

平成 29 年度 卒業研究論文

題目 A-KAZE 特徴量を用いた
キノコ画像認識に関する研究

指導教員 服部 峻

提出者 室蘭工業大学 情報電子工学系学科

氏 名 古川 拓人

学籍番号 14024161

提出年月日 平成 30 年 2 月 13 日

目次

第 1 章	序論	1
第 2 章	人によるキノコの判別方法	2
第 3 章	提案システム	3
3.1	システム全体の流れ	3
3.2	類似度算出	4
3.3	A-KAZE (acceralated-KAZE) 特徴量 [2]	8
3.3.1	非線形拡散フィルタリング	8
3.3.2	AOS 手法	9
3.3.3	非線形スケールスペースの計算	9
3.3.4	特徴点検出	10
3.3.5	特徴点について	10
第 4 章	評価実験	12
4.1	実験環境	12
4.2	実験結果	13
4.3	考察	16
第 5 章	まとめと今後の課題	18
	謝辞	19
	参考文献	20

目次

2.1	キノコの各部位の名称 [1]	2
3.1	システム全体の流れ	3
3.2	特徴点類似度算出	5
3.3	色ヒストグラム類似度算出	6
3.4	色ヒストグラム類似度の閾値以上に特徴点類似度を適用	7
4.1	それぞれの特徴量のみの精度グラフ	13
4.2	手法 3 における閾値ごとの精度グラフ (RGB 空間)	14
4.3	手法 3 における閾値ごとの精度グラフ (HSV 空間)	15
4.4	単生のキノコ画像の類似特徴点抽出結果	16
4.5	群生のキノコ画像の類似特徴点抽出結果	16

表目次

4.1	それぞれの特徴量のみの順位結果	13
4.2	手法 3 における閾値ごとの順位結果 (RGB 空間)	14
4.3	手法 3 における閾値ごとの順位結果 (HSV 空間)	15

第 1 章

序論

近年、毒キノコによる食中毒の被害は少なくなってきたが、毎年最低でも 100 人から 200 人の被害を出している。被害が絶えないのはキノコを見分けることが困難であることが挙げられる。キノコを見分けるときには多くの情報が必要となる。かといってキノコの図鑑を持っていくと山の中を歩く時には重いことからあまり適していない。また、スマートフォンなどで調べるにしても情報が多いためその中から特定するのは難しく、時間もかかる。それにも関わらず世の中にはキノコを判別するための情報の中には「派手な色のものは毒キノコである」といった間違った情報が溢れている。このような状況の中で素人の人たちがキノコ狩りをするのは大変である。

そこで本研究では、キノコ図鑑などの正しい情報をもとに素人の人たちでも簡単にキノコを判別できるようにするための判別ツールの開発を目的とし、そのための手法として、A-KAZE 特徴量と色ヒストグラム類似度を利用したキノコ画像認識を提案する。

第2章

人によるキノコの判別方法

図 2.1 にキノコの各部位の名称を示す。人によってキノコを見分ける時は、各部位の特徴の違いで判断する。ほとんどのキノコは色や形状、生え方で見分けることができる。キノコの色は主に茶、赤、黄、白、その他で大きく分けることができ、傘やヒダ、柄などの各部位に特徴として現れる。形状では傘の部分には半球型、釣鐘型、饅頭型など大体の形に分類でき、その他にも柄やいぼ、つば、つぼなどにもそれぞれで形や大きさまたは有無で特徴として確認できる。生え方には単生や群生、束生などの特徴がある。ほとんどのキノコはこれらの特徴を組み合わせることで、どのキノコであるのか判断することができる。ここからキノコの特定するには、さらにどのような場所に生えているのか、キノコを割いた時に内側がどうなっているのか、などの単純に見ただけでは分からない情報も必要となる。本研究ではこれらの様々な情報から、色や形状に焦点を当てている。

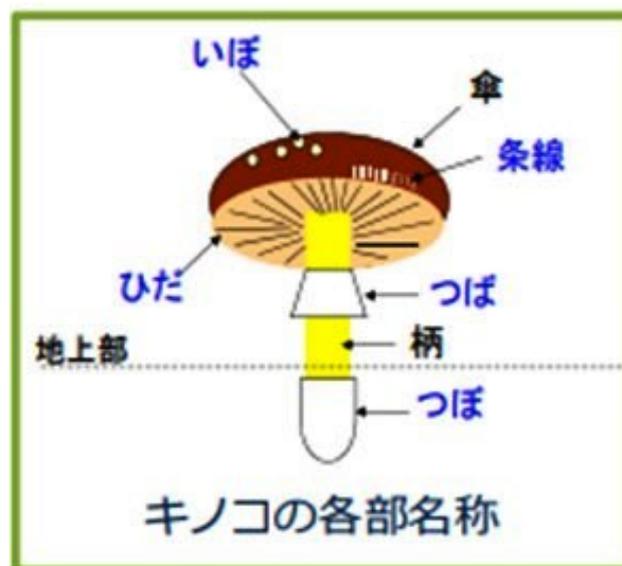


図 2.1 キノコの各部位の名称 [1]

第3章

提案システム

本章では，提案システムについて述べる．まず3.1節でシステム全体の流れについて，次に3.2節では類似度算出手法について，3.3節では A-KAZE 特徴量について記述する．

3.1 システム全体の流れ

図 3.1 にシステムの全体の構造を示す．

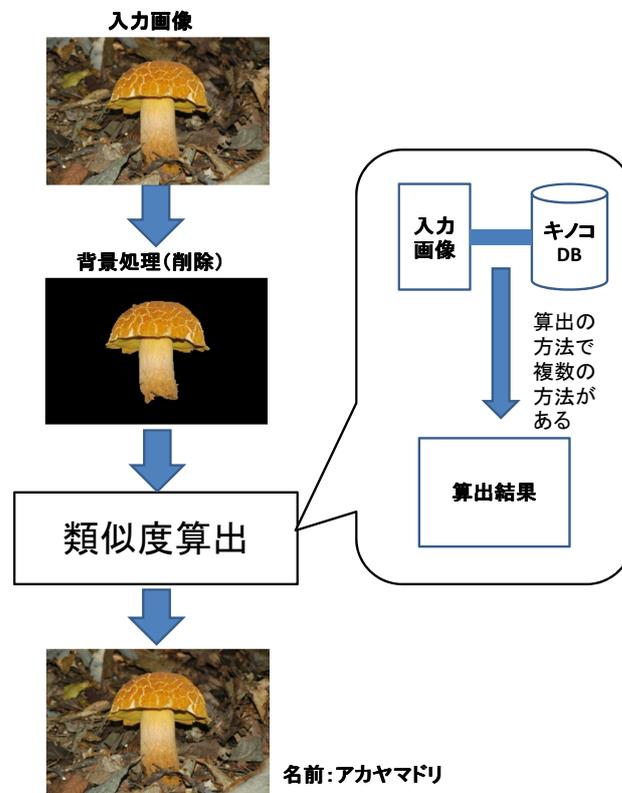


図 3.1 システム全体の流れ

システムの流れはキノコの画像を入力し、その画像に対して背景処理を行い、キノコデータベース内のキノコ画像と背景処理された入力画像に類似度算出を適用して最終的にキノコの名前を表示するという流れになっている。但し、本研究において入力画像とデータベース画像内の背景処理はあらかじめ物体（キノコ）と背景に分けて背景だけ削除した画像を用意した。

また本研究では、類似度算出の段階で3つの手法を提案する。

手法1：特徴点類似度算出

手法2：色ヒストグラム類似度算出

手法3：閾値を利用して2つの特徴量を適用

類似度算出の方法は上記の3つの手法に分かれている。また手法2, 3ではRGB空間とHSV空間という2つの色空間を使用し、それぞれの空間ごとの類似度算出の結果を出している。3.2節からそれぞれの手法の説明をしていく。

3.2 類似度算出

本節では、前節で記した3つの類似度算出手法について、順に記述していく。

手法1：特徴点類似度算出

手法1のシステムの流れを図3.2に示す。手法1は特徴点による類似度算出のみで画像を判定する方法である。キノコ画像（写真）から抽出する特徴点として「A-KAZE 特徴量」を採用し、2つの画像間の類似特徴点の探索には総当たり法により特徴点を探索している。類似度計算には平均距離によって算出している。このとき2点間の距離はハミング距離によって算出している。距離の最小の値は0であり、求められた平均距離の値は $0 \sim \infty$ の範囲で表されて、0に近ければ近いほど類似度が高いことを示し、全く同じ画像であれば値は0を示す。以下に平均距離の計算式を示す。

$$\text{平均距離} = \frac{\text{特徴点の距離の総和}}{\text{画像全体の特徴点の個数}} \quad (3.1)$$

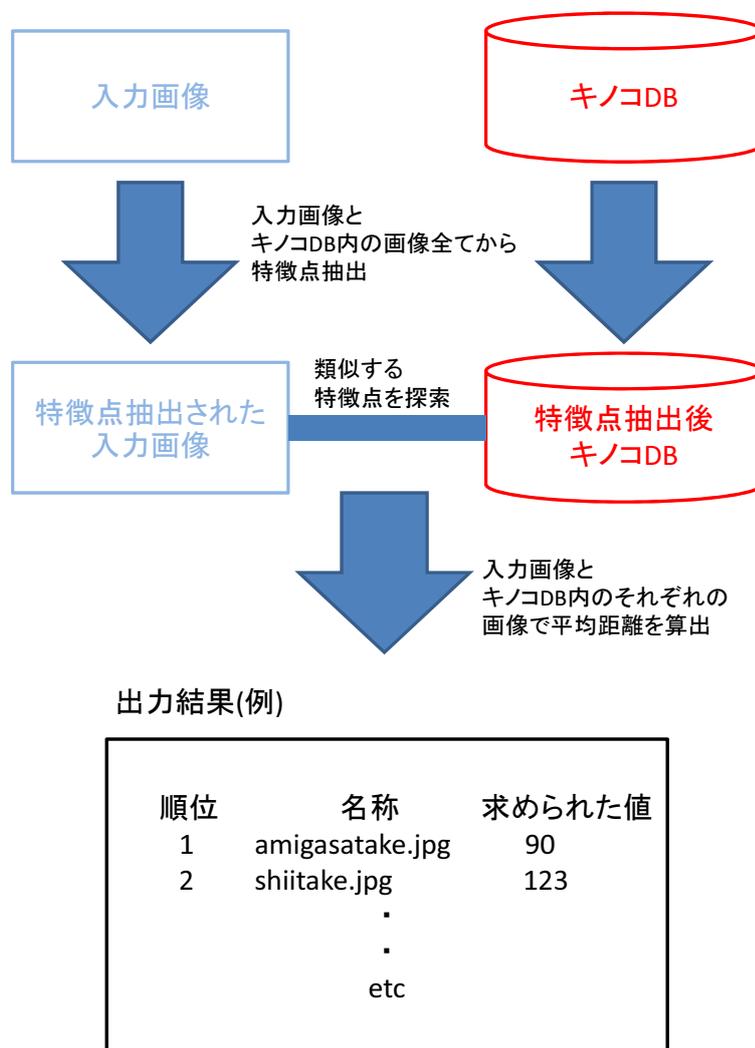


図 3.2 特徴点類似度算出

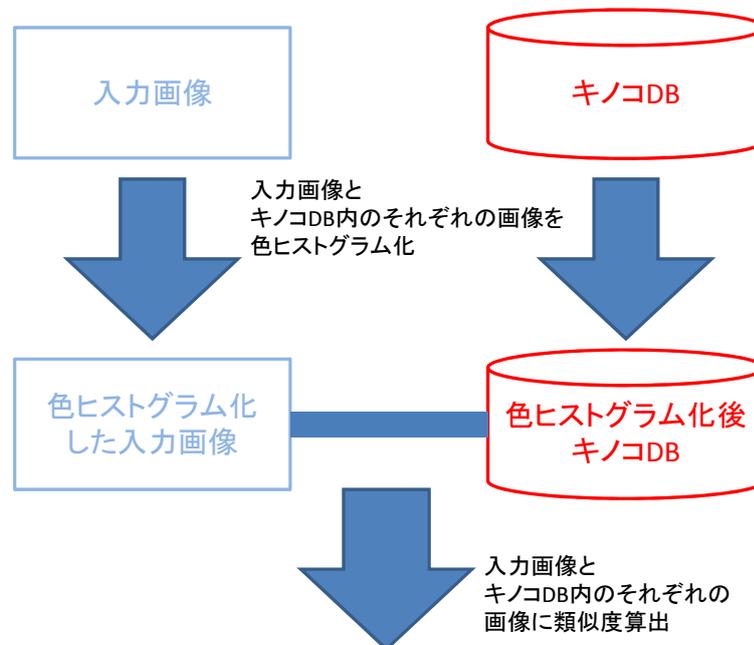
手法 2 : 色ヒストグラム類似度算出

手法 2 のシステムの流れを図 3.3 に示す。手法 2 では与えられた入力画像とキノコデータベース内の全ての画像に対して色ヒストグラムの作成を行った後に、入力画像とキノコデータベースの画像のヒストグラム同士の類似度を求める方法である。本研究では相関を使用している。下式に相関についての計算式を記述する。

$$d(H_1, H_2) = \frac{\sum_I (H_1(I) - \bar{H}_1)(H_2(I) - \bar{H}_2)}{\sqrt{\sum_I (H_1(I) - \bar{H}_1)^2 \sum_I (H_2(I) - \bar{H}_2)^2}} \quad (3.2)$$

$$\bar{H}_k = \frac{1}{N} \sum_I H_k(I) \quad (3.3)$$

このうち H_1 は入力画像のヒストグラム, H_2 はデータベースの画像のヒストグラムである. また N は, ヒストグラムのビンの総数を表す. 本研究では, ビンの総数は 255 である. 求められた値は -1 から 1 の範囲で表されて 1 に近ければ近いほど類似度が高いことを示し, 全く同じ画像であれば 1 を示す.



出力結果(例)

順位	名称	求められた値
1	amigasatake.jpg	0.9
2	shiitake.jpg	0.8
	.	
	.	
	etc	

図 3.3 色ヒストグラム類似度算出

手法3：閾値を利用して2つの特徴量を適用

手法3のシステムの流れを図3.4に示す。手法3では初めに色類似度を求めるときに閾値を設定し変化させ、求められた色類似度の値が設定された閾値以上の画像に対してA-KAZE特徴量に基づく特徴点類似度算出を適用することで画像を判定する方法である。類似度算出の順番の理由は、特徴点類似度算出は平均距離で計算しているため、一定の範囲に値が収まっていないので、全ての画像で絶対に適用できるという閾値が存在しなかったためこのような順番で定義した。

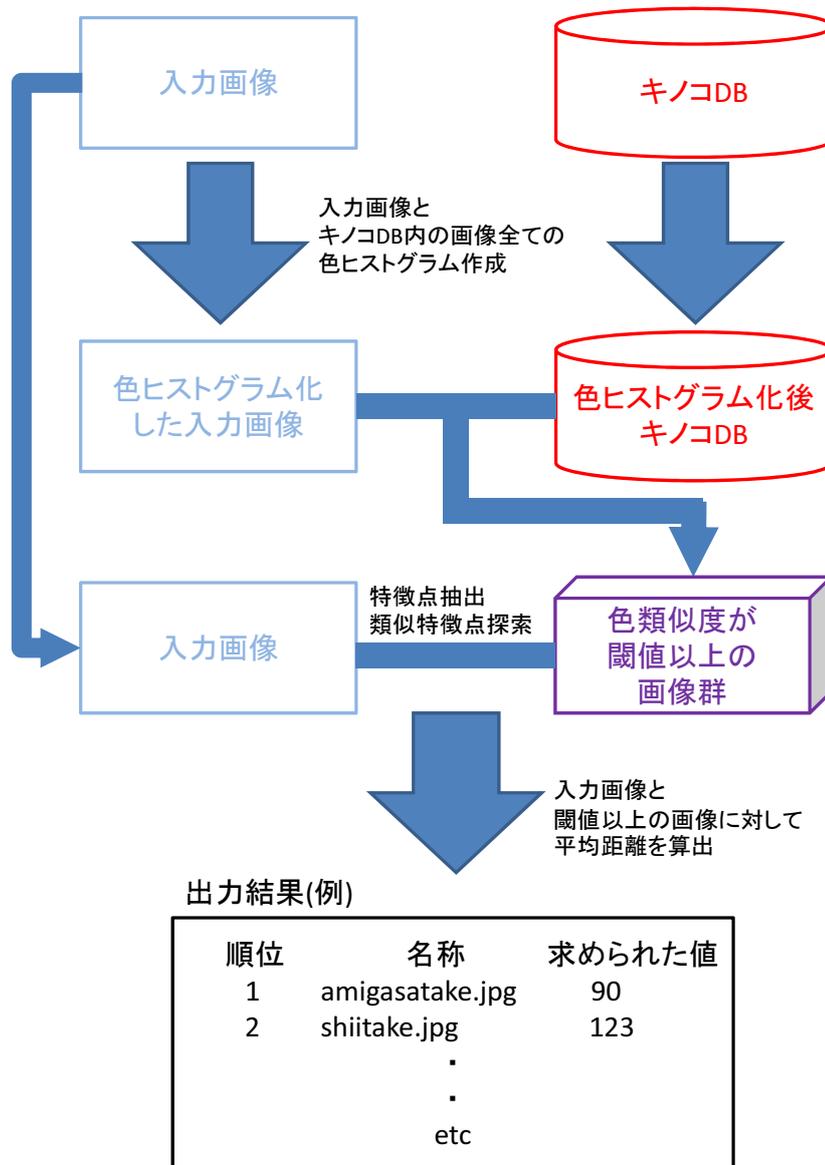


図 3.4 色ヒストグラム類似度の閾値以上に特徴点類似度を適用

3.3 A-KAZE (acceralated-KAZE) 特徴量 [2]

A-KAZE 特徴量とは、SIFT [3], SURF [4] 等と同様に画像から特徴点を検出する手法である。初期の頃は KAZE 特徴量という名称で局所的な部分の特徴点をより正確に抽出するという目的で開発された。しかし、特徴点を抽出する際に SIFT, SURF と比べると時間がかかり過ぎてしまい実用性が低かった。そこで弱点である時間を短縮するために生まれたのが A-KAZE 特徴量である。

A-KAZE 特徴量は SIFT, SURF が局所的な特徴量を取るのに向いていないと分かり、局所的な特徴量を取れるようにするという目的で作られた方法である。そのために SIFT, SURF ではコントラスト, 回転, 拡大縮小に対して不変性を持っているが, A-KAZE 特徴量では Blur (ぼかし) 耐性がコントラスト, 回転, 拡大縮小不変に追加される。

以下, 3.3.1 節~3.3.5 節で A-KAZE 特徴量ではどのような計算をしているのかを記述していく。

3.3.1 非線形拡散フィルタリング

非線形空間における明るさの拡散を表す偏微分方程式は一般的に下式で示される。

$$\frac{\partial L}{\partial t} = \text{div}(c(x, y, z) \cdot \nabla L) \quad (3.4)$$

このうち L は明るさ (luminance) を表し, 関数 c は conductivity function の頭文字で, 対象画像の構造に上式を対応させるための関数である。関数 c はスカラーでもベクトルでも良いが対象画像の微分構造に依存する。A-KAZE ではそれぞれのスケールレベルで勾配の大きさが拡散をコントロールしている領域に着目して, 画像のエッジ部分の拡散を抑える関数 c を定義している。下式に関数 c を示す。

$$c(x, y, t) = g(|\nabla L_\sigma(x, y, t)|) \quad (3.5)$$

$$g_1 = \exp\left(-\frac{|\nabla L_\sigma|^2}{k^2}\right) \quad (3.6)$$

$$g_2 = \frac{1}{1 + \frac{|\nabla L_\sigma|^2}{k^2}} \quad (3.7)$$

k は拡散レベルを制御するコントラストパラメータで任意の数に設定できる。 g_1 はハイコントラストのエッジを促進し, g_2 は小さなコントラストのエッジより広い領域を促進する。エッジの両側での平滑化がエッジ部分の平滑化よりもはるかに強く, 拡散を大幅に減少させるために拡散を異なるように設定している。選択的な平滑化としては局所的なぼかしより局所的な平滑化のほうが好ましいので, その関数を g_3 として下式のように定義する。

$$g_3 = \begin{cases} 1 & |\nabla_\sigma|^2 = 0 \\ 1 - \exp\left(-\frac{3.315}{(|\nabla L_\sigma|/k)^8}\right) & |\nabla_\sigma|^2 \neq 0 \end{cases} \quad (3.8)$$

コントラストパラメータ k は任意に設定できるが画像の評価の平均から自動的に算出することができる。各エッジは k の数値が大きくなるほど、大きな勾配を持つ部分が残され、弱いエッジが減少する。

3.3.2 AOS 手法

AOS 手法とは微分方程式を差分の形式で表し離散的にすることで PC で計算できるようにする手法である。偏微分方程式を離散化する手法としては線形の方式などがある。KAZE 特微量ではベクトル行列で表記された方程式を離散化する方法を使用し、その手法は下式で表される。

$$\frac{L^{i+1} - L^i}{\tau} = \sum_{l=1}^m A_l(L^i)L^{i+1} \quad (3.9)$$

A_l は各次元のための画像伝導率を行列で表している。また、 L^{i+1} は下式で表される。

$$L^{i+1} = \left(I - \tau \sum_{l=1}^m A_l(L^i) \right)^{-1} L^i \quad (3.10)$$

この式を解くために線形方程式が条件として必要であるが、ガウスの消去法によって解くことができる。

3.3.3 非線形スケール空間の計算

スケール空間を対数刻みで離散化し、一連の O オクターブ及び S サブレベルに配列する。オクターブとサブレベルのセットは、離散オクターブインデックス o とサブレベル s で識別され、オクターブインデックスとサブレベルインデックスは、対応するスケール σ に下式でマッピングされる。

$$\begin{aligned} \sigma_i(o, s) &= \sigma_0 2^{\frac{o+s}{S}} \\ o &\in [0 \cdots O-1] \\ s &\in [0 \cdots S-1] \\ i &\in [0 \cdots N] \end{aligned} \quad (3.11)$$

ここで σ_0 はベーススケールレベルであり、 N はフィルタされた画像の総数である。このように変換するのは、非線形拡散フィルタリングが時間的に定義されるためである。この変換を適用して一連の進化時間を求め、下式を用いてスケール空間 $\sigma_i(o, s)$ を時間単位に変換する。

$$t_i = \frac{1}{2} \sigma_i^2, \quad (i = 0, 1, \dots, N) \quad (3.12)$$

一般に、各フィルタ画像 t_i における非線形スケール空間では得られる画像は、標準偏差 σ_i のガウス分布を有する原画像の畳み込みに対応しない。しかし、拡散関数 g を 1 に等しく（すなわち、一定の関数）設定することで一致させることができる。

3.3.4 特徴点検出

関心のある点を検出するためには、複数のスケールレベルにおいてヘッセ行列のスケール正規化行列式の応答を計算する。多スケールの特徴検出を行うためには、微分演算子のセットをスケールに関して正規化する必要がある。下式は正規化する際に用いる微分演算子である。

$$L_{Hessian} = \sigma^2(L_{xx}L_{yy} - L_{xy}^2) \quad (3.13)$$

ここで (L_{xy}, L_{yy}) はそれぞれ2次の水平及び垂直導関数であり、 L_{xy} は2次交差導関数である。非線形スケール空間 L_i からフィルタ画像の集合を与えられたとき、異なるスケールレベル σ_i における検出器応答を分析する。現在の i 、上側 $i+1$ および下側 $i-1$ フィルタ画像上のサイズ $\sigma_i \times \sigma_i$ の矩形の窓を越えて、 $E=0$ および $i=N$ を除いて全てのフィルタリングされた画像において実行される。極値探索の高速化のために、最初に 3×3 ピクセルの大きさのウィンドウ上で応答を確認し、非最大の応答を破棄する。最後に特徴点の位置は、文献 [5] で提案された方法を用いて、サブピクセルレベルで推測できる。第1次導関数と第2次導関数の組は、異なる微分ステップサイズ σ_i の 3×3 Scharr フィルタによって近似される。2次導関数は、導関数の所望する座標で連続する Scharr フィルタを用いて近似される。これらのフィルタによって、他のポピュラーなフィルタよりも回転に対して強くなる。

3.3.5 特徴点について

支配的な方位を見つける

回転不変記述子を得るためには、特徴点の位置を中心とする支配的な方位の中心に位置することが必要である。SURF と同様に、半径 σ_i の円形領域内の支配的な方向を、サイズ σ_i のサンプリングステップで見つける。円形領域内のそれぞれのサンプルについて、1次導関数 L_x および L_y は、関心点を中心とするガウス関数によって重み付けされる。次に、微分応答はベクトル空間の点として表され、支配的な方向は、 $\pi/3$ の角度をカバーする滑らかな円セグメント内の応答を合計することによって見出される。最も長いベクトルから支配的な方位が得られる。

記述子の構築

非線形スケール空間フレームワークに適合した M-SURF 記述子を使用する。尺度 σ_i で検出された特徴に対して、サイズ σ_i の1次導関数 L_x および L_y は、 $24\sigma_i \times 24\sigma_i$ の長方形グリッドにわたって計算される。このグリッドは、 4×4 のサブ領域と $9\sigma_i \times 9\sigma_i$ に $2\sigma_i$ のオーバーラップで分割されている。各サブ領域内の出力応答は、サブ領域中央で計算されたガウス関数 ($\sigma_1 = 2.5\sigma_i$) によって重み付けされ、ベクトル記述子 $dv = (\sum L_x, \sum L_y, \sum |L_x|, \sum |L_y|)$ に合算される。次に、各サブ領域ベクトルは 4×4 のマスク上に定義されたガウス関数 ($\sigma_2 = 1.5\sigma_i$) を用いて重み付けされる。主点の特徴点の方向を考慮して、各領域のサンプルもその方向に従って回転する。さらに、

導関数もその方位に従って計算される。最終的に、長さ 64 の記述子ベクトルを単位ベクトルに正規化し、コントラストの不変性を得る。

第 4 章

評価実験

本章では，第 3 章で記述した 3 つの手法の実験結果と考察を記述していく．

4.1 実験環境

本研究で用意したデータベース内の画像はきのこ図鑑 [6] から 89 枚，テスト画像はネットから適当に検索したキノコの画像 40 枚を用意した．またデータベース内のキノコ画像 89 枚は全て異なる種類のキノコを使用し，テスト画像の 40 枚はその 89 種類の中から 40 種類を選び，同じ種類で異なる画像を用意した．

本研究ではそれぞれの手法で 40 枚のテスト画像をキノコデータベース内の画像 89 枚全てに類似度算出し，算出した結果から類似度の高い順に順位を付けていく．順位付けを行った後は 1 位，3 位以内，5 位以内，10 位以内，20 位以内，30 位以内のどこに入っているのかを確認し，それぞれの順位内でどれくらいの件数が入っているのかを，割合にして表示している．またグラフは 40 枚のキノコ画像が実際に何位に入っているのかを示している．

4.2 実験結果

表 4.1 それぞれの特徴量のみの順位結果

手法	1位	3位以内	5位以内	10位以内	20位以内	30位以内
手法 1:特徴点のみ	0.000	0.125	0.200	0.275	0.325	0.600
手法 2:RGB 空間	0.075	0.175	0.225	0.325	0.425	0.675
手法 2:HSV 空間	0.200	0.275	0.375	0.475	0.675	0.750

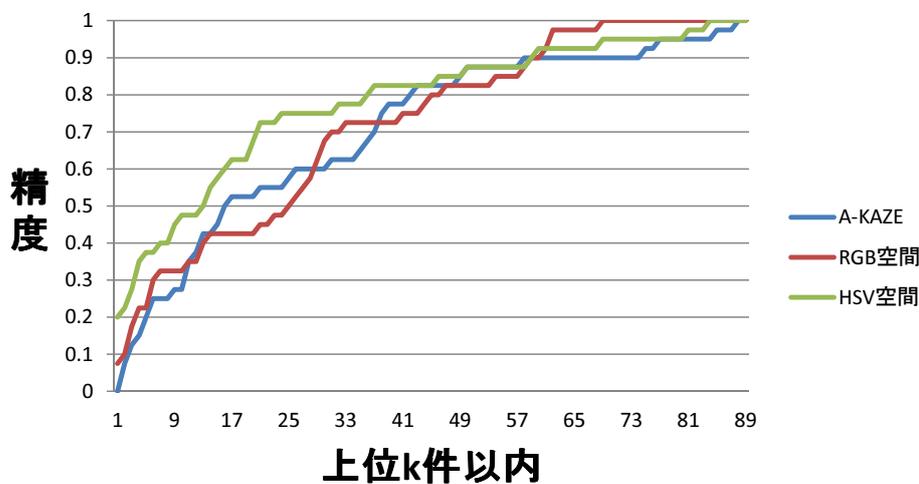


図 4.1 それぞれの特徴量のみの精度グラフ

表 4.1 と図 4.1 では、特徴点類似度と色ヒストグラム類似度それぞれの方法のみでテスト画像とデータベース内の画像に類似度算出を行い、データベース内の画像を算出結果から類似度が高い順に並べて正解画像が何位にあるかを調べた結果を表示している。

それぞれの特徴量のみの類似度算出を行うと、特徴点のみと RGB 空間については低い結果となった。3 位以内では 2 割を切っていて、30 位以内で特徴点のみでは 6 割、RGB 空間では 6 割を超える結果になった。HSV 空間については全ての順位において一番高い結果であることが分かる。

表 4.2 手法 3 における閾値ごとの順位結果 (RGB 空間)

閾値	1 位	3 位以内	5 位以内	10 位以内	20 位以内	30 位以内
0.5	0.025	0.150	0.275	0.450	0.550	0.600
0.6	0.025	0.225	0.230	0.400	0.525	0.600
0.7	0.050	0.175	0.225	0.275	0.325	0.400
0.8	0.075	0.150	0.225	0.275	0.350	0.400

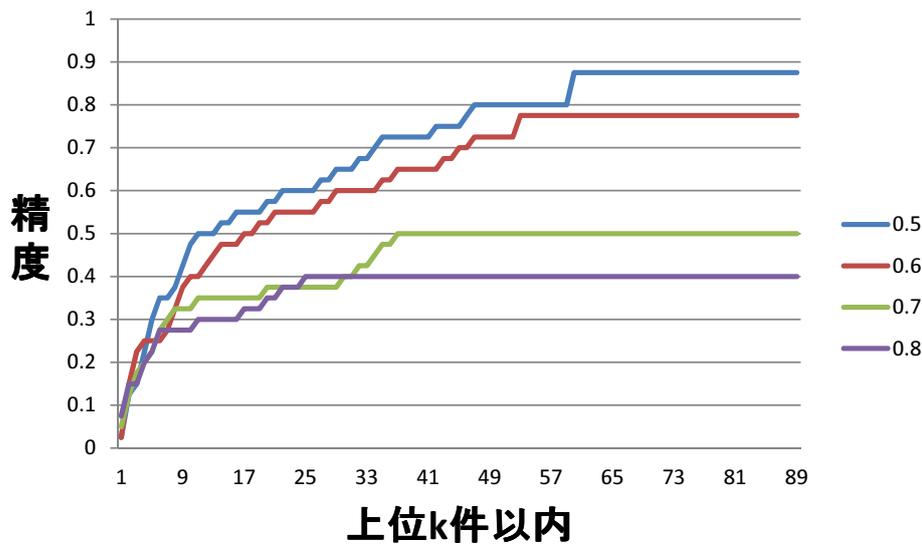


図 4.2 手法 3 における閾値ごとの精度グラフ (RGB 空間)

表 4.2 と図 4.2 では、RGB 空間において色ヒストグラム類似度に閾値を設定して閾値以上の画像に対して特徴点類似度算出を適用した結果を表示している。閾値は 0.5 から 0.1 刻みで増加し、0.8 で最も高い閾値に設定した。また、閾値を設定しているため閾値以下の画像はその時点で排除される。

図 4.2 のグラフから、閾値が 0.7 より高い時は 5 位以内までは大きく差は見られないが、10 位を超えると精度が大きく下がる結果となっている。また、閾値が 0.5 の時の精度が一番高い値になっている。手法 1 の特徴点のみを使った方法と比べると、閾値を用いることで閾値が 0.6 以下の 10 位以内と 20 位以内で精度が向上し、手法 1 では 1 位の精度が 0.000 であったが閾値を用いることによって 1 位での精度の向上が見られる。手法 2 の RGB 空間と精度が一番高かった閾値が 0.5 の時と比べると 1 位～5 位以内ではあまり変化を得られず、10 位以内と 20 位以内では精度が約 0.1 向上している。また $k = 1$ から 30 までの平均精度が RGB 空間のみでは 0.393、精度が最良の時の閾値を適用した結果の方は 0.475 であり、平均した結果から見ても閾値を用いた方が精度の向上が見られる。しかしながら、全体 ($k = 1$ から 89) の平均精度は RGB 空間のみでは 0.721、精度が最良の時の閾値を適用した結果の方では 0.702 であり、全体から見ると RGB 空間のみで類似度算出を行った方が良いという結果になっている。

表 4.3 手法 3 における閾値ごとの順位結果 (HSV 空間)

閾値	1 位	3 位以内	5 位以内	10 位以内	20 位以内	30 位以内
0.5	0.050	0.175	0.250	0.375	0.500	0.650
0.6	0.025	0.100	0.200	0.300	0.425	0.525
0.7	0.050	0.100	0.200	0.300	0.425	0.525
0.8	0.075	0.125	0.225	0.350	0.500	0.525

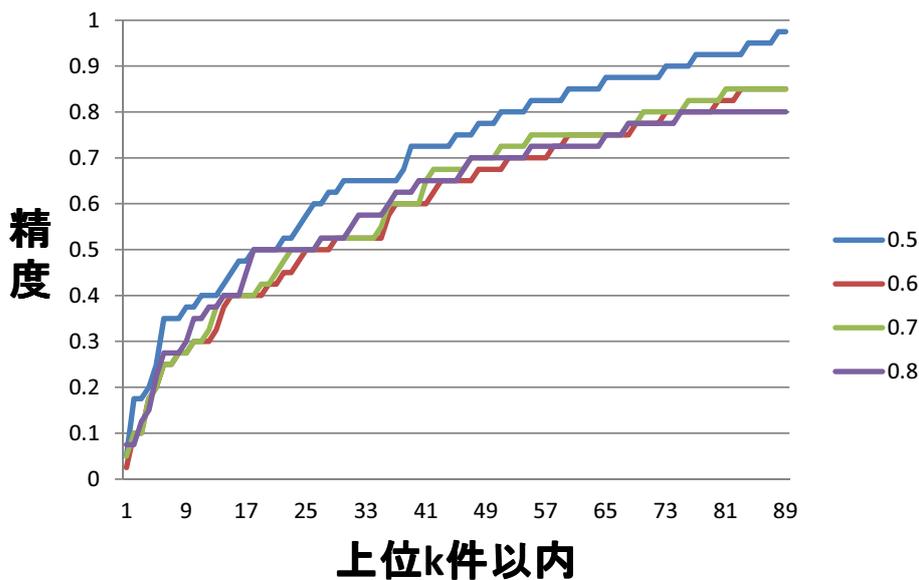


図 4.3 手法 3 における閾値ごとの精度グラフ (HSV 空間)

表 4.3 と図 4.3 では、HSV 空間において色ヒストグラム類似度に閾値を設定して閾値以上の画像に対して特徴点類似度算出を適用した結果を表示している。閾値は 0.5 から 0.1 刻みで増加し、0.8 で最も高い閾値に設定した。また、閾値を設定しているため閾値以下の画像はその時点で排除される。

HSV 空間に閾値を用いた結果は、閾値が 0.5 の時に精度が一番高い結果となった。また、RGB 空間に閾値を用いた結果と比べると 0.5 と 0.6 の時に RGB 空間で閾値を用いた結果の方が少しであるが精度が高くなっていて、閾値が 0.7 と 0.8 の時には HSV 空間で閾値を用いた結果の方が精度が高くなっている。RGB 空間では閾値が増加するにつれて全体的に結果が悪くなっているが、HSV 空間では閾値が増加しても変化しないという結果になった。RGB 空間で閾値を適用した結果では色ヒストグラム類似度算出をした時に正解画像との類似度が 0.5 ~ 0.6 の値が多いため、閾値が 0.7 以上から正解画像が多く排除されている。しかしながら、HSV 空間の方では色ヒストグラム類似度算出をした時に正解画像と類似度が高くなっているため、閾値を用いることで正解画像が排除されることが少なくなっている。手法 1 と比べると特徴点類似度算出結果とあまり変わらない結果となっているが、10 位以内と 20 位以内では精度を上げることが出来ている。手法 2 の HSV 空間と平均精度が一番高かった閾値が 0.5 の時

と比べると、全体の平均精度が HSV 空間のみでは 0.782, 閾値を適用した方の結果では 0.691 となり、まだ HSV 空間のみで類似度算出した方が良い結果となっている。また、RGB 空間に閾値を用いた結果の平均精度 0.702 と比べても HSV 空間に閾値を用いた結果の方が低い結果となっている。

4.3 考察

本研究の実験では、それぞれの特徴量のみを用いた結果は表 4.1 から低い結果となった。しかしながら、特徴点のみで類似度算出した結果を見ていくと、10 位以内の結果で得られたキノコは群生しているキノコが多いという結果であった。理由としては、群生しているキノコは単生しているキノコより局所的な特徴点を多く見つけることができ、その特徴点の中からより距離が近い特徴点を見つけていることができるので、群生しているキノコは上位に入りやすいからではないかと考えられる。逆に、単生のキノコは群生よりも異なる特徴点が少なく、他の単生のキノコと似ていると判断される。ただ、単生のキノコ全てが似ているわけではなくアミガサタケのような形状に特徴のあるものは例外として上位に位置する。

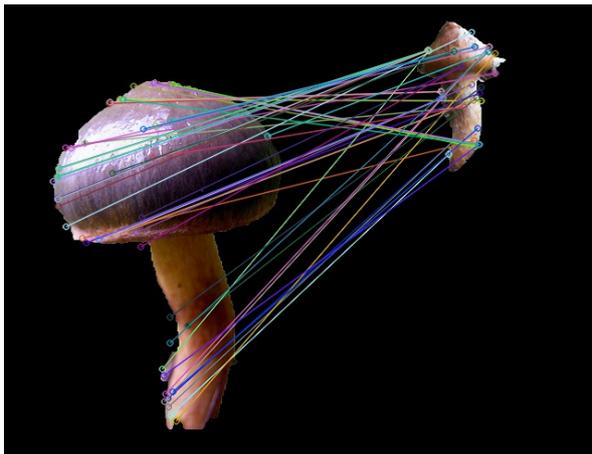


図 4.4 単生のキノコ画像の類似特徴点抽出結果

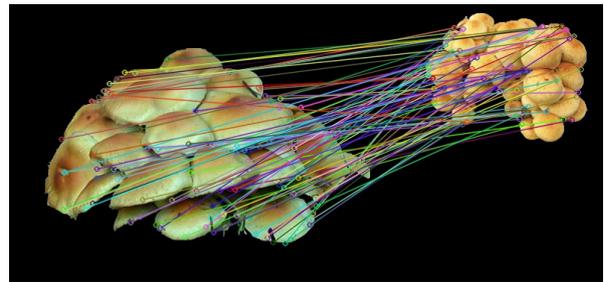


図 4.5 群生のキノコ画像の類似特徴点抽出結果

図 4.4 は単生のキノコの類似した特徴点同士を結んだ画像、図 4.5 は群生のキノコの類似した特徴点同士を結んだ画像である。例としてこの 2 つの画像を挙げる。図 4.4 では特徴点の個数が 47 個、図 4.5 では特徴点が 96 個検出された。このように単生と群生では検出された特徴点の個数に差があり、群生の方が特徴点は多い。他にも単生のキノコや群生のキノコを調べたが単生のキノコの特徴点の個数の平均が約 40 個、群生のキノコの特徴点の個数の平均が約 140 個であった。全体的に見ても群生の方が特徴点が多いことが分かる。また、上位に入っている単生のキノコは群生とまではいかないが多くの特徴点を抽出出来ていることが分かった。つまり、単生と群生にかかわらず多くの特徴点を抽出できるキノコ画像であるならば精度が良くなり、その中で群生のキノコは特徴点を多く抽出しやすいから、群生のキノコに対して精度が良くなっていると考えられる。

RGB 色空間については、赤などの派手な色が上位に入る傾向にある。一方、白などの色は閾値より小さい時が多い。理由としては、影が影響していると考えられる。RGB 色空間において色は赤、緑、青の3色の組み合わせで構成され、影という概念がないため RGB 色空間では影がかかると黒や黒に近い色と判断される。そのため、元から黒に近い色や、赤のような影の影響を受けにくい色は類似度にもあまり影響はないが、白色のキノコなどは影の影響を受けると色が全く別のものに変色し類似度が大きく下がる結果になると考えられる。一方、RGB 空間に比べ HSV 空間では影の影響を受けにくくなり白色のキノコも閾値よりも高くなった。また、蛍光色などの色も RGB 空間では低い結果になっていたが、HSV 空間では類似度も高くなり上位に位置することが多くなった。しかしながら、手法3の閾値を用いた結果では閾値が0.5の時だけではあるが RGB 空間の時の方が平均精度が高くなっていた。理由としては、閾値が0.5の時では RGB 空間では正解ではない画像の排除が出来ていて、HSV 空間では類似度が0.5より下の画像が少なく、正解ではない画像の排除がうまく出来ていないからであると考えられる。

第5章

まとめと今後の課題

本研究では、画像間の類似度算出として3種類の手法を用いてキノコ画像認識を行い、それぞれの手法の精度を確かめた。本研究の目的としては3位以内に9~10割の精度が出れば成功であったが、結果はどの手法も3割程度しか出せなかった。しかしながら、3つの手法をそれぞれ考察してみると色空間はHSV色空間を使用すると影の影響が受けにくいことが分かり、特徴点類似度算出を用いれば、群生と単生に分けることが出来る可能性があることが分かった。

本研究のキノコ画像認識の精度は低かった。理由としてはキノコ全体（背景は削除済み）を特徴点類似度算出で認識しようとする、ヒダなどのキノコの細かい特徴を捉えることが出来ないからであると考えられる。また、影の影響で色ヒストグラム類似度の精度に影響が出てしまうのは、キノコの柄にキノコの傘の影で覆ってしまうからであると考えられる。今後の課題としてはA-KAZE特徴量を使った結果から群生と単生に分けることが出来ると考えられるため、群生と単生に焦点を当てて、あらかじめ群生と単生の2つに分けることが出来るか検討していく必要があり、単生のキノコに対しては傘と柄に分けて、傘と傘、柄と柄にHSV空間で色ヒストグラム類似度算出と特徴点類似度算出を用いると精度を上げられると考えられる。また、キノコは形状や色の特徴だけではなくどんな場所に生えているのかという情報も判断する基準になっているため、背景情報を利用出来れば違った観点から推測できると考えられる。方法としては色による分類を検討している。緑や茶系が多いならば地上（地面）に生えていると分類し、灰色や黒が多いならば倒木や木に生えていると分類することで背景情報を得ようと考えている。

謝辞

本研究に際して、様々なご指導を頂きました服部峻助教を初めとして、服部研究室の皆様にご感謝を致します。また、実験に使った A-KAZE 特徴量の製作者の皆様にも感謝致します。そして、本研究で用いたキノコ画像の情報を載せて下さった方々に感謝致します。

参考文献

- [1] 健康かんぼうブログ, <http://kenkokampo.com/lifestyle/meal/mushroom/7042/>.
- [2] P. F. Alcantarilla, A. Bartoli and A. J. Davison, “KAZE Features,” European Conference on Computer Vision 2012, Part VI, Lecture Notes in Computer Science Vol.7577, pp.214–227, 2012.
- [3] Lowe. D, “Distinctive image features from scale-invariant keypoints,” International Journal of Computer Vision 60, pp.91–110, 2004.
- [4] Bay. H, Ess. A, Tuytelaars. T, Van Gool. L, “SURF: Speeded up robust features,” Computer Vision and Image Understanding 110, pp.346–359, 2008.
- [5] Brown. M, Lowe. D, “Invariant features from interest point groups.” In: British Machine Vision Conference, pp.253–262, 2002.
- [6] きのこ図鑑, <http://www.kinoco-zukan.net/>.