

グラフ-ベクトル変換を用いたグラフ構造表現による一般物体認識

堀 貴博[†] 滝口 哲也^{††} 有木 康雄^{††}

[†] 神戸大学大学院システム情報学研究科 〒 657-8501 兵庫県神戸市灘区六甲台町 1-1

^{††} 神戸大学自然科学系先端融合研究環 〒 657-8501 兵庫県神戸市灘区六甲台町 1-1

E-mail: [†]horitaka@me.cs.scitec.kobe-u.ac.jp, ^{††}{takigu, ariki}@kobe-u.ac.jp

あらまし コンピュータによる一般物体認識は、近年、ロボットビジョンや画像検索といった分野で、実現が強く求められている。従来研究には、画像から局所の特徴である SIFT(Scale-Invariant Feature Transform) を抽出し、クラスタリングにより出現頻度ヒストグラム表現に変換する Bag of Features があるが、画像全体を対象とするために位置情報や特徴点間の関係性が失われるという問題がある。この問題に対応するため、提案手法では、SIFT 特徴点を線でつなぎ、グラフ化することで特徴点間に関係性を持たせ、位置情報を伴う構造表現を実現する。しかし、グラフ表現は統計処理に適していない。そこで、グラフをグラフ編集距離によりベクトル表現に変換してクラス分類を行う。10 クラスの画像データセットを用いた評価実験の結果、提案手法は従来手法と比較して認識率が 14.08% 向上した。

キーワード 一般物体認識, グラフ, SIFT, グラフ編集距離, グラフ-ベクトル変換

1. はじめに

一般物体認識とは、制約のない図 1 のような実世界シーンを撮影した画像に含まれる物体を、計算機が一般的名称で認識することを指し、コンピュータビジョンの分野において困難な研究課題の一つである。機械による人間の高次視覚機能の実現という観点から、ロボットビジョンへの応用などが期待されている。また近年、デジタルカメラの普及、ハードディスクの大容量化、Web の発達に伴いデジタル画像が爆発的に増大し、人手による管理が困難になっている。そこで計算機による画像の検索や分類が求められており、一般物体認識の重要性がますます高まってきている。



図 1 一般物体認識

一般物体認識に関する従来研究のアプローチには、2 つの代表的な方法がある。ひとつは、領域に基づく方法である。これは、領域分割された画像領域に対して自動アノテーションを行う手法であり、Barnard らによる word-image-translation model [1] [2] [3] などが有名である。しかしこの方法は、オクルージョンがある場合や、画像が複雑で領域分割が上手くいかない場合には、対処

することが難しい。

そこで、この問題を解決するために、局所パターンに基づく方法が提案されている。これは、画像の局所的な特徴の組み合わせによって、画像の照合を行う手法である。これには、図 2 のような Bag of Features という局所の特徴の出現頻度ヒストグラムを求め、画像全体を特徴付ける手法がよく用いられる [4]。しかし、画像全体を対象とするために、位置情報や特徴点間の関係性が失われるという問題がある。

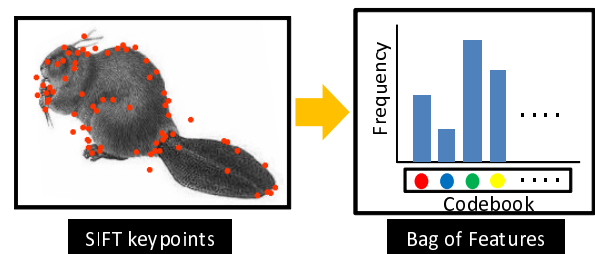


図 2 Bag of Features

本研究ではこの問題に対処するために、図 3 に示すように特徴点間を線で結び、局所の特徴の集合をグラフとして表現する。また、そのグラフをグラフ編集距離を用いて統計的学習が容易なベクトル表現に変換することで、物体構造と局所の特徴を統合した認識性能の高い手法を提案する。

本論文の構成は以下の通りである。2 章で、提案手法の流れについて述べ、3 章で画像をグラフとして表現する方法について述べる。4 章ではグラフ編集距離について述べ、5 章でグラフをベクトル表現に変換する方法について述べる。6 章では、提案手法の有効性を 10 クラスの画像データセットで評価した結果について述べ、7 章で、本論文のまとめと今後の方針について述べる。

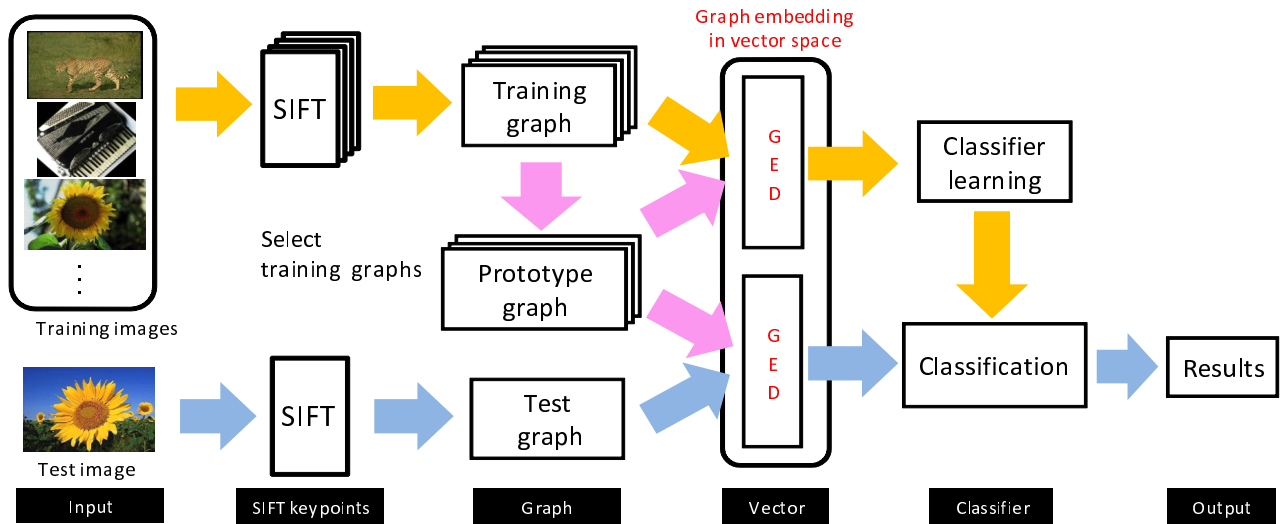


図 4 提案手法の流れ

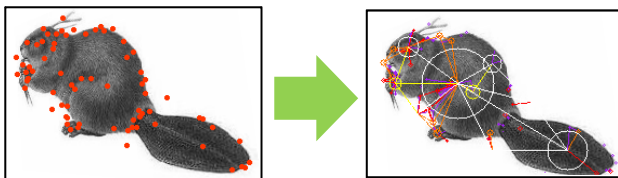


図 3 グラフの作成

2. 提案手法の流れ

簡単な提案手法の流れを図 4 に示す。まず、全ての画像について SIFT 特徴 [5] を抽出する。抽出された特徴点をつないで各画像のグラフを作成する。学習画像から作成したグラフをトレーニンググラフ、テスト画像から作成したグラフをテストグラフとする。次に、トレーニンググラフの中から n 個のプロトタイプグラフを選び出し、各グラフ（トレーニンググラフ及びテストグラフ）と n 個のプロトタイプグラフとの間で、グラフ編集距離 GED (Graph Edit Distance) を計算する。こうすることで、グラフを n 次元ベクトル空間に埋め込む。得られた学習データの n 次元ベクトルを用いてクラス分類器を学習する。学習したクラス分類器により、テストデータを分類して認識結果を出力する。

次に、提案手法の各処理について、それぞれ詳細に述べる。

3. 画像のグラフ構造表現

本研究では、[6] で提案したグラフの記法、構造表現を採用する。

3.1 グラフの定義

グラフ G を $G = (V, E, X)$ と定義する。 V はノードの集合、 E はエッジの集合、 X はノードの関連単項測定値の集合を表す。ノードは SIFT によって検出された特徴点であり、その関連単項測定値は、対応する特徴点に関

する 128 次元の特徴量である。以降では、プロトタイプグラフを G^p 、それ以外のグラフをシーングラフと呼び G^s として区別する。同様に、各グラフのノードの集合を以下のように表す。

$$V^p = \{u_1, u_2, \dots, u_\alpha, \dots\} \quad (1)$$

$$V^s = \{v_1, v_2, \dots, v_i, \dots\} \quad (2)$$

ここで、区別のためにノードに関連する添え字はグラフごとに異なるものを用いる。プロトタイプグラフにラテン文字の添え字、シーングラフにギリシャ文字の添え字を使用する。関連単項測定値 $\mathbf{x}_\alpha \in X^p$ と、 $\mathbf{x}_i \in X^s$ も同じように扱う。また、2つのノード u_α と u_β 間のエッジは $e_{\alpha\beta}$ (シーングラフの場合は e_{ij}) と表す。

3.2 近接グラフ

画像から抽出した特徴点をエッジで接続してグラフを作成するが、全ての点を相互に接続すると完全グラフとなり、通常それは計算上適していない。また、遠方の特徴点との関係性は薄いので、近傍の特徴点のみと接続するのが望ましい。そこで、遠方の特徴点に関連していないグラフを近接グラフと定義し、以下の式により近接グラフを構成するエッジを作成する。

$$E = \left\{ e_{ij} \mid \forall i, j, \frac{\|\mathbf{p}_i - \mathbf{p}_j\|}{\sqrt{\sigma_i \sigma_j}} < \chi \right\} \quad (3)$$

ここで、 $\mathbf{p} = (p_x, p_y)$ は特徴点の座標、 σ はスケール、 χ は定数を表す。この式により、スケールの大きい特徴点ほどより遠方の特徴点と接続し、定数 χ 以上の場合、エッジは作成されない。余分なエッジが作成されないことで、作成された近接グラフはコンピュータの負担をかなり軽減し、同時に検出性能を改良する。プロトタイプグラフ、シーングラフともに近接グラフとして作成する。

3.3 擬似階層グラフ

一般的に、SIFT 特徴点は、スケールが大きいと高い信頼度を示す。そこで、特徴点のスケールの大きさをグラフを階層に分割し、信頼度の高い大きいスケールの階層からグラフマッチングを開始して、徐々に階層を下げていくことにより認識性能の向上を図る。これを擬似階層グラフと定義する。また、上位の階層でマッチングしない場合には、早期に処理を終了することで、全てのノードを比較する必要がなくなり、計算時間が短縮できる。グラフの階層数を L 、階層レベルを l とする。特徴点のスケールに基づいてグラフを部分グラフ $\{G_l\}_{l=1}^L$ の集合に分解するには、次式に示す閾値 s_l を用いる。

$$s_l = \sigma_{\min} \left(\frac{\sigma_{\max} - \sigma_{\min}}{\sigma_{\min}} \right)^{\frac{l-1}{L-1}} \quad (4)$$

ここで、 $\sigma_{\max}$ 、 $\sigma_{\min}$ は各グラフにおける特徴点のスケールの最大値と最小値である。各階層レベル l において、スケール σ が閾値 s_l より大きい特徴のみを保有する。2つのグラフをグラフマッチングする際、両方ではなく片方のグラフのみを擬似階層グラフとする。これは、各グラフのマッチングすべきノードが、同じスケール階層にあるとは限らないからである。本研究では、プロトタイプグラフを近接グラフとして作成し、擬似階層グラフに変換する。階層数3に分割したプロトタイプグラフ $\{G_l^p\}_{l=1}^3$ の例を図5に示す。

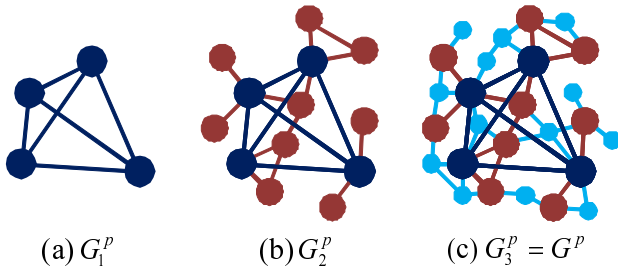


図5 プロトタイプグラフ

4. グラフ編集距離

2つのグラフの構造的類似性を評価する過程は、一般的にグラフマッチングと呼ばれる。この問題に対して多くの研究が行われている[7]。本論文ではその中でグラフ編集距離[8][9]を使用する。これは、2つのグラフの相違を計算するためによく用いられる方法であり、多くの最適近似アルゴリズムが提案されている[10][11][12]。それらの論文で主に問題とされているのはグラフ編集距離の計算量であり、それを改善するために本論文では擬似階層グラフを用いる。

4.1 グラフ編集距離の定義

グラフ編集距離の基本的な考え方は、1つのグラフを他のグラフに変換するのに必要である最小の編集数として、2つのグラフの相違を定義することである。つまり、ノードとエッジの挿入、削除、置換で構成される編集操作 e_d の数で定義される。全てのグラフの組 G_1 と G_2 において、特定の編集操作で G_1 を G_2 に変換する編集経路の集合 $h(G_1, G_2) = (e_{d1}, \dots, e_{dk})$ (ここで各 e_{di} が編集操作を表す) が存在する。図6で2つのグラフ G_1 と G_2 間の編集経路の例を与える。この編集経路は、(a) 1つのエッジの削除、(b) 1つのノードの置換、(c) 1つのノードの挿入、および(d) 2つのエッジの挿入から成る。一般的に、2つのグラフ間の編集経路は複数存在する。どの編集経路が最良かを評価するために、編集コストを導入する。基本的な考え方は、それぞれの編集操作によって起こるひずみを、編集コスト c で表すことである。2つのグラフ G_1 と G_2 の間の編集距離 d は、 $d(G_1, G_2)$ で表され、1つのグラフを他のグラフに変換する最小コストである編集経路のコストとなる。それは以下の式によって表される。

$$d(G_1, G_2) = \min_{(e_{d1}, \dots, e_{dk}) \in h(G_1, G_2)} \sum_{i=1}^k c(e_{di}) \quad (5)$$

ここで、 $h(G_1, G_2)$ は G_1 から G_2 へ変換する編集経路のセット、 $c(e_d)$ は編集操作 e_d のコストを示す。

4.2 グラフ編集距離の定義からの変更

4.1のグラフ編集距離の定義を、本研究用に変更する。変更する主な理由は、本来階層的なグラフに使用できないグラフ編集距離を、擬似階層グラフに適用可能にするためである。

まず、定義では2つのグラフを完全に一致させるまで編集を行うが、1つのグラフの一部と他のグラフの一部を等しくするまでの編集に留める。これは、クラス分類すべき物体の領域外の必要ないノードが、グラフに含まれる可能性があるからである。このため、物体領域外のノードとエッジを消してしまう可能性がある編集操作の削除は行わない。

次に、定義では2つのグラフ間の編集経路を複数発見し、最小コストの経路を選択するが、グラフの擬似階層構造を用いて階層ごとに最適化を行うことで、最良の一つの経路のみを求める。

最後に、グラフ編集距離 d は距離なので、値が小さいほど類似度が高い。しかし、今回はそれを逆転させて、類似度が高いほど値が大きくなるグラフ編集距離 $\tilde{d}$ を用いる。編集距離に関する全ての値(編集コスト c など)の大小を逆転させるのみなので本質的に変化はない。これは、擬似階層グラフの特性を生かすためである。3.3で示したように、2つのグラフの類似度が低ければ、擬似

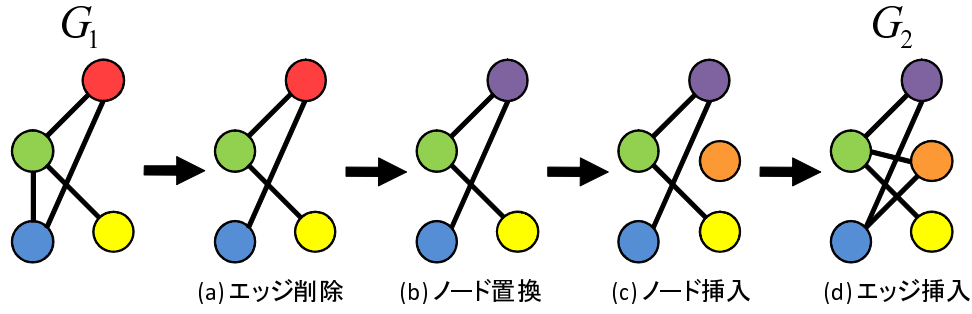


図 6 グラフ間の編集経路

階層グラフによって上位階層のグラフマッチングのみで早期に処理を終了する。このとき、距離の指標を用いると上位階層で終了するほうが下位階層で続けて処理を行うよりも、編集コスト c の累積が少なく、類似度が低いにも関わらずグラフ編集距離 d が小さくなってしまう。よって編集コストが累積するほど類似度が高くなるような指標に変更したグラフ編集距離 $\tilde{d}$ を用いる。

4.3 ノードとエッジのマッチング

4.2 のグラフ編集距離 $\tilde{d}$ を用いるために、プロトタイプグラフとシーングラフ間で、ノードとエッジのマッチングを定義する。この結果によってどの編集操作 (挿入, 置換) を行うかを決定する。つまり、プロトタイプグラフのノード, エッジに対してシーングラフに類似度の高いノード, エッジが存在すれば置換, 存在しなければ挿入を行う。以下で、ノードとエッジそれぞれについて述べる。

ノードのマッチングには、関連単項測定値である 128 次元の SIFT 特徴量を用いる。原則として、1 つのノードとマッチングするノードは 1 つのみとする。階層レベル l において、プロトタイプノード u_α とシーンノード v_i を考える。SIFT 特徴量 $\mathbf{x}_\alpha \in X^p$ と $\mathbf{x}_i \in X^s$ を用いて、ノード間の相違を計算する。その値が、プロトタイプノード u_α に関して最小となるシーンノード v_i を、以下の式により探索する。

$$\mu(v_i, u_\alpha) = \min_{v_i \in V^s} \|\mathbf{x}_\alpha - \mathbf{x}_i\| \quad (6)$$

ここで $\|\mathbf{x}_\alpha - \mathbf{x}_i\|$ はユークリッド距離である。この最小相違値 μ が定数 ε_1 以下の時、ノードとノードはマッチングしているとみなし ($v_i \rightarrow u_\alpha$)、ノードの置換を行う。すなわち、シーンノード v_i を階層レベル l のプロトタイプノード u_α で置き換える。それ以外はマッチングするシーンノードが存在しないと判定し、ノードの挿入を行う。すなわち、シーングラフに階層レベル l のプロトタイプノード u_α を挿入する。ただし、シーンノードにマッチングするプロトタイプノードが複数ある場合は、最小相違値 μ が小さいほうのみがマッチングしていると

次に、エッジのマッチングは、以下の不変な特徴を抽出することによって行う。式 (7) から式 (10) は、正規化されたエッジの長さ, 正規化された角度, 正規化されたスケール差, 角度差を表す。

$$e_{\alpha\beta}^{(1)} = \frac{\|\mathbf{p}_\alpha - \mathbf{p}_\beta\|}{(\sigma_\alpha + \sigma_\beta)} \quad (7)$$

$$e_{\alpha\beta}^{(2)} = \theta_{\alpha\beta} - \theta_\alpha \quad (8)$$

$$e_{\alpha\beta}^{(3)} = \frac{|\sigma_\alpha - \sigma_\beta|}{\max(\sigma_\alpha, \sigma_\beta)} \quad (9)$$

$$e_{\alpha\beta}^{(4)} = \theta_\alpha - \theta_\beta \quad (10)$$

ここで $\theta_{\alpha\beta}$ はエッジのオリエンテーション, θ_α と θ_β は特徴点のオリエンテーションを示す。このエッジ記述子 $\{e_{ij}^{(n)}\}_{n=1}^4$ が 4 つの独立している正規分布に従うと仮定し、以下の式によってエッジの類似度 δ を求める。

$$\delta(e_{ij}, e_{\alpha\beta}) = \prod_{n=1}^4 \frac{1}{\sqrt{2\pi}\sigma^{(n)}} \exp\left(-\frac{(e_{\alpha\beta}^{(n)} - e_{ij}^{(n)})^2}{2\sigma^{(n)2}}\right) \quad (11)$$

ここで、 $\sigma^{(n)}$ は各成分の分散である。この類似度 δ が定数 ε_2 以上の時、エッジはマッチングしているとみなし、エッジの置換を行う。それ以外はエッジの挿入を行う。ノード間のエッジには、方向の違う 2 通りが存在するので、重複して編集しないように計算を行う。

4.4 グラフ編集距離の計算

4.3 のマッチング方法を用いて、グラフ編集距離を計算するアルゴリズムを図 7 に示す。ここでは、このアルゴリズムについて詳細に述べる。

T_1, T_2 はマッチングしたノードの組合せのセットであり、 T_1 はマッチングした全てのノードの組、 T_2 は繰り返しの際に新たにマッチングしたノードの組を格納する。

まず、最もスケールの大きい階層レベル 1 において、初期化を行う。4.3 で述べたノードのマッチングを、階層レベル 1 のプロトタイプノード $u_\alpha \in V_1^p$ とシーンノード $v_i \in V^s$ との間で行う。この時、ノードがマッチングしたらその組合せを T_1 に加える (図 7 の Initialization(2))。次に、マッチングしたプロトタイプノード同

Initialization (level $l = 1$):

- (1) Initialize T_1 to an empty list.
- (2) For each $u_\alpha \in V_1^p$ and for each $v_i \in V^s$:
Try to match nodes (4.3).
if $v_i \rightarrow u_\alpha$: Add to $(v_i \rightarrow u_\alpha)$ to T_1 .
- (3) For each $e_{\alpha\beta} \in E_1^p$ and for each $e_{ij} \in E^s$:
if $v_i \rightarrow u_\alpha$ and $v_j \rightarrow u_\beta$:
Try to match edges (4.3).

Iteration: For each $l \in [2..L]$:

- (1) Initialize T_2 to an empty list.
- (2) if $l = L$: exit and return GED.
- (3) For each match $(v_i \rightarrow u_\alpha) \in T_1$:
For each $u_\beta \in V_l^p$ neighboring to u_α :
For each $v_j \in V^s$ neighboring to v_i :
Try to match nodes (u_β and v_j) (4.3).
if $v_j \rightarrow u_\beta$:
Substitution of an edge ($e_{ij} \rightarrow e_{\alpha\beta}$).
Add to $(v_j \rightarrow u_\beta)$ to T_1 and T_2 .
else: Insertion of a node (u_β) to V^s .
Insertion of an edge ($e_{\alpha\beta}$) to E^s .
- (4) For each match $(v_i \rightarrow u_\alpha) \in T_2$:
For each u_β neighboring to u_α :
For each v_j neighboring to v_i :
if $(v_j \rightarrow u_\beta) \in T_1$:
Try to match edges ($e_{\alpha\beta}$ and e_{ij}) (4.3).

図7 グラフ編集距離 アルゴリズム

士を接続するエッジ $e_{\alpha\beta}$ と、対応するシーンノード同士を接続するエッジ e_{ij} に対してマッチングを行う (図7の Initialization(3)).

以降は、階層レベル l が階層数 L になるまで繰り返す。すでにマッチングしているノード (u_α, v_i) に接続している階層レベル l のプロトタイプノード $u_\beta \in V_l^p$ と、シーンノード $v_j \in V^s$ に対して、マッチングを行う。マッチングに成功すれば、エッジの置換を行い、 T_1 と T_2 にノードの組合せを加える。つまり、マッチングしたノード間のエッジは、その相違にかかわらず必ず置換を行う。マッチングしなかった場合は、類似度の高いシーンノードが存在しないと判定し、ノードの挿入を行う。また、新たに挿入したノードと、それが接続していたプロトタイプノードにマッチングしているシーンノードとの間に、エッジを挿入する。ここで、新たに挿入したノードやエッジはもともと存在しないものなので、以後の処理を行わない (図7の Iteration(3)). 次に、 T_2 に格納されているこの繰り返しで新たにマッチングしたノードの組 (u_α, v_i) ($(v_i \rightarrow u_\alpha) \in T_2$) について考える。 T_1 に格納されているマッチングした全てのノードの組 (u_β, v_j) ($(v_j \rightarrow u_\beta) \in T_1$) として、マッチングしたプロトタイプノード同士を接続するエッジ $e_{\alpha\beta}$ と、対応するシーンノード同士を接続するエッジ e_{ij} に対してマッチングを行う (図7の Iteration(4)).

このように初期化の際にマッチしたノードから、エッジをたどってマッチングを行っていく。こうして、ノードとエッジの編集を行い、その編集コストの総和をグラフ編集距離として出力する。

5. グラフ-ベクトル変換

本研究で用いる変換手法は、[13] で提案された手順に従う。この手法は、メディアングラフの導出などに使用され、その有効性が示されている [14] [15]。以下でそのアプローチについて述べ、その概略を図8に示す。

トレーニンググラフ $T = \{G_1, G_2, \dots, G_n\}$ のセットを用意し、グラフ編集距離 $\tilde{d}(G_i, G_j)(i, j = 1, \dots, n; G_1, G_2 \in T)$ を計算する。まず、 m 個のプロトタイプグラフのセット $P = G_1^p, \dots, G_m^p \in T$ を、 $T(m \leq n)$ から選ぶ。次に、シーングラフ G^s と全てのプロトタイプグラフ $G^p \in P$ の相違を計算する。従って、各シーングラフについて、 m 個の相違評価値 $\tilde{d}_1, \dots, \tilde{d}_m$ (ここで $\tilde{d}_k = \tilde{d}(G^s, G_k^p)$) が求められる。この評価値を m 次元ベクトルとする。このように、プロトタイプグラフセット P を用いて、全てのシーングラフ G^s を m 次元ベクトルに変換できる。次式のように埋め込み手順を定義する。

$$\psi: G^s \rightarrow \mathbf{R}^m \quad (12)$$

$$\psi \rightarrow (\tilde{d}(G^s, G_1^p), \tilde{d}(G^s, G_2^p), \dots, \tilde{d}(G^s, G_m^p)) \quad (13)$$

ここで、 $\tilde{d}(G^s, G_i^p)$ はグラフ相違評価値、つまりグラフ編集距離である。本研究では、この定義に従って、グラフの埋め込みを実行する際に、プロトタイプグラフ P のセットとトレーニンググラフ T のセットは同一のものをを用いる。

6. 評価実験

本章では、10 クラスの物体の分類実験を行い、提案手法であるグラフ構造表現を用いた一般物体認識と、従来手法である BoF を比較することで、その有効性を評価する。

6.1 実験条件

実験には、Caltech-101 データベース [16] を用いる。Caltech-101 データベースは、一般物体認識を対象とするデータベースで、101 クラスで構成され、同一クラスの物体でも色や形が大きく異なる。本実験では、その中から 10 クラスを選び、提案手法と従来手法 (BoF) で認識率の比較を行う。今回実験で用いたクラスの画像の例を、図9に示す。

各クラスの画像から、それぞれランダムに 30 枚選んで学習画像とし、残りをテスト画像とする (Caltech-101 はクラスごとに画像数が異なる)。それを 1 セットとして、学習画像の異なる 10 セットのデータセットを作成し、それぞれの分類実験による認識率の平均を、実験結

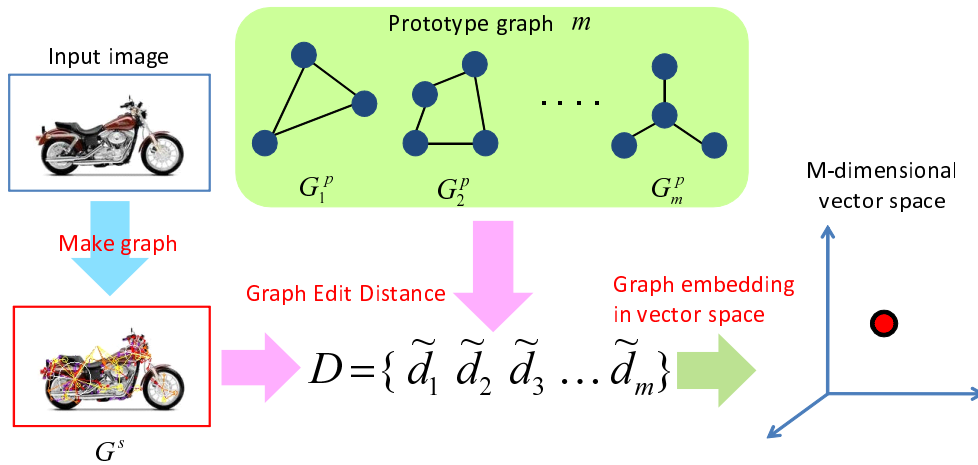


図 8 グラフ-ベクトル変換

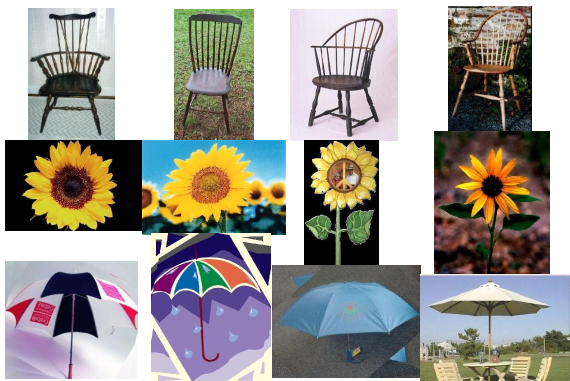


図 9 Caltech-101 データベース

果として評価する．10 クラスの画像 841 枚から選んだ学習画像 300 枚，テスト画像 541 枚のデータセット 10 セットを用いて実験を行う．また，提案手法で用いる閾値の定数 χ , ε_1 , ε_2 や擬似階層グラフの階層数 L ，グラフ編集距離の編集コスト c は実験的に最適値を用いている．

6.2 実験概要

実験は，まず 3 章で述べたように，全ての画像から SIFT 特徴を抽出し，グラフ構造で表現する．次に，トレーニンググラフの中からプロトタイプグラフを選び擬似階層グラフへと変換する．この際，5 章で述べたように，プロトタイプグラフのセットとトレーニンググラフのセットは同一のものをを用いる．次に，4 章，5 章で示したように，シーングラフとプロトタイプグラフの間のグラフ編集距離を求めて，グラフをベクトル空間に埋め込む．学習画像が 300 枚なので，全ての画像は，提案手法により 300 次元のベクトル表現に変換される．また，従来手法では，BoF の Codebook サイズを 1000 とし，1000 次元でベクトル量子化を行う．本実験では，比較のため，それぞれの手法で作成されたベクトルをクラス分類するための識別器として，k-NN 法 (k-Nearest Neighbor algorithm) と SVM (Support Vector Machine) を用いる．

SVM は線形入力素子を利用して，2 クラスのパターン識別器を構成する手法である．現在知られている多くの手法の中でも，最も認識性能の優れた学習モデルの一つである．SVM は，基本的には 2 クラスの識別にしか対応していないが，複数の SVM を組み合わせることでマルチクラスに対応できる．本実験では，マルチクラス SVM (線形カーネル linear) とマルチクラス SVM (非線形カーネル radial basis function (RBF)) を用いる．

従って，k-NN 法 ($k = 10$)，マルチクラス SVM (線形カーネル linear)，マルチクラス SVM (非線形カーネル radial basis function (RBF)) の 3 つの識別器によって，画像のクラス分類を行い認識率を比較する．

6.3 実験結果と考察

まず，全ての実験に関する認識率を図 10 に示す．次に各クラスごとの認識率を図 11，図 12，図 13 に示す．

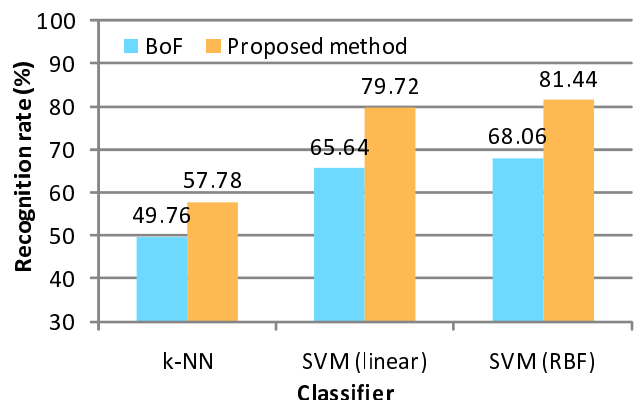


図 10 認識結果

図 10 より，グラフ構造表現を用いた提案手法を導入することで，どの識別器においても，従来手法 (BoF) より認識率が向上した．識別器ごとにみると，SVM (RBF) で 13.38%，SVM (linear) で 14.08%，k-NN 法で 8.02% 向上した．これより，グラフ構造表現をベクトル化し

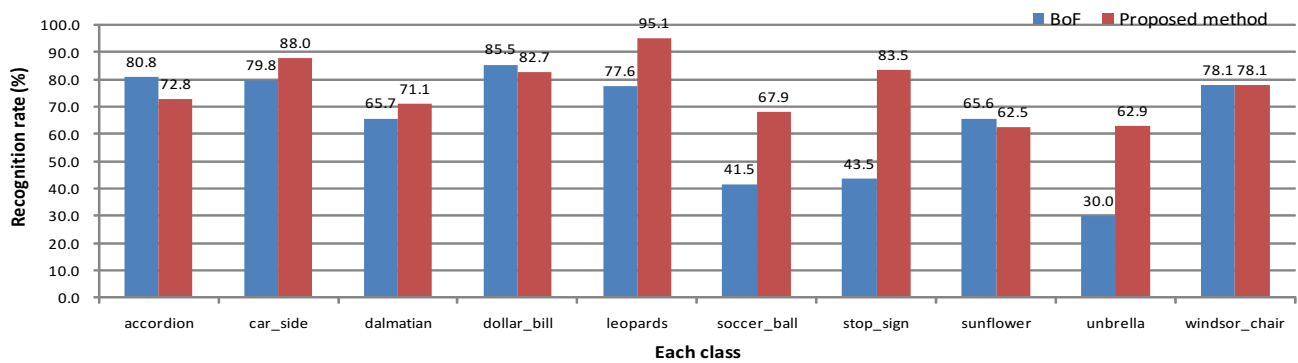


図 11 各クラスの認識結果 (SVM(RBF))

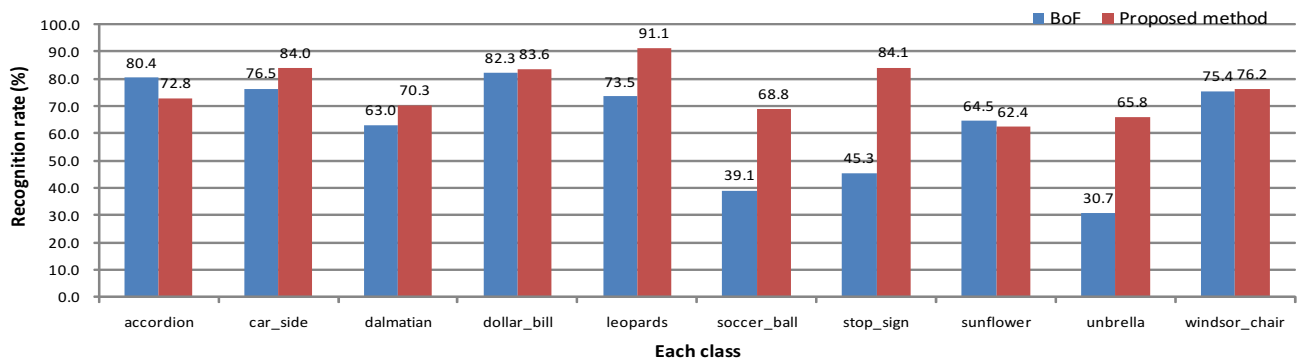


図 12 各クラスの認識結果 (SVM(linear))

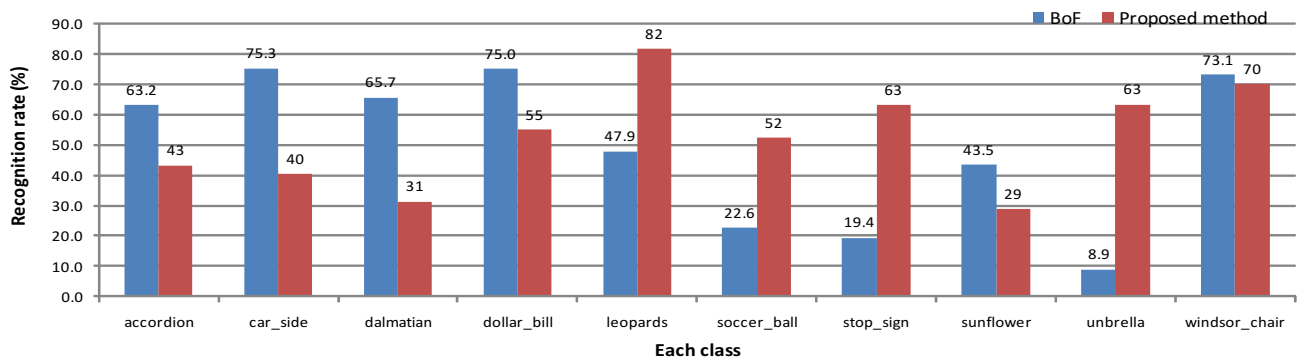


図 13 各クラスの認識結果 (k-NN)

た提案手法の有効を示すことができた。認識率が最も高かったのは、提案手法で SVM (RBF) を用いた場合で、81.44%であった。また、3つの識別器の中で、従来手法 (BoF) と比較して最も認識率が向上したのは SVM (linear) であり、14.08%の向上を示した。

図 11, 図 12, 図 13 において、認識率をクラスごとにみると、leopards, soccer_ball, stop_sign, umbrella の認識率が、BoF に比べて大幅に向上している。このように、従来手法 BoF よりも認識率が向上した理由について、以下に考察する。

本研究で用いた SIFT 特徴量は、画像の輝度勾配情報から求められる。従って、輝度勾配が少ない、あるいは同一クラス他画像と著しく輝度が異なると抽出した特

徴量の値が適切ではないので、従来手法の BoF では認識率が低下する。これに対して形状によって物体を比較する提案手法では、特徴量の値が適切でなくても、特徴点さえ抽出できていればグラフを作成できるので、認識率の低下は起こらない。図 14 は、提案手法で認識が成功し、従来手法で認識が失敗した画像である。下段右から 2 番目の傘画像には、ほとんど輝度情報が含まれていないが、提案手法では正しく認識できていることが分かる。

また、図 13 において、識別器 k-NN 法による認識率をクラスごとにみると、accordion, car_side, dalmatian, dollar_bill の認識率が、BoF と比較して大幅に低下している。しかし、識別器に SVM を用いた図 11, 図 12 では、この低下は改善されている。従って、提案手法は識



図 14 認識が提案手法で成功し，従来手法で失敗した画像

別器に SVM を用いることで，より性能を発揮できることが分かる。

以上より，グラフ構造表現を一般物体認識に導入することは，有効であることが分かった。しかし，導入方法には改善の余地があると思われる。特にグラフ編集距離の計算は，擬似階層グラフや近接グラフを用いることで軽減したとはいえ，まだかなりの時間がかかる。また，今回はプロトタイプグラフをトレーニンググラフと同一として実験を行ったが，トレーニンググラフからプロトタイプグラフを選択する手法を工夫することで，より画像の特徴を明確に表現できるようになり，認識率が向上すると考えられる。

7. ま と め

本論文では，ベクトル空間へのグラフの埋め込みを用いたグラフ構造表現による，新しい一般物体認識手法を提案した。識別器としてマルチクラス SVM (RBF) を用いた場合，提案手法は従来手法の BoF に比べて，認識率が 14.08% 向上した。今後は，プロトタイプグラフの選択手法の改良や，グラフ編集距離の計算時間を更に短縮する予定である。また，この提案手法を，3 次元画像から作成された 3 次元グラフに拡張し，一般物体認識に応用することも検討していく予定である。

文 献

- [1] K. Barnard and D. A. Forsyth, "Learning the Semantics of Words and Pictures," IEEE International Conference on Computer Vision, Vol. 2, pp. 408–415, 2001.
- [2] K. Barnard, P. Duygulu, N.d. Freitas, D. Forsyth, D. Blei and M. Jordan, "Matching Words and Pictures," Journal of Machine Learning Research, Vol. 3, pp. 1107–1135, 2003.
- [3] P. Duygulu, K. Barnard, N.d. Freitas and D. Forsyth, "Visual categorization with bags of keypoints," European Conference on Computer Vision, pp. IV:97–112, 2002.
- [4] G. Csurka, C.R. Dance, L. Fan, J. Willamowski, C. Bray, "Object Recognition as Machine Translation: Learning aLexicons for a Fixed Image Vocabulary," ECCV Workshop on Statistical Learning in Computer Vision, pp. 1–22, 2004.
- [5] D. G. Low, "Distinctive image features from scale-invariant keypoints," Journal of Computer Vision, Vol. 60, No. 2, pp. 91–110, 2004.
- [6] J. Revaud, Y. Arikawa and A. Baskurt, "Scale-Invariant Proximity Graph for Fast Probabilistic Object Recognition," Conference on Image and Video Retrieval (CIVR), pp. 414–421, 2010.
- [7] D. Conte, P. Foggia, C. Sansone and M. Vento, "Thirty years of graph matching in pattern recognition," International Journal of Pattern Recognition and Artificial Intelligence, Vol. 8, No. 3, pp. 265–298, 2004.
- [8] H. Bunke and G. Allerman, "Inexact graph matching for structural pattern recognition," Pattern Recognition Letters, Vol. 1, pp. 245–253, 1983.
- [9] A. Sanfeliu, and K. Fu, "A distance measure between attributed relational graphs for pattern recognition," IEEE Transactions on Systems, Man, and Cybernetics, Vol. 13, No. 3, pp. 353–362, 1983.
- [10] D. Justice and A. Hero, "A Binary Linear Programming Formulation of the Graph Edit Distance," IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 28, No. 8, pp. 1200–1214, 2006.
- [11] M. Neuhaus, K. Riesen and H. Bunke "Fast suboptimal algorithms for the computation of graph edit distance," Joint IAPR International Workshops, SSPR and SPR 2006, Lecture Notes in Computer Science, Vol. 4109, pp. 163–172, 2006.
- [12] K. Riesen and H. Bunke, "Approximate graph edit distance computation by means of bipartite graph matching," Image and Vision Computing, Vol. 27, No. 7, pp. 950–959, 2009.
- [13] K. Riesen, M. Neuhaus and H. Bunke, "Graph Embedding in Vector Spaces by Means of Prototype Selection," Graph-based representations in pattern recognition (GbrPR), ed. F. Escolano et al, pp. 383–393, Springer-Verlag Berlin, Heidelberg, 2007.
- [14] E. Valveny and M. Ferrer, "Application of Graph Embedding to solve Graph Matching Problems," CIFED, pp. 13–18, 2008.
- [15] M. Ferrer, E. Valveny, F. Serratos, K. Riesen and H. Bunke, "Generalized median graph computation by means of graph embedding in vector spaces," Pattern Recognition, Vol. 43, No. 4, pp. 1642–1655, 2010.
- [16] Caltech-101 Database, http://www.vision.caltech.edu/Image_Datasets/Caltech101/