高速な特定物体認識が可能であることを示した。評価実験により、物品管理サービスに物体認識技術を利用することで簡単に物品検索が出来るようになったことを示した。

今後の課題としては、物品数をさらに増やした場合の性能評価をすることである。物品数を増やすことによってベースラインシステムの検索時間は増加すると考えられるが、特定物体認識の精度も低下してしまうだろう。大規模なデータベースに適用した場合でも同じような評価結果となるか調べる必要がある。また、一部物品について認識しにくいものがあったので、認識精度の向上が必要である。そのためには、処理速度は遅くなるが精度のよいSURFなどの特徴の導入や、色情報の考慮、OCRによるテキスト情報の利用などが考えられる。

## 文 献

- [1] はてなモノリス. <http://mono.hatena.ne.jp/>.
- [2] ブクログ. <http://booklog.jp/>.
- [3] Google Goggle. <http://www.google.com/mobile/goggles/>.
- [4] G. Takacs, V. Chandrasekhar, N. Gelfand, Y. Xiong, W. C. Chen, T. Bismpiagiannis, R. Grzeszczuk, K. Pulli, and B. Girod. Outdoors augmented reality on mobile phone using loxel-based visual feature organization. In *Proc. of ACM International Conference on Multimedia Information Retrieval*, 2008.
- [5] D. N. Ta, W. C. Chen, N. Gelfand, and K. Pulli. SURFTrac: Efficient Tracking and Continuous Object Recognition using Local Feature Descriptors. In *Proc. of IEEE Computer Vision and Pattern Recognition*, pp. 20–25, 2009.
- [6] S. Leutenegger, M. Chli, and R. Y. Siegwart. BRISK: Binary Robust Invariant Scalable Keypoints. In *Proc. of IEEE International Conference on Computer Vision*, 2011.
- [7] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski. ORB: an efficient alternative to SIFT or SURF. In *Proc. of IEEE International Conference on Computer Vision*, 2011.
- [8] D. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, pp. 91–110, 2004.
- [9] H. Bay, A. Ess, T. Tuytelaars, and L. V. Gool. SURF: Speeded Up Robust Features. In *Computer Vision and Image Understanding*, pp. 346–359, 2008.
- [10] E. Rosten and T. Drummond. Machine learning for high-speed corner detection. *Proc. of European Conference on Computer Vision*, pp. 430–443, 2006.
- [11] M. Calonder, V. Lepetit, C. Strecha, and P. Fua. Brief: Binary robust independent elementary features. *Proc. of European Conference on Computer Vision*, pp. 778–792, 2010.