

# 一般物体認識における学習データセットのサイズ拡張による影響の解析

外山託海<sup>†</sup> 黄瀬浩一<sup>†</sup>

<sup>†</sup> 大阪府立大学大学院工学研究科  
〒599-8531 大阪府堺市中区学園町 1-1

E-mail: †takumi@m.cs.osakafu-u.ac.jp, ††kise@cs.osakafu-u.ac.jp

**あらまし** 一般にパターン認識において、学習データセットの内容は識別性能に大きく影響することが知られている。これまで一般物体認識の分野において用いられてきた多くのデータセットでは、1カテゴリあたりのサンプル数が幾分か少ない場合が多いため、データセットを拡張して内容を充実させることができれば認識性能をさらに上げることができると考えられる。これまでそのようなサンプル数を増やすための手法はいくつか提案されてはいるが、それらの手法で用いられるデータの数は数万程度であり、さらにデータを増やすことで認識性能はより向上すると考えられる。そこで我々はインターネットから数十万枚の画像を集め、それらの画像をフィルタリングしてデータセットに加え、データセットのサイズを大きくした場合とそのままの大きさの場合とで認識性能を比べる。その際識別器を変えた上でも実験を行い、それぞれどのような影響を及ぼすかを解析する。

**キーワード** 一般物体認識, データセット拡張, フィルタリング, Support Vector Machine,  $k$ -Nearest Neighbor

## An Analysis of the Impact of a Training Dataset Expansion for Generic Object Recognition

Takumi TOYAMA<sup>†</sup> and Koichi KISE<sup>†</sup>

<sup>†</sup> Graduate School of Engineering, Osaka Prefecture University  
1-1 Gakuencho, Naka, Sakai, Osaka, 599-8531 Japan

E-mail: †takumi@m.cs.osakafu-u.ac.jp, ††kise@cs.osakafu-u.ac.jp

**Abstract** In the field of pattern recognition, it is well known that the contents of a training dataset affects the recognition performance. Most datasets used in the field of generic object recognition have a somewhat small number of samples per category. Thus, it is reasonable to think that we can acquire higher recognition performance by expanding the dataset. Although several approaches have attempted to increase the number of samples, the amount of these increased samples is about ten thousands which can be increased more to acquire much higher recognition performance. Thus, we collect hundred thousands of images and compare the recognition performance between the dataset with original size and the dataset which is expanded with our dataset expansion method. In addition, we also analyze the impact of using different classifiers for these expanded datasets.

**Key words** Generic object recognition, Dataset expansion, Filtering, Support Vector Machine,  $k$ -Nearest Neighbor

### 1. はじめに

パターン認識の研究において、学習に用いるデータセットが認識の結果に大きく影響を与えることが古くから報告されている。例えば、手書き文字の認識において学習データの大幅な増大による認識性能の向上を行った研究などがある [1]。また、データセット中に含まれるノイズを除去することにより認識性能を向上させた例もある [2]。

本稿では、画像中の物体のカテゴリ認識、すなわち一般物体認識に着目し、そのためのデータセットを扱う。一般物体認識のタスクにおいては、Caltech101 [5], Caltech256, PASCAL VOC Challenge, MSRC等のデータセットがリソース、ベンチマークとして広く用いられており、それぞれカテゴリごとに分類されたデータを用いて学習を行う。ところが、これらのデータセットにおいての各カテゴリのサンプル数には偏りがある事が問題として挙げられる。例えば Caltech101 では最も多いカ

テゴリ (airplane) では 800 枚の画像データがあるが、最も少ないカテゴリ (inline skate) では 31 枚しか用意されていない。一般物体認識というタスクでは、物体の外見はカテゴリ内でも大きく変動する。そういった変動の大きさから考えて、少ないサンプル数ではカテゴリ内すべての物体の見え方をカバーできるとは考えにくい。

そこで我々は、これらのデータセットにおいて、各カテゴリでのサンプル数を増加させ、カテゴリ内におけるより多くの物体の見え方をカバーし、認識率の向上を図る。サンプル数を増やすために、インターネットの画像検索を利用し、各カテゴリの物体が写っている画像を可能な限り集める。ところが、インターネットの一般的な画像検索で得られる画像は、必ずしも目的とする物体の写っている画像ではない事が多い。また写っていてもオクルージョンがある、全体像が写っていないという事も多いため、得られた結果をフィルタにかけ、認識に有効な画像だけを取り出すという操作が必要となる。取り出された画像をデータセットに加え、データセットの拡張とする。

インターネットからのデータ獲得に関連した内容では、インターネットからテキストデータ、画像データ両方を用いてデータを収集してくる研究 [3] や、得られるデータのフィルタリングのための研究 [4] などがある。ただし、それらの研究で用いられている画像の枚数は数万枚程度である。一般に、インターネットの画像検索で得られる結果のノイズの多さから考えて、数万枚のダウンロードしてきたデータからは、目的とする物体の写っている画像はせいぜい数千枚しか含まれていない。さらに、物体に対するオクルージョンのない画像、また物体の全体像が写っている画像という条件を加えると、数百枚程しか残らない。そういった状況下でより多くの必要とする画像、すなわち学習データセットに追加することで認識性能が向上するような画像を集めようとする、画像の絶対数をさらに増やすという事が必要となる。そこで我々は、それらの研究より多い数十万枚の画像を集め実験する。

本稿では、Caltech101 を対象にし、その拡張を行い認識実験を行った結果について述べる。実験を行った結果、拡張したデータセットでの学習は、オリジナルのデータセットでの学習と比較して、3~4% 認識率が向上した。

## 2. データセット

本研究では、[4] に倣い、Caltech101 から 21 カテゴリを選び、それらのデータに対して認識実験を行う。選ぶカテゴリを表 1 に示し、それぞれ含まれているデータ数も示す。表に示すように、inline skate が最もサンプル数が少なく、31 枚の画像があるのみである。一方、最も多い画像数を持つカテゴリは Faces で画像数は 435 枚である。また、Faces, panda, inline skate の画像例を図 1 に示す。これらの画像例が示すように、panda のカテゴリでは、カテゴリ内で見え方の大きく異なる画像が与えられている一方で、inline skate のカテゴリでは物体の向き、光の加減、背景等、画像間で似ているものが多い。このように Caltech101 にはカテゴリ間でデータの偏りがあり、カテゴリによって認識性能の差が大きく表れる [6].

表 1 Caltech101 より選んだ 21 カテゴリのデータ数とそれぞれのカテゴリでインターネットからダウンロードしてきた画像数

|              | Caltech101 の画像数 | インターネットから得た画像数 |
|--------------|-----------------|----------------|
| Faces        | 435             | 15987          |
| accordion    | 55              | 16004          |
| bonsai       | 128             | 15986          |
| euphonium    | 64              | 4466           |
| grand piano  | 99              | 5165           |
| inline skate | 31              | 7622           |
| laptop       | 81              | 15984          |
| menorah      | 87              | 11418          |
| nautilus     | 55              | 12690          |
| pagoda       | 47              | 15992          |
| panda        | 38              | 15988          |
| pyramid      | 57              | 15976          |
| revolver     | 82              | 15624          |
| schooner     | 63              | 15998          |
| soccer ball  | 64              | 12202          |
| starfish     | 86              | 15991          |
| stop sign    | 64              | 14290          |
| strawberry   | 35              | 15964          |
| sunflower    | 85              | 15978          |
| umbrella     | 75              | 14973          |
| watch        | 239             | 15980          |

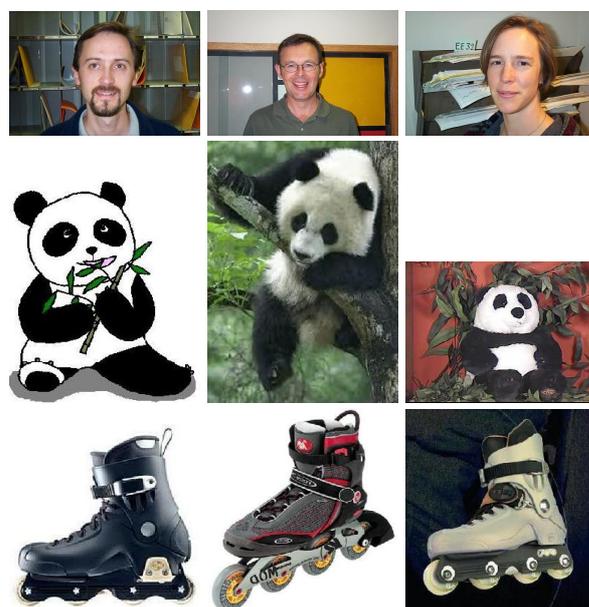


図 1 Caltech101 に含まれる画像の例。上段から順に Faces, panda, inline skate を示す。

## 3. データセット拡張

Caltech101 を用いた認識器の性能を上げるため、データの拡張を行う。まず、画像をインターネットからダウンロードし、集めた画像に対してフィルタリングを行い、データセットに加える。

### 3.1 ダウンロード

カテゴリ毎にサンプル数を増やすために、Caltech101 から選

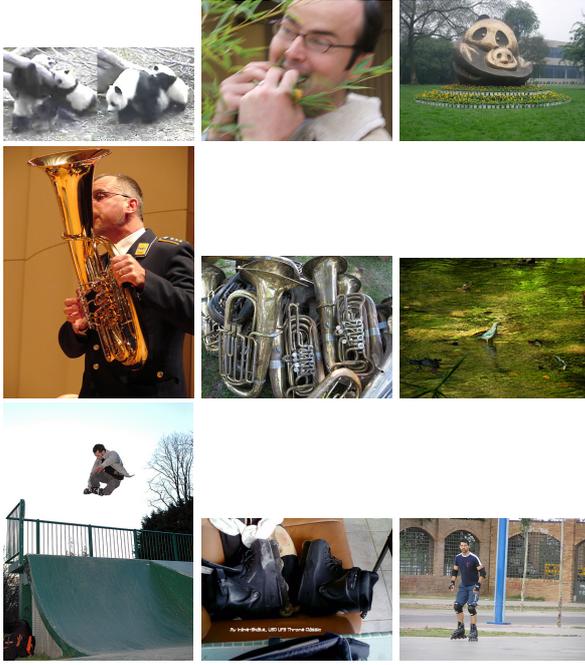


図 2 インターネットから集められた画像. 上段から順に panda, euphonium, inline skate を示す.

んだ 21 カテゴリそれぞれのカテゴリ名をキーワードとし、インターネットで画像検索を行い、キーワードのタグが付いた画像をダウンロードする.

実際にダウンロードして得られた枚数をカテゴリ毎に表 1 に示す. ダウンロードしてきた画像の合計枚数は 287,785 枚である. euphonium カテゴリの画像が最も少ない数となったが、それでも 5000 枚近くの画像があり、過去の研究 [4] と比べても、画像枚数は多い. また、集まった画像の例を図 2 に示す. これらの画像が示すように、タグをもとに集められた画像は、必ずしも目的の物体が写っている画像であるとは限らない. 今回集まった画像をランダムに選び、ダウンロードした画像のサブセットで目的とする物体の写っている画像とそれ以外のノイズとなる画像の割合を求めてみると、各カテゴリで平均して 30% 程度である. しかしこの割合にも幅があり、目的とする物体が写っている画像が多いカテゴリもあれば非常に少ないカテゴリもある. 特に図 2 からわかるように inline skate といったカテゴリは、物体としての inline skate の画像よりも、シーン、すなわちスポーツとしての inline skate の画像が多く存在している.

これらダウンロード画像に含まれる、目的とする物体の写っていないような画像を除去するために、これらのデータに対してフィルタをかけ、目的とする物体の写っている画像のみを取り出す.

### 3.2 フィルタリング

初めから与えられている Caltech101 のデータを用いて各カテゴリの検出器を学習させ、それをフィルタとして用いる. つまり、ダウンロードしたデータを Caltech101 のデータと比較し、特定のカテゴリを含む画像かどうかを決定する.

#### 3.2.1 検出器の学習

まず、検出器の学習について述べる.

本研究では、広くその実用性が知られている、SIFT(Scale-Invariant Feature Transform) [7] を用いた Bag-of-Features の手法を採用する.

SIFT は画像の特徴を記述するための手法の一つである<sup>(注1)</sup>. 画像の局所領域を予め抽出すれば、局所領域毎に SIFT 特徴量を得ることができる. 本研究では、局所領域抽出手法として Hessian-Affine detector [8] を用いる. SIFT 特徴量は 1 画像につきおよそ数十～数千個得ることができ、それらの特徴量を使って画像をベクトルで表現する. ここでは [9] で用いられている手法と同様に、カテゴリ毎に SIFT 特徴量をクラスタリングし、各クラスタを visual word として設定する. 画像の SIFT 特徴量がどの visual word に含まれているかをすべての特徴量で調べ、画像毎の Bag-of-Features 表現を得る. これにより、画像が 1 つのベクトルで表される.

画像 1 枚のベクトルの次元数はすなわち visual word の数に相当する. この次元数を変えることにより検出の性能は変化する. 本研究では、クラスタリング手法として k-means 法を用いるため、各カテゴリ毎に最適な visual word 数を実験的に求めることができる.

得られたベクトルを学習データとし、検出器の学習をする. 本研究では検出器として 2 値識別器である Support Vector Machine (SVM) [10] を採用する.

SVM では決定関数として以下の式を用いる.

$$g(\mathbf{x}) = \sum_i \alpha_i y_i K(\mathbf{x}_i, \mathbf{x}) - b$$

ここで  $\mathbf{x}_i$  は学習データ、 $\mathbf{x}$  はテストデータ、 $y_i$  は  $\mathbf{x}_i$  のカテゴリラベル (+1 or -1)、すなわち  $\mathbf{x}_i$  がカテゴリに属するか属さないかを表す. また、 $\alpha_i$  は学習データ  $\mathbf{x}_i$  に対する重み、 $b$  は閾値を表す. また、カーネル関数  $K(\mathbf{x}_i, \mathbf{x})$  にはガウシアンカーネルを用いる. 検出では、テストデータが特定のカテゴリに属するかどうかを調べる. そのため、学習においての +1 のラベルはその特定のカテゴリ、-1 のラベルを持つデータとして、Caltech101 に存在する BACKGROUND Google からランダムに選んだ画像を用いる.

#### 3.2.2 データセットへの追加

各カテゴリの SVM モデルの決定関数を用いて、ダウンロードしてきた画像  $\mathbf{x}$  に対する値、すなわち  $g(\mathbf{x})$  を求める. 一般的にはここで、 $g(\mathbf{x}) > 0$  であれば、目的とするカテゴリに属するという決定がなされるが、今回データセットに実際に追加するデータは  $g(\mathbf{x})$  の値が大きなものから順に選んでいく. なげなら、たとえ  $g(\mathbf{x}) > 0$  であっても、その値が 0 に近ければ、それは少しの条件変化で  $g(\mathbf{x}) < 0$  となる可能性を秘めており、不安定なデータと言えるからである.

本研究では、追加するデータ数を変えてデータ数と認識率の解析を行う. データ追加の際、本研究で用いるの手法は次の 2 つである.

A: 全カテゴリに対して一定の数だけデータを追加する.

(注1): SIFT と一般に呼ばれるものには、局所領域の抽出も含まれる場合があるが、本稿では SIFT は局所領域の抽出は含まないものとする.

B: オリジナルの学習データを用いて認識実験の交差検定を行い、カテゴリ毎に学習データ内での認識率を求め、その認識率が低いカテゴリにデータを追加する。

一般的に、うまく認識できないカテゴリには、認識できない部分のデータが不足していると考えられる。Bにはそのようなカテゴリに対して学習サンプルを加え、認識性能を強化するというねらいがある。最も直観的で単純な方法として、認識率の逆数を用いて追加する数を求めるという方法を採用する。具体的には事前に一般物体認識の交差検定を行い、各カテゴリの学習データにおける認識率を求め、それぞれの認識率を  $p$ 、各カテゴリのサンプル数を  $n$  とすると、B では  $\frac{1}{p}n$  枚の画像が新しくデータセットに追加される。

以上のフィルタを用いて、ダウンロードしてきた画像から、関係のない画像を除き、より関係のあるデータを取り出していく。取り出したデータをデータセットに追加して、認識実験を行う。

#### 4. 認識手法

データセットの拡張により、データの数が大きく変わる。すると、一般物体認識を行う際、認識手法によっては学習が困難になる場合がある。例えば、SVM は高い識別性能をもつが、学習には多くのコストが必要とされる。一方で  $k$ -NN 法 ( $k$ -Nearest Neighbor 法) といった単純な方法は、SVM のように高い認識性能を持つ識別器は簡単には学習できないものの、単純であるが故にコストの少ない識別手法であると言える。本研究では、認識手法として  $k$ -NN 法、そして SVM を用い拡張されたデータセットに対して、それぞれの性能を評価する。

どちらの場合もフィルタリングと同様に、画像は SIFT の Bag-of-Features 表現を用いる。ただし、フィルタリングと異なる点として、すべてのカテゴリのデータは同じ visual word を用いる必要があり、カテゴリ個別で visual word を設定することはできないことが挙げられる。

##### 4.1 $k$ -NN 法を用いる手法

$k$ -NN 法は事例に基づく学習手法であり、テストデータに対し  $k$  個の学習データからそれぞれのクラスに投票することにより認識がなされる。 $k$  個のデータはテストデータに対してユークリッド距離の近い学習データから順に選ばれる。また、投票結果は各カテゴリの学習サンプル数により正規化を行う。事例に基づく学習なので、サンプル数が増えれば多いほどより複雑な境界を描くことができる。よってデータ拡張によりデータを増やすことでさらに高い認識性能を得られると考えられる。

##### 4.2 SVM を用いる手法

カテゴリ認識においても、フィルタリングと同様に SVM を用いる。一つ異なる点は、複数のカテゴリの認識となるので、SVM を二値識別器から複数クラス識別器にするような方法が必要である。本研究では、Crammer と Singer により提案されている SVM を用いた複数クラスの認識手法の実装である BSVM [11] を用いる。

SVM でもサンプル数はある程度は必要であるが、 $k$ -NN 法と違いあまりサンプル数が増えすぎると、学習が収束しないとい

表 2 各カテゴリ毎の最も結果の良くなった visual word 数とそのときの検出器の Precision

|              | visual word 数 | Precision[%] |
|--------------|---------------|--------------|
| Faces        | 800           | 93.6         |
| accordion    | 800           | 86.7         |
| bonsai       | 600           | 88.4         |
| euphonium    | 600           | 85.5         |
| grand piano  | 1000          | 82.3         |
| inline skate | 600           | 91.3         |
| laptop       | 1000          | 81.5         |
| menorah      | 800           | 80.8         |
| nautilus     | 600           | 92.7         |
| pagoda       | 1000          | 91.9         |
| panda        | 800           | 96.3         |
| pyramid      | 400           | 77.5         |
| revolver     | 1000          | 84.3         |
| schooner     | 1000          | 89.4         |
| soccer ball  | 1000          | 91.1         |
| starfish     | 1000          | 84.1         |
| stop sign    | 1000          | 88.3         |
| strawberry   | 1000          | 80.0         |
| sunflower    | 1000          | 89.0         |
| umbrella     | 1000          | 75.8         |
| watch        | 800           | 80.3         |

う問題がでる。

### 5. 実験と結果の解析

#### 5.1 フィルタリング

まず、Caltech101 のデータからそれぞれのカテゴリの検出器を学習する。検出器の学習には SVM<sup>light</sup> [12] を用いた。visual word の数を 200 から 1000 まで 200 ずつ変えていき、それぞれのカテゴリ毎に最も結果の良い visual word の数を SVM<sup>light</sup> の XiAlpha-estimate による Precision の値から決定した。XiAlpha-estimate は Leave-One-Out estimate の簡易手法である。表 2 にそれぞれのカテゴリ別に最も結果の良くなった visual word の数とそのときの Precision を示す。これらの Precision はダウンロード画像のフィルタリングにおける検出性能を表すとも考えられる。

では、実際にこれらの検出器にダウンロードした画像を与え、得られた結果を図 3 に示す。ここでは、 $g(\mathbf{x}) > 0$  となった  $\mathbf{x}$  の数を、それぞれ  $0 < g(\mathbf{x}) < 0.5$ ,  $0.5 \leq g(\mathbf{x}) < 1.0$ ,  $1.0 \leq g(\mathbf{x}) < 1.5$ ,  $1.5 \leq g(\mathbf{x})$  の 4 つの範囲で分けて示す。この図から、ダウンロードしてきた画像には、カテゴリ毎に目的とする物体が写っているかどうかかなりばらつきがあることが分かる。euphonium, grand piano, inline skate は表 1 が示すように、もともとダウンロードされた画像数が少ないので当然他より  $g(\mathbf{x}) > 0$  となる画像の数は少ないが、それら以外にも、schooner, strawberry や umbrella といったカテゴリはダウンロードされた数は多いにも関わらず、フィルタリングをすると  $g(\mathbf{x}) > 0$  となる画像の数が少ないことがわかる。

ここで得られた  $g(\mathbf{x})$  の値を降順にソートし、値が大ききなも

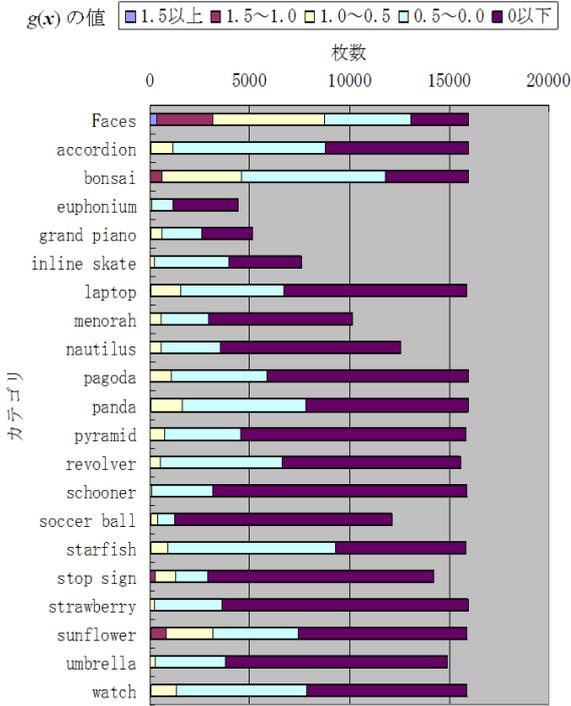


図 3 それぞれの  $g(\mathbf{x})$  の値を得た  $\mathbf{x}$ (画像) の枚数

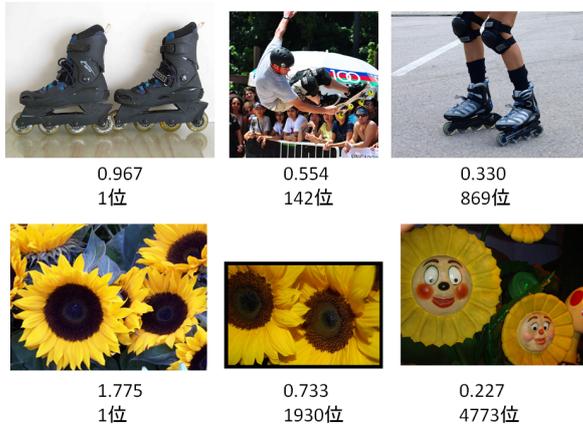


図 4 画像の例をフィルタリングでの  $g(\mathbf{x})$  の値とその順位とともに示す。上段が inline skate で下段が sunflower である。

のからデータセットに追加していく。図 4 にソートされた順位とともにいくつか画像を示す。ここでは例として inline skate, sunflower カテゴリの画像を示す。どちらも 1 位には目的とするカテゴリの画像が現れているが、inline skate は 100 位で早くもその物体ではない画像が現れており、100 位以下にまた目的とする物体の画像が位置した。一方、sunflower は 1000 位台においても目的とする物体の写った画像が現れており、ここにカテゴリ間での差異がはっきり表れている。

## 5.2 一般物体認識実験

ダウンロードとフィルタリングによって得られた画像をデータセットに加え、一般物体認識の実験を行った。ここではその

表 3 全ダウンロードデータを用いた  $k$ -NN 法での認識実験の結果

|         | $k = 1$   | $k = 5$   | $k = 10$    | $k = 100$   |
|---------|-----------|-----------|-------------|-------------|
| 認識率 [%] | 37.4      | 40.5      | 46.2        | 47.6        |
|         | $k = 300$ | $k = 500$ | $k = 1,000$ | $k = 1,500$ |
| 認識率 [%] | 50.6      | 51.0      | 52.3        | 51.4        |

結果について述べる。この実験において visual word 数は各カテゴリから 50 ずつ、すなわち 1,050 とした。これは以降すべての実験について用いられている。また、テスト画像として、予め学習データと分けておいた Caltech101 の画像を用いた。フィルタリング、visual word 作成の際のクラスタリングにはこれらのテストデータを一切用いていない。

まず、ベースラインとして拡張をしない状態での認識実験の結果を述べる。 $k$ -NN 法では  $k$  を変えて実験を行い、結果  $k = 5$  のときに 60.0 % となり、これが最も良い認識率であった。また、SVM を用いた手法では、62.4 % となり、拡張なしの場合では SVM を用いた手法が  $k$ -NN 法を上回った(注2)。

次に、ダウンロードした画像すべてをデータセットに加えた場合の  $k$ -NN 法での認識率を表 3 に示す。この場合、学習に用いた総画像数は 287,785 枚となり、これはオリジナルの 1,758 枚と比べると 150 倍近くある。オリジナルの Caltech101 のデータセットでの実験と異なり、この場合  $k = 1,000$  としたときに最も認識率が高くなった。ダウンロードした画像のノイズの影響により認識率は下がってはいるが、それでもまだ 50 % 近くある。SVM ではこのデータセットに対して、学習が収束しなかった。このことから  $k$ -NN 法が大規模なデータに対して SVM より有効であることがわかる。

画像を追加する方法を変えた場合の認識率を示す。その方法はそれぞれ、

- A: 全カテゴリに対して一定の数だけ画像を追加した場合
- B: オリジナルの学習データを用いて認識実験の交差検定を行い、カテゴリ毎の認識率の逆数の割合でデータを追加した場合である。

A では、追加する画像数を 20, 50, 80, 100, 150, 200, 300, 500 と変えて実験を行った。また、B においてまず Caltech101 のオリジナルの学習データで認識実験の交差検定を行い、その結果をもとに認識率の逆数の割合で画像を追加する。

A, B の認識実験の結果をそれぞれ表 4, 5 に示す。

A の実験結果から、SVM を用いた手法は概ねオリジナルのデータのみの方よりも高い認識性能を得られている事が分かる。一番認識率が高かったのは各カテゴリに 100 枚ずつ追加した場合で、その時の認識率は 66.7 % となった。また、一方で  $k$ -NN 法では 20 枚や 50 枚の追加では認識率は上がらず、逆にオリジナルのみデータセットでの認識率よりも下がる結果となった。これは、すでに認識にある程度成功しているカテゴリに対してもデータ拡張を行ったため、認識にあまり成功しなかったカテ

(注 2) : 本研究の目的はデータセットの拡張による認識性能の解析であるので、絶対的な認識率の向上を図っているわけではない。そのため、ここで示す実験の結果よりも高い認識率が、他の有効な手法、例えば、Multiple Kernel Learning 等を用いた手法等により達成可能である。

表 4 A:全カテゴリに対して一定の数だけ画像を追加した場合の認識率

|                                 | 各カテゴリに追加する画像数 |      |      |      |             |             |       |       |       |
|---------------------------------|---------------|------|------|------|-------------|-------------|-------|-------|-------|
|                                 | 0 枚 (オリジナルのみ) | 20 枚 | 50 枚 | 80 枚 | 100 枚       | 150 枚       | 200 枚 | 300 枚 | 500 枚 |
| SVM での認識率 [%]                   | 62.4          | 63.8 | 65.2 | 66.2 | <b>66.7</b> | 64.8        | 64.8  | 61.4  | 60.0  |
| $k$ -NN 法での認識率 ( $k = 3$ )[%]   | 59.5          | 56.7 | 59.5 | 58.6 | 58.1        | 57.1        | 59.5  | 60.0  | 56.2  |
| $k$ -NN 法での認識率 ( $k = 5$ )[%]   | 60.0          | 55.2 | 58.6 | 59.0 | 60.0        | 61.0        | 58.6  | 59.0  | 58.6  |
| $k$ -NN 法での認識率 ( $k = 10$ )[%]  | -             | 58.1 | 58.1 | 58.6 | 60.5        | <b>62.4</b> | 60.0  | 56.7  | 58.6  |
| $k$ -NN 法での認識率 ( $k = 30$ )[%]  | -             | 55.2 | 57.1 | 56.2 | 59.0        | 57.1        | 59.0  | 58.1  | 57.6  |
| $k$ -NN 法での認識率 ( $k = 100$ )[%] | -             | -    | -    | 54.3 | 55.7        | 53.8        | 52.9  | 54.8  | 55.7  |

表 5 B:カテゴリ毎に認識率の逆数の割合でデータ数を追加した場合の認識率

|                                |             |
|--------------------------------|-------------|
| SVM での認識率 [%]                  | <b>66.7</b> |
| $k$ -NN 法での認識率 ( $k = 3$ )[%]  | 59.5        |
| $k$ -NN 法での認識率 ( $k = 5$ )[%]  | 58.6        |
| $k$ -NN 法での認識率 ( $k = 10$ )[%] | 59.0        |
| $k$ -NN 法での認識率 ( $k = 30$ )[%] | <b>61.0</b> |

ゴリの票がさらに分散された事が理由である。認識率が高くなったのは 100 枚からで、150 枚の追加ではオリジナルを上回る 62.4 % の認識率を得た。これは、オリジナルのみのデータにおいて SVM で学習した場合とほぼ同じ認識率である。またさらに枚数を増やすと、認識率は下がった。これはカテゴリによっては、ダウンロードされた画像に正しいデータが含まれている確率に大きくばらつきがあり、数百枚以上のデータにもはや正しいデータがあまり入っていないカテゴリがいくつかあるためである。

次に、B の実験結果をみると、SVM を用いた手法では A の場合の最も高い認識率と同じ認識率を得られていることが分かる。A の最も高い認識率の時のデータ追加の画像枚数は 100 枚であり、B の  $k$ -NN 法の結果もその時の結果と似ている。一定数だけ追加した場合と比べて認識率が向上していないのは、認識できなかったカテゴリのテストデータを認識できるようにはなっているが、同時に今まで認識できていたテストデータが認識できなくなったという理由がある。

以上の実験結果から二つのことが分かる。まず第一に、適切なデータを用いてのデータセットの拡張が認識性能の向上につながるということである。追加するデータセットに多少ノイズが含まれていても、ある一定の割合を超えない範囲であれば、追加することにより認識性能は向上する。本実験ではその一定の割合を超えたのが 100~150 枚の時であり、それ以降はノイズの割合が大きくなり、認識率は下がる一方となる。この割合を超えないようにさらに認識率を上げるには、さらに多くのデータを用意しデータの絶対数を上げる必要がある。第二に、データを増やしても学習が収束するうちは SVM の認識性能は  $k$ -NN を凌駕するということである。ただし、SVM で学習が収束しないような量のデータに対しても対応できるという強みが  $k$ -NN にはあり、テストデータとしてさらに多様なものが与えられるような場合は、 $k$ -NN といった単純な識別機で大量のデータを処理する必要が出てくると思われる。

## 6. おわりに

我々は本研究において、一般物体認識におけるデータセットの拡張をし、その影響の解析をした。本稿で示した結果により、データセットの適切な拡張が、一般物体認識の性能向上のために有効であることが分かった。今後の課題として、さらに画像の枚数を増やして認識性能を検証することが挙げられる。

謝辞 本研究の一部は科学研究費補助助成金基盤研究(B)(19300062, 22300062)の補助による。

### 文 献

- [1] J. Cano, J.-C. Perez-Cortes, J. Arlandis and R. Llobet: "Training set expansion in handwritten character recognition", Structural, Syntactic and Statistical Pattern Recognition, pages 548- 556. LNCS 2396 (2002).
- [2] A. Angelova, Y. Abu-Mostafa and P. Perona: "Pruning training sets for learning of object categories", Vol. 1, Los Alamitos, CA, USA, IEEE Computer Society, pp. 494-501 (2005).
- [3] F. Schroff, A. Criminisi and A. Zisserman: "Harvesting image databases from the web", Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on, pp. 1-8 (2007).
- [4] K. Wnuk and S. Soatto: "Filtering internet image search results towards keyword based category recognition", pp. 1-8 (2008).
- [5] L. Fei-fei: "Learning generative visual models from few training examples: an incremental bayesian approach tested on 101 object categories" (2004).
- [6] J. Ponce, T. Berg, M. Everingham, D. Forsyth, M. Hebert, S. Lazebnik, M. Marszalek, C. Schmid, B. Russell, A. Torralba, C. Williams, J. Zhang and A. Zisserman: "Dataset issues in object recognition", pp. 29-48 (2006).
- [7] D.Lowe: "Distinctive image features from scale-invariant keypoints", International Journal of Computer Vision, Vol. 60, No. 2, pp. 91-110 (2004).
- [8] K. Mikolajczyk and C. Schmid: "An affine invariant interest point detector", Proceedings of the 8th International Conference on Computer Vision (2002).
- [9] J.Zhang, M.Marszalek, S.Lazebnik and C.Schmid: "Local features and kernels for classification of texture and object categories: A comprehensive study", International Journal of Computer Vision (2006).
- [10] Scholköpf.B and Smola.AJ: "Learning with Kernels: Support Vector Machines, Regularization, Optimization and Beyond", MIT Press.
- [11] C. wei Hsu and C.-J. Lin: "A comparison of methods for multi-class support vector machines" (2001).
- [12] T. Joachims: "SVM<sup>light</sup> Support Vector Machine", <http://www.cs.cornell.edu/People/tj/svm%5Fflight/>.