

「小学生と中学生の絵に違いはあるのか？」

～色の違いに着目して～

氏名：青木 雄大

所属学部学科（専攻）：自然科学研究科 環境科学専攻

1. 動機、背景、目的

2023年11月初旬、私は北海道開拓の村を訪れた。北海道開拓の村は2023年で40周年を迎え、今年は40周年を記念して、開村当時から行っている「北海道開拓の村児童写生会」の歴代特別賞受賞作品を展示していた[1]。私は第1回（1983年）から第39回（2023年）までの絵を見て、昔は濃く暗い色で描かれている絵が多く、現在にかけて薄く明るい色で描かれている絵が増えていると感じ、絵で用いられる色が昔から現在にかけて変化しているのが面白いと思った。これをきっかけに、私は「絵で用いられる色」の分析を試みたいと思った。

次に、今回のデータサイエンスコンテストで使用するデータの取得を考えた。北海道開拓の村児童写生会では、開拓の村の中にある建物から自分が描きたいものを描くため、昔から現在に至るまでに同じ建物について描かれることがしばしばある。私が「絵で用いられる色が昔から現在にかけて変化している」と感じたのも、同じ建物や似たような建物について描かれている絵を比較できたからである。そのため、「絵で用いられる色」を分析するために、私は同じものを対象に描いた絵がたくさんある絵画のコンテストをインターネットで検索し、その結果、灯台絵画コンテストを見つけた。このコンテストは灯台を対象に絵を描くもので、小学生と中学生の部がそれぞれある。このコンテストの絵を見て私は、同じものを対象としているのに、人によって絵を描くのに用いる色が異なるため面白いと思った。また、小学生の方が中学生よりも絵に用いる色が多いのではないかと感じた。

そこで、今回のデータサイエンスコンテストで私は、「絵で用いられている色に着目して小学生と中学生の絵の違い」を確かめた。

2. データについての説明

今回のデータサイエンスコンテストでは、公益社団法人燈光会が開催する「灯台絵画コンテスト」の2005年から2023年までの絵をデータ分析に用いた[2]。データの分析には、小学生の作品計323点、中学生の作品計161点、総数484点を用いた。これらのデータを分析に用いた理由は、1. 動機、背景、目的でも挙げているように、同じもの（灯台）を対象に描かれた絵がたくさんあったため、色の違いのみに着目しやすと考えたからである。データはWebスクレイピングによって取得した。Webスクレイピングによってデータを取得する前に、灯台絵画コンテストの作品が掲載されているhtmlを解析し、データを読み取る関数(get_picture)を作成した。Webスクレイピングで取得できなかったデータのみ、手作業でダウンロードした。データの取得後、各年のディレクトリ内で小学生と中学生のディレクトリを作成し、小学生の絵と中学生の絵を分類した。

3. 分析についての説明と結果・考察

本研究では、小学生の絵と中学生の絵それぞれに使われた色を分析するために、機械学習の一種であるクラスタリングを用いた。クラスタリングとは、データ間の類似度に基づいてデータをグループ分けしていく手法のことである。クラスタリングにより、小学生の絵と中学生の絵それぞれでよく使われた色をグループ分けした。

分析にあたり、最初に小学生の絵と中学生の絵すべての絵の解像度を 100dpi に変更した。すべての絵の解像度を一致させることで、クラスタリングの際、絵の解像度によるグループ分けの違いを生まないようにした。解像度の調整後、絵の色空間を RGB から HSV に変更した。RGB は赤、緑、青の 3 要素の原色の組み合わせで色を表現する方法である。一方、HSV は色を色相、彩度、明度の 3 要素で表現する方式である。HSV 色空間では、鮮やかさや明るさといった人間の色の認識と似た方法で色を表現している。そのため、HSV の方が RGB に比べて小学生と中学生によって描かれた絵を詳細に分析できると考え、HSV に変更した。

色空間の変更後、クラスタリングにより、各絵の色をグループ分けした。データのクラスタリングには、クラスタ分析で一般的に使用される k-means を使用した。k-means 法は最初に指定したクラスタ（グループの数）だけ、重心をランダムに指定し、その重心を基準にクラスタを分けていくという手法である。クラスタリングをするにあたり、最初に小学生の絵 323 点と中学生の絵 161 点それぞれの最適なクラスタ数を調べた。各絵の最適なクラスタ数を調べるために、Calinski-Harabasz クラスタリング評価基準を使用した。この評価基準は、比較的計算効率が良く、また、直感的で数値的な評価が容易な基準である。そのため、データ数の多い今回の研究には、Calinski-Harabasz クラスタリング評価基準が適していると考え採用した。また、今回、クラスタ数の上限は、ぺんてるのパレットを参考にして 25 と設定した。ぺんてるパレット [3] には 15 色シールが貼られている。そして、絵具で色を作る際は、そういった色を組み合わせで作る。今回の研究では、絵具を混ぜて作る色の数を 10 色と考えて、15 色に 10 色を加えた 25 をクラスタ数の上限とした。クラスタ数は多すぎても少なすぎても絵に使われた色の特徴を調べるのに適さないため、この上限は適切であると考えられる。

最適なクラスタ数を決定した後、クラスタリングにより各絵の色をグループ分けした。k-means の最大反復回数は 1000 回とした。k-means アルゴリズムのデフォルトの反復回数は 100 回であるが、最初に 100 回で実行したとき、最適なクラスタリング結果に到達しなかった。そこで、最大反復回数を 1000 回としたところ最適なクラスタリング結果に到達したため、今回の研究では最大反復回数を 1000 回とした。

図 1 は小学生と中学生それぞれの各絵のクラスタ数を学年ごとに調べた散布図である。ここでのクラスタ数は、各絵に使われた色をグループに分けたときのグループの数を意味している。そのため、この図は小学生と中学生それぞれの学年に関わらず、各絵に 2 から 25 までの色のグループがまんべんなく存在することを示している。

図 2, 3 は小学生と中学生それぞれの各絵のクラスタ数上位 10 個の割合を表示した円グラフである。クラスタ数が大きな値に着目して分析する。小学 1 年生は最も多いクラスタ数は 25、2 番目に多いクラスタ数は 11、2 年生は 3 番目に多いクラスタ数が 18 である。また、3 年生は 7 番目に 24、4 年生は 5 番目に 13、5 年生は 4 番目に 10、6 年生は 7 番目に 12 が入っている。中学生に関しては、中学 1 年生は 6 番目に 21、5 番目に 21、5 番目に 25 が入っている。そのため、この結果から、年齢が小さい方が大きい方に比べて絵に用いられた色の数が多いことが分かる。年齢が小さい子供の方が年齢が大きい方と比べて使われた色が多い（クラスタ数が大きい）理由として、私は年齢が小さい子供の方が様々なものを描いたからだと考える。例えば、小学 1 年生や 2 年生は、灯台そのものもちろん描いているが、それに加えて家族の絵も描いてい

る。一方で、年齢が上がるにつれて、灯台とともに人を描いている絵が減っており、灯台そのものに焦点を当てた絵が増えている。人を描くということはそれに付随して洋服なども描かれるため、人が多い絵には使われている色が多くなる。そのため、人も描いている絵が多い小学生の方が中学生に比べてクラスター数が大きくなったと考えられる。

図 4, 5, 6, 7 は小学生と中学生それぞれの各絵に使われた色を分類した後に、分類した各絵の色（色相、彩度、明度の値）をまとめてプロットした図である。小学生の方が中学生に比べて作品数が多いということもあり、プロットされた点も多いが、小学生の絵の方が色相の値が大きく、明度が高い赤やピンクなどの色が多く使われている。一方で、明度が低い色（黒に近い色）に関しては、小学生と中学生の間で使われた色の数に大きな違いは見られなかった。このことから、暗い色は中学生の方が小学生よりも使っていると考えられる。また、色相と彩度の関係から、小学生の方が中学生よりも様々な青や緑の色を使っていることが分かる。この結果からも小学生の方が中学生よりも使う色が多いことが分かる。この結果から、小学生は絵を描く際に対象とするもの（今回は灯台）だけでなく、周りにあるものを重視する傾向にあると考えられる。例えば、小学生は絵の中に家族や自分を描いており、絵を描くのと同時に家族と一緒に同じ場所・時間を共有することを楽しんでいると考えられる。一方で、中学生は対象とするものに焦点を当てて絵を描いており、絵を描くことそのものを楽しんでいると考えられる。

また、これらの図から、小学生と中学生に共通する全体的な特徴として、色相、彩度、明度に関して小学生と中学生で同じような傾向を示していることが分かる。これは、たくさんの絵を解析したことで、データのばらつきが小さくなり、データの特徴が表れたからだと考えられる。よって、この結果から灯台によく用いられる色を推測するのは統計学的に問題ないと考えられるので、ここで灯台の絵によく使われる色やそうでない色を推測する。まず、絵を描く人によって使う色は異なるが、色相が 0.7 付近の色は灯台の絵を描くときには用いられることが少ないと考えられる。また、色相が 0.2 から 0.3 で彩度が 0.6 以上の色や、明度が低い色は、灯台の絵に用いられることが少ないと考えられる。そして、色相が 0.5 から 0.6 の色（青など）は灯台の絵に用いられることが多いと考えられる。

図 8, 9 はそれぞれ小学生と中学生の各絵に使われた色をクラスタリングした後に、その結果を再びクラスタリングした結果を表す（図 4, 5, 6, 7 でプロットした点をクラスタリングした）。ここでのクラスター数の上限は 15 とした。この結果から、クラスタリングした後、再びクラスタリングすると、小学生と中学生の絵に使われた色がほとんど同じであることが分かる。1 度目のクラスタリングによる結果は図 4, 5, 6, 7 から確認できるが、これらも小学生と中学生の絵に使われた色は全体的に同じような傾向を示している。2 度目のクラスタリングはこれらの結果をクラスタリングをしたものであるため、より似たような色のグループが作られたと考えられる。全体として青系統や白系統の色が多いが、これらの色は灯台や海、空の色であると考えられる。また、赤系統の色や橙色があるが、これらの色は夕日を描いた絵の影響を受けたものであると考えられる。緑系統の色は木や芝生など自然を描いた絵によるものと考えられる。また、中学生の方が似たような色でもやや異なる色（例えば青系統の色）でグループ分けされている数が多いように見える。これは中学生の方が色の繊細な違いに気づいたり、色を作る技術が上達したりしているからだと考えられる。

4. 結論

今回の分析の結果から、小学生と中学生でそれぞれが絵に用いる色は小学生の方が中学生よりも多いことが分かり、絵で用いられる色は小学生と中学生で異なることを確かめることができた。そのため、小学生と中学生の絵には色の違いがあることが分かり、本研究の目的を達成できた。このような結果になった理由として、例えば、小学生は絵の中に家族や自分など様々なものを描いている一方で、中学生は対象とするものに焦点を

当てて絵を描いているため、小学生の方が中学生よりも絵に用いる色が多くなったと考えられる。また、この背景として、小学生は絵を描くのと同時に家族と一緒に同じ場所・時間を共有することを楽しんでおり、中学生は絵を描くことそのものを楽しんでいると考えられる。

また、今回の結果から、同じものを対象とした絵の場合、データ分析に用いるサンプル数を多くすることで、対象とするものが描かれる際に一般的に多く用いられる色を推定できると考えられる。今回の結果は、灯台の絵を描く際には青が多く用いられることを示した。

今後に向けた改善点として、比較に用いるデータ数を揃えることが挙げられる。今回のデータ分析では小学生と中学生の絵の色の分析にそれぞれ、323点、161点の絵を用いたが、これが揃っていないことにより散布図を作成する際にプロットする点の数が揃わなかった。そのため、小学生と中学生の絵に使われる色をより正確に比較するために、今後、比較をするデータの数は揃えたいと思う。

最後に、今回のデータ分析を通して様々な絵を見たが、どれもその年齢でしか描くことができない価値のある絵であると感じた。そのため、年齢に応じて楽しんで絵を描くことが大切であると思う。

5. 参考文献・参考 URL

[1] 北海道開拓の村：児童写生会 (<https://www.kaitaku.or.jp/event/%E5%85%90%E7%AB%A5%E5%86%99%E7%94%9F%E4%BC%9A%E3%80%80%E6%AD%B4%E4%BB%A3%E7%89%B9%E5%88%A5%E8%B3%9E%E4%BD%9C%E5%93%81%E5%B1%95/>)

[2] 橙光会：灯台絵画コンテスト (<https://www.tokokai.org/picture/%e7%b5%b5%e7%94%bb%e3%82%b3%e3%83%b3%e3%83%86%e3%82%b9%e3%83%88/>)

[3] ペんてるパレット (<https://www.pentel.co.jp/products/artmaterials/pentel-palette/>)

6. 感想

今回のデータサイエンスコンテストを通して、今まで使ったことのなかった様々な matlab の関数を使うことができたので良かった (webread, pie, histcounts, kmeans, evalclusters など)。今後、私は写真からも今回のような分析を行えるようにして、様々なデータの分析をできるようにしたい。また、今後も使ったことのない様々な matlab の関数を使い続けて matlab の技術を磨き、頭の中でひらめいたことの具現化や、面白いと思ってもらえるデータ分析をしたい。

```
%%%%%%%%%%
clear
clc
% データの URL。2021 年から 2023 年までは、2005 年から 2020 年の URL と形が異なるため別に変数
% を作成し、後で変数をまとめる。
url2023 = [cellstr('https://www.tokokai.org/picture/
2023%e5%b9%b4%e3%81%ae%e5%8f%97%e8%b3%9e%4%bd%9c%e5%93%81/'), 2023];
url2022 = [cellstr('https://www.tokokai.org/picture/
2022%e5%b9%b4%e3%81%ae%e5%8f%97%e8%b3%9e%4%bd%9c%e5%93%81/'), 2022];
```

```

url2021 = [cellstr('https://www.tokokai.org/picture/
2021%e5%b9%b4%e3%81%ae%e5%85%a5%e8%b3%9e%e4%bd%9c%e5%93%81-2/'), 2021];
% https://www.tokokai.org/picture/pic ○○/ ...
url2020_2005temp = 'https://www.tokokai.org/picture/pic';
url2020_2005_year = (2020:-1:2005)';
url2020_2005 = cell(length(url2020_2005_year),2);
for a = 1:length(url2020_2005_year)
    url2020_2005{a,1} =
string(strcat(url2020_2005temp,num2str(url2020_2005_year(a,1))));
    url2020_2005{a,2} = url2020_2005_year(a,1);
end

```

% このプログラム内で次の変数をデータの取得に用いる。

```

url_use = [url2023; url2022; url2021; url2020_2005]; % 1行目は2023。最後の行が2005年
の順となっている。
number_url = size(url_use,1);

```

```

clear url2020_2005 url2020_2005_year url2020_2005temp url2021 url2022 url2023

```

```

data = cell(number_url,2);
for a = 1:number_url
    data{a,1} = webread(url_use{a,1});
    data{a,2} = url_use{a,2};
end

```

% ここでデータを取得する。

```

save_data_dir = './data/'; %
for a = 1:number_url
get_picture(url_use{a,1},url_use{a,2},save_data_dir)
end

```

```

%%%%%%%%%%

```

```

%%%%%%%%%%

```

% データが入っているディレクトリを設定

```

data_dir = './data/';
ES_dir = '/小学生/';
JS_dir = '/中学生/';
jpg_dir = '*.jpg';

```

% 読み取る画像名を取得

```

for a = 1:number_url
    data_dir_temp = strcat(data_dir,num2str(url_use{a,2}));
    data_dir_temp_ES = strcat(data_dir_temp,ES_dir,jpg_dir);
    data_dir_temp_JS = strcat(data_dir_temp,JS_dir,jpg_dir);

    filelist_es_temp = dir(data_dir_temp_ES);
    filelist_js_temp = dir(data_dir_temp_JS);
    if a ==1

```

```

        filelist_es = filelist_es_temp;
        filelist_js = filelist_js_temp;
    else
        filelist_es = [filelist_es; filelist_es_temp]; % 小学生の絵の名前のリスト
        filelist_js = [filelist_js; filelist_js_temp]; % 中学生の絵の名前のリスト
    end
end
es_number = length(filelist_es); % 小学生の絵の数
js_number = length(filelist_js); % 中学生の絵の数

dpi_picture = 100; % change

% 小学生の画像を読み取る。
for a = 1:es_number
    % 画像の読み込み
    imagePath_es = strcat(filelist_es(a,1).folder, '/', filelist_es(a,1).name); % 画像
    ファイルのパスを指定。
    img_es = imread(imagePath_es);
    resizedImg_es = imresize(img_es, [dpi_picture, dpi_picture]); % 画像サイズを
    100x100に変更
    img_es = im2double(resizedImg_es);

    % RGB から HSV へ変換
    img_es = rgb2hsv(img_es);

    % 画像を 3 次元配列から 2 次元配列にする。
    [rows, cols, ~] = size(img_es);
    data_to_k_es = reshape(img_es, rows*cols, 3);

    if a == 1
        data_to_k_es_all = data_to_k_es;
    else
        data_to_k_es_all = [data_to_k_es_all; data_to_k_es];
    end
    data_to_k_es_save{a,1} = data_to_k_es; % 小学生のそれぞれの絵の RGB
    imagepath_es{a,1} = imagePath_es; % 小学生の絵の名前、パス
    resizedImg_es_save{a,1} = resizedImg_es; % 小学生のそれぞれの絵
end
clear data_to_k_es
data_to_k_es = data_to_k_es_all; % 小学生の絵全体の RGB 配列

% 中学生の画像を読み取る。
for a = 1:js_number
    % 画像の読み込み
    imagePath_js = strcat(filelist_js(a,1).folder, '/', filelist_js(a,1).name); % 画像
    ファイルのパスを指定。
    img_js = imread(imagePath_js);
    resizedImg_js = imresize(img_js, [dpi_picture, dpi_picture]); % 画像サイズを
    100x100に変更
    img_js = im2double(resizedImg_js);

```

```

% RGB から HSV へ変換
img_js = rgb2hsv(img_js);

% 画像を 3 次元配列から 2 次元配列にする。
[rows, cols, ~] = size(img_js);
data_to_k_js = reshape(img_js, rows*cols, 3);

if a == 1
    data_to_k_js_all = data_to_k_js;
else
    data_to_k_js_all = [data_to_k_js_all; data_to_k_js];
end
data_to_k_js_save{a,1} = data_to_k_js; % 中学生のそれぞれの絵の RGB
imagepath_js{a,1} = imagePath_js; % 中学生の絵の名前、パス
resizedImg_js_save{a,1} = resizedImg_js; % 中学生のそれぞれの絵
end
clear data_to_k_js
data_to_k_js = data_to_k_js_all; % 中学生の絵全体の RGB 配列
% imshow(resizedImg_es_save{2,1})

% 学年の読みとり
es_grade_tmp = cell(es_number,1);
js_grade_tmp = cell(js_number,1);
es_grade = zeros(es_number,1);
js_grade = zeros(js_number,1);

% 小学生
pat_es = digitsPattern;
for a = 1:es_number
    es_grade_tmp{a,1} = extract(filelist_es(a).name,pat_es);
    if es_grade_tmp{a,1}{1,1} == " 1 "
        es_grade_tmp{a,1}{1,1} = "1";
    elseif es_grade_tmp{a,1}{1,1} == " 2 "
        es_grade_tmp{a,1}{1,1} = "2";
    elseif es_grade_tmp{a,1}{1,1} == " 3 "
        es_grade_tmp{a,1}{1,1} = "3";
    elseif es_grade_tmp{a,1}{1,1} == " 4 "
        es_grade_tmp{a,1}{1,1} = "4";
    elseif es_grade_tmp{a,1}{1,1} == " 5 "
        es_grade_tmp{a,1}{1,1} = "5";
    elseif es_grade_tmp{a,1}{1,1} == " 6 "
        es_grade_tmp{a,1}{1,1} = "6";
    end
    es_grade(a,1) = str2double(es_grade_tmp{a,1}{1,1});
end

% 中学生
pat_js = digitsPattern;
for a = 1:js_number

```

```

js_grade_tmp{a,1} = extract(filelist_js(a).name,pat_js);
if js_grade_tmp{a,1}{1,1} == " 1 "
    js_grade_tmp{a,1}{1,1} = "1";
elseif js_grade_tmp{a,1}{1,1} == " 2 "
    js_grade_tmp{a,1}{1,1} = "2";
elseif js_grade_tmp{a,1}{1,1} == " 3 "
    js_grade_tmp{a,1}{1,1} = "3";
end
js_grade(a,1) = str2double(js_grade_tmp{a,1}{1,1});
end
%%%%%%%%%%%%
%%%%%%%%%%%%
% 各絵の分類に用いる最大クラスター数の設定。
k_max = 25; % change

% kmeans 関数の設定 % 1000:最大反復回数 5:初期クラスター位置決定のための反復回数。
myfunc = @(X,K)(kmeans(X, K,
'EmptyAction', 'singleton', "MaxIter",1000, 'Replicates',5));

% 小学生の絵のクラスタリング
OptimalK_es = zeros(es_number,1);
cluster_idx_es_save = cell(es_number,1);
cluster_center_es_save = cell(es_number,1);
parfor a = 1:es_number
    %disp(strcat("Number of picture = ",num2str(a), " / ", num2str(es_number)))
    % 乱数発生器をリセット
    rng('default'); % 同じシードを使用
    eva_es = evalclusters(data_to_k_es_save{a,1}, myfunc,...
        'CalinskiHarabasz', 'KList',1:k_max);
    OptimalK_es(a,1) = eva_es.OptimalK; % 小学生の各絵の最適なクラスター数

    % 各絵の最適なクラスター数による k-means クラスタリング
    k = eva_es.OptimalK; % クラスターの数
    [cluster_idx, cluster_center] = kmeans(data_to_k_es_save{a,1}, k,...
        'EmptyAction', 'singleton', "MaxIter",1000, 'Replicates',5);

    % cluster_idx は各ピクセルのクラスターの番号。クラスターの番号は各クラスターに対応する。
    cluster_idx_es_save{a,1} = cluster_idx;
    cluster_center_es_save{a,1} = cluster_center; % cluster_center はクラスターの重心位置。
end
NumberOfClusters_es_mean = ceil(mean(OptimalK_es)); % 各絵の色のクラスター数の平均 (小学生の絵)

save('cluster_center_es_save.mat','cluster_center_es_save')
save('OptimalK_es.mat','OptimalK_es')

% 中学生の絵のクラスタリング
OptimalK_js = zeros(js_number,1);

```

```

cluster_idx_js_save = cell(js_number,1);
cluster_center_js_save = cell(js_number,1);
parfor a = 1:js_number
    %disp(strcat("Number of picture = ",num2str(a), " / ", num2str(js_number)))
    % 乱数発生器をリセット
    rng('default'); % 同じシードを使用
    eva_js = evalclusters(data_to_k_js_save{a,1}, myfunc,...
        'CalinskiHarabasz', 'KList', 1:k_max);
    OptimalK_js(a,1) = eva_js.OptimalK; % 中学生の各絵の最適なクラスター数

    % 各絵の最適なクラスター数による k-means クラスタリング
    k = eva_js.OptimalK; % クラスターの数
    [cluster_idx, cluster_center] = kmeans(data_to_k_js_save{a,1}, k,...
        'EmptyAction', 'singleton', "MaxIter", 1000, 'Replicates', 5);

    % cluster_idx は各ピクセルのクラスターの番号。クラスターの番号は各クラスターに対応する。
    cluster_idx_js_save{a,1} = cluster_idx;
    cluster_center_js_save{a,1} = cluster_center; % cluster_center はクラスターの重心位置。
end
NumberOfClusters_js_mean = ceil(mean(OptimalK_js)); % 各絵の色のクラスター数の平均 (中学生の絵)

save('cluster_center_js_save.mat', 'cluster_center_js_save')
save('OptimalK_js.mat', 'OptimalK_js')

%%%%%%%%%%%%
%%%%%%%%%%%%
% データ (RGB 空間での小学生の絵の色の 3D プロット、小学生の絵全体のクラスター分析に用いるデータ)。
for a = 1:es_number
    if a == 1
        plot_data_es_and_to_k = cluster_center_es_save{a,1};
    else
        plot_data_es_and_to_k = [plot_data_es_and_to_k; cluster_center_es_save{a,1}];
    end
end

plot_data_es_rgb = hsv2rgb(plot_data_es_and_to_k);
% データ (RGB 空間での中学生の絵の色の 3D プロット、中学生の絵全体のクラスター分析に用いるデータ)。
for a = 1:js_number
    if a == 1
        plot_data_js_and_to_k = cluster_center_js_save{a,1};
    else
        plot_data_js_and_to_k = [plot_data_js_and_to_k; cluster_center_js_save{a,1}];
    end
end
end

```

```

plot_data_js_rgb = hsv2rgb(plot_data_js_and_to_k);

%%%%%%%%%%%%
%%%%%%%%%%%%
% 各学年のクラスター数
% 小学生1年から6年まで
OptimalK_es_each_grade = cell(6,1);
OptimalK_js_each_grade = cell(3,1);
OptimalK_es_max_each_grade = zeros(6,1);
OptimalK_js_max_each_grade = zeros(3,1);
total_optimalK_counts_es_each_grade = cell(6,1);
total_optimalK_counts_js_each_grade = cell(3,1);
length_cluster_for_pie_graph_es = zeros(6,1);
length_cluster_for_pie_graph_js = zeros(3,1);
cluster_ratios_each_grade_es = cell(6,1);
cluster_ratios_each_grade_js = cell(3,1);
sorted_ratios_es_each_grade = cell(6,1);
sorted_ratios_js_each_grade = cell(3,1);
sorted_labels_es_each_grade = cell(6,1);
sorted_labels_js_each_grade = cell(3,1);

for a = 1:6
    c = 0;
    for b = 1:es_number
        if es_grade(b,1) == a
            c = c + 1;
            OptimalK_es_each_grade_tmp(c,1) = OptimalK_es(b,1);
        end
    end
    OptimalK_es_each_grade{a,1} = OptimalK_es_each_grade_tmp;
    OptimalK_es_max_each_grade(a,1) = max(OptimalK_es_each_grade_tmp);
    total_optimalK_counts_es_each_grade{a,1} =
histcounts(OptimalK_es_each_grade_tmp, 1:(OptimalK_es_max_each_grade(a,1) + 1));
    length_cluster_for_pie_graph_es(a,1) = length(OptimalK_es_each_grade_tmp); % 各
学年の絵の数
    cluster_ratios_each_grade_es{a,1} = total_optimalK_counts_es_each_grade{a,1} /
length_cluster_for_pie_graph_es(a,1);
    % 割合に基づいてクラスターを並べ替え
    labels_es_each_tmp = string(1:1:OptimalK_es_max_each_grade(a,1));
    [sorted_ratios_es_each_grade_tmp, sort_order_es_each_grade] =
sort(cluster_ratios_each_grade_es{a,1});
    sorted_ratios_es_each_grade{a,1} = sorted_ratios_es_each_grade_tmp;
    sorted_labels_es_each_grade{a,1} =
labels_es_each_tmp(:,sort_order_es_each_grade); % 割合を小さい順に並べた時のラベル
    clear OptimalK_es_each_grade_tmp labels_es_each_tmp
end

for a = 1:3
    c = 0;
    for b = 1:js_number

```

```

    if js_grade(b,1) == a
        c = c + 1;
        OptimalK_js_each_grade_tmp(c,1) = OptimalK_js(b,1);
    end
end
OptimalK_js_each_grade{a,1} = OptimalK_js_each_grade_tmp;
OptimalK_js_max_each_grade(a,1) = max(OptimalK_js_each_grade_tmp);
total_optimalK_counts_js_each_grade{a,1} =
histcounts(OptimalK_js_each_grade_tmp, 1:(OptimalK_js_max_each_grade(a,1) + 1));
length_cluster_for_pie_graph_js(a,1) = length(OptimalK_js_each_grade_tmp); % 各
学年の絵の数
cluster_ratios_each_grade_js{a,1} = total_optimalK_counts_js_each_grade{a,1} /
length_cluster_for_pie_graph_js(a,1);
% 割合に基づいてクラスターを並べ替え
labels_js_each_tmp = string(1:1:OptimalK_js_max_each_grade(a,1));
[sorted_ratios_js_each_grade_tmp, sort_order_js_each_grade] =
sort(cluster_ratios_each_grade_js{a,1});
sorted_ratios_js_each_grade{a,1} = sorted_ratios_js_each_grade_tmp;
sorted_labels_js_each_grade{a,1} =
labels_js_each_tmp(:,sort_order_js_each_grade); % 割合を小さい順に並べた時のラベル
clear OptimalK_js_each_grade_tmp labels_js_each_tmp
end

```

```

%%%%%%%%%%
%%%%%%%%%%
% 散布図の点のサイズ
plot_size = 20;

%% クラスター数の散布図
% 小学生
figure('Position', [100 100 990 280]); % [left bottom width height]
f_cluster_es_js = tiledlayout(1,2);
nexttile
scatter(es_grade,OptimalK_es,plot_size);
title("小学生の絵に使われた色の数")
xlabel('学年');
ylabel('クラスター数');
xlim([0.5 6.5])
xticks([1 2 3 4 5 6])
xticklabels(["1年生","2年生","3年生","4年生","5年生","6年生"])
ax = gca; ax.BoxStyle = 'full'; ax.FontSize = 10;
grid off; box on

% 中学生
%f_cluster_js = figure('Position', [100 100 990 280]); % [left bottom width height]
nexttile
scatter(js_grade,OptimalK_js,plot_size)
title("中学生の絵に使われた色の数")
xlabel('学年');

```

```

ylabel('クラスター数');
xlim([0.5 3.5])
xticks([1 2 3])
xticklabels(["1年生","2年生","3年生"])
ax = gca; ax.BoxStyle = 'full'; ax.FontSize = 10;
grid off; box on

```

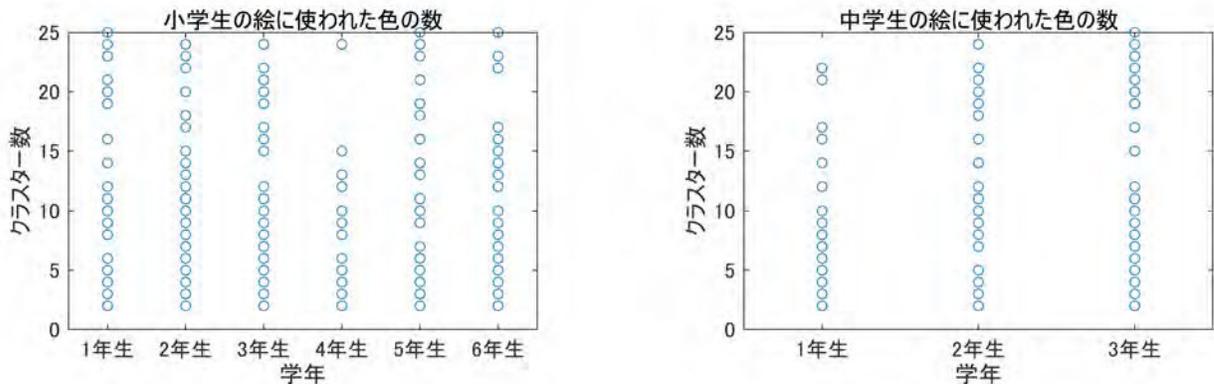


図1 小学生と中学生それぞれの各絵のクラスター数を学年ごとに調べた散布図

```

%%%%%%%%%%
%%%%%%%%%%
% 各学年のクラスター数の円グラフ
% 円グラフに表示させる数
pie_num = 10;

figure('Position', [100 100 500 800]); % [left bottom width height]
f_cluster_pie_es = tiledlayout(3,2,'TileSpacing','tight');
for a = 1:6
    nexttile
    p = pie(sorted_ratios_es_each_grade{a,1}(1,end-pie_num+1:end));
    pText = findobj(p,'Type','text');
    percentValues = get(pText,'String');
    txt = strcat((sorted_labels_es_each_grade{a,1}(1,end-pie_num+1:end))',' : ');
    combinedtxt = strcat(txt,percentValues);
    for b = 1:length(txt)
        pText(b).String = combinedtxt(b);
    end
    title(strcat(num2str(a),'年生'))
    clear pText percentValues txt combinedtxt
end
title(f_cluster_pie_es,'小学生の絵のクラスター数（上位10個）の円グラフ')

```

小学生の絵のクラスター数(上位10個)の円グラフ

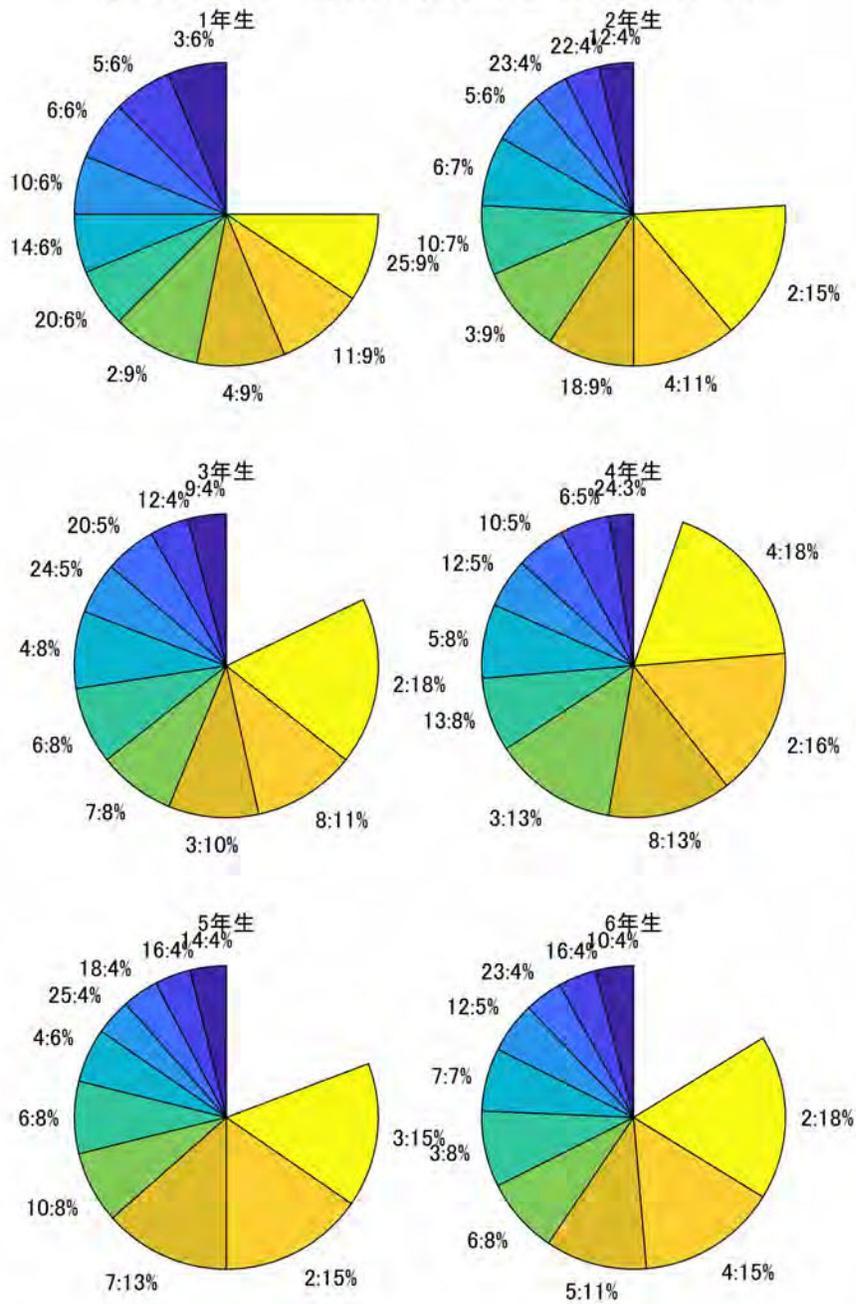


図2 小学生の各絵のクラスター数上位10個の割合を調べた円グラフ

```
figure('Position', [100 100 500 800]); % [left bottom width height]
f_cluster_pie_js = tiledlayout(3,1,'TileSpacing','tight');
```

```
for a = 1:3
    nexttile
    p = pie(sorted_ratios_js_each_grade{a,1}(1,end-pie_num+1:end));
    pText = findobj(p, 'Type', 'text');
    percentValues = get(pText, 'String');
    txt = strcat((sorted_labels_js_each_grade{a,1}(1,end-pie_num+1:end))', ': ');
    combinedtxt = strcat(txt,percentValues);
    for b = 1:length(txt)
        pText(b).String = combinedtxt(b);
    end
    title(strcat(num2str(a), '年生'))
    clear pText percentValues txt combinedtxt
end
title(f_cluster_pie_js, '中学生の絵のクラスター数（上位10個）の円グラフ')
```

中学生の絵のクラスター数(上位10個)の円グラフ

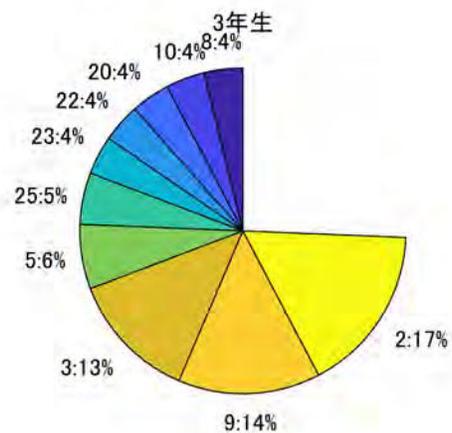
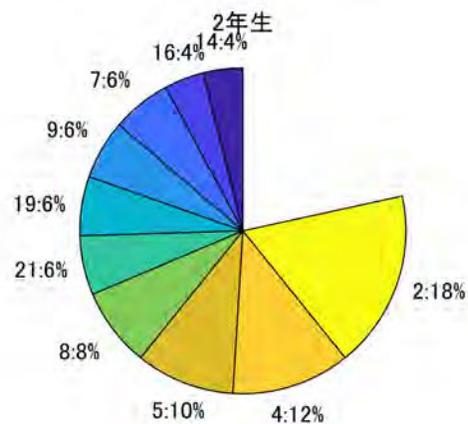
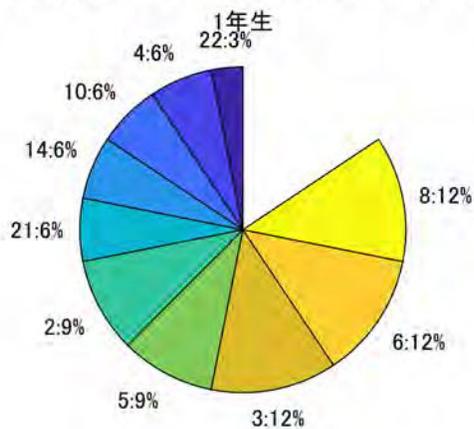


図3 中学生の各絵のクラスター数上位10個の割合を調べた円グラフ

%%%%%%%%

```

%%%%%%%%%%
% RGB 空間での小学生の絵の色の 3 次元プロット
figure('Position', [100 100 990 280]); % [left bottom width height]
f_3d_es = tiledlayout(1,3);

% 1 つ目
nexttile
scatter3(plot_data_es_and_to_k(:, 1), plot_data_es_and_to_k(:, 2),...
         plot_data_es_and_to_k(:, 3), plot_size, plot_data_es_rgb, 'filled');

view(40, 25); % 表示する図の角度を調整
xlabel('色相', "Color", 'r');
ylabel('彩度', "Color", [0, 0.6, 0]);
zlabel('明度', "Color", [0, 103, 192] ./ 255);
xlim([0 1]); ylim([0 1]); zlim([0 1])
xticks([0 1]); yticks([0 1]); zticks([0 1])
ax = gca; ax.BoxStyle = 'full'; ax.FontSize = 10;
grid off; box on

% 2 つ目
nexttile
scatter3(plot_data_es_and_to_k(:, 1), plot_data_es_and_to_k(:, 3),...
         plot_data_es_and_to_k(:, 2), plot_size, plot_data_es_rgb, 'filled');

view(40, 25); % 表示する図の角度を調整
xlabel('色相', "Color", "r");
ylabel('明度', "Color", [0, 103, 192] ./ 255);
zlabel('彩度', "Color", [0, 0.6, 0]);
xlim([0 1]); ylim([0 1]); zlim([0 1])
xticks([0 1]); yticks([0 1]); zticks([0 1])
ax = gca; ax.BoxStyle = 'full'; ax.FontSize = 10;
grid off; box on

% 3 つ目
nexttile
scatter3(plot_data_es_and_to_k(:, 2), plot_data_es_and_to_k(:, 3),...
         plot_data_es_and_to_k(:, 1), plot_size, plot_data_es_rgb, 'filled');
view(40, 25); %表示する図の角度を調整

xlabel('彩度', "Color", [0, 0.6, 0]);
ylabel('明度', "Color", [0, 103, 192] ./ 255);
zlabel('色相', "Color", "r");
xlim([0 1]); ylim([0 1]); zlim([0 1])
xticks([0 1]); yticks([0 1]); zticks([0 1])
ax = gca; ax.BoxStyle = 'full'; ax.FontSize = 10;
grid off; box on
%
title(f_3d_es, '3D Scatter Plot of HSV Values (小学生の絵) ');

```

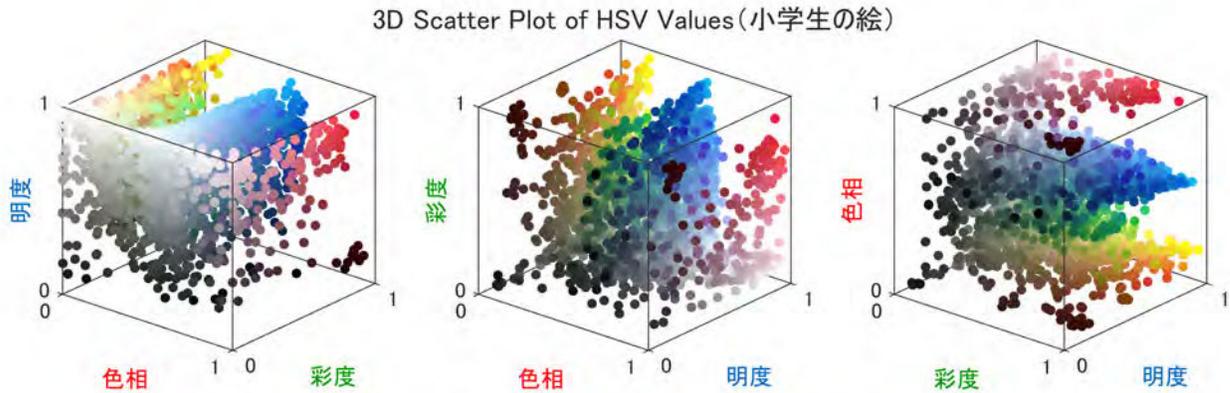


図4 小学生の絵に使われた色を色相、彩度、明度の三要素に分けた時の三次元散布図

```

%%%%%%%%%%
%%%%%%%%%%
% RGB空間での中学生の絵の色の3次元プロット
figure('Position', [100 100 990 280]); % [left bottom width height]
f_3d_js = tiledlayout(1,3);

% 1つ目
nexttile
scatter3(plot_data_js_and_to_k(:, 1), plot_data_js_and_to_k(:, 2),...
         plot_data_js_and_to_k(:, 3), plot_size, plot_data_js_rgb, 'filled');

view(40, 25); % 表示する図の角度を調整
xlabel('色相', "Color", 'r');
ylabel('彩度', "Color", [0, 0.6, 0]);
zlabel('明度', "Color", [0, 103, 192] ./ 255);
xlim([0 1]); ylim([0 1]); zlim([0 1])
xticks([0 1]); yticks([0 1]); zticks([0 1])
ax = gca; ax.BoxStyle = 'full'; ax.FontSize = 10;
grid off; box on

% 2つ目
nexttile
scatter3(plot_data_js_and_to_k(:, 1), plot_data_js_and_to_k(:, 3),...
         plot_data_js_and_to_k(:, 2), plot_size, plot_data_js_rgb, 'filled');

view(40, 25); % 表示する図の角度を調整
xlabel('色相', "Color", "r");
ylabel('明度', "Color", [0, 103, 192] ./ 255);
zlabel('彩度', "Color", [0, 0.6, 0]);
xlim([0 1]); ylim([0 1]); zlim([0 1])
xticks([0 1]); yticks([0 1]); zticks([0 1])
ax = gca; ax.BoxStyle = 'full'; ax.FontSize = 10;
grid off; box on

```

```

% 3つ目
nexttile
scatter3(plot_data_js_and_to_k(:, 2), plot_data_js_and_to_k(:, 3),...
    plot_data_js_and_to_k(:, 1), plot_size, plot_data_js_rgb, 'filled');
view(40, 25); %表示する図の角度を調整

xlabel('彩度', "Color", [0, 0.6, 0]);
ylabel('明度', "Color", [0, 103, 192] ./ 255);
zlabel('色相', "Color", "r");
xlim([0 1]); ylim([0 1]); zlim([0 1])
xticks([0 1]); yticks([0 1]); zticks([0 1])
ax = gca; ax.BoxStyle = 'full'; ax.FontSize = 10;
grid off; box on
%
title(f_3d_js, '3D Scatter Plot of HSV Values (中学生の絵) ');

```

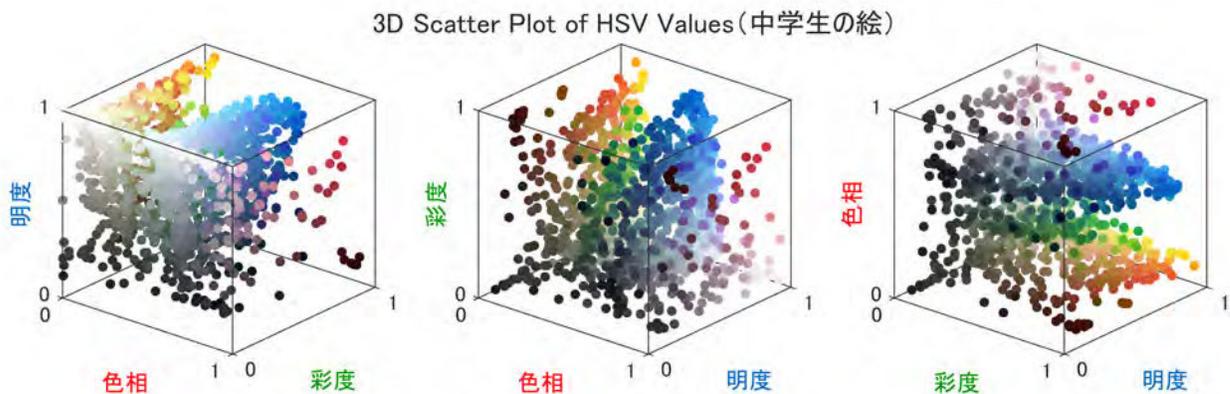


図5 中学生の絵に使われた色を色相、彩度、明度の三要素に分けた時の三次元散布図

```

%%%%%%%%%%
%%%%%%%%%%
% RGB空間での小学生の絵の色の2次元プロット
figure('Position', [100 100 890 270]); % [left bottom width height]
f_2d_es = tiledlayout(1,3);

% 1つ目 (色相と彩度)
nexttile
scatter(plot_data_es_and_to_k(:, 1), plot_data_es_and_to_k(:, 2),...
    plot_size, plot_data_es_rgb, 'filled');

title('色相と彩度の値')
xlabel('色相', 'Color', 'r'); ylabel('彩度', "Color", [0, 0.6, 0]);
xlim([0 1]); ylim([0 1]); zlim([0 1])
xticks([0 1]); yticks([0 1]); zticks([0 1])
ax = gca; ax.FontSize = 10; box on

% 2つ目 (色相と明度)
nexttile

```

```

scatter(plot_data_es_and_to_k(:, 1), plot_data_es_and_to_k(:, 3),...
        plot_size, plot_data_es_rgb, 'filled');

title('色相と明度の値');
xlabel('色相', "Color", 'r'); ylabel('明度', "Color", [0, 103, 192] ./ 255);
xlim([0 1]); ylim([0 1]); zlim([0 1])
xticks([0 1]); yticks([0 1]); zticks([0 1])
ax = gca; ax.FontSize = 10; box on

% 3つ目 (彩度と明度)
nexttile
scatter(plot_data_es_and_to_k(:, 2), plot_data_es_and_to_k(:, 3),...
        plot_size, plot_data_es_rgb, 'filled');

title('彩度と明度の値');
xlabel('彩度', "Color", [0, 0.6, 0]); ylabel('明度', "Color", [0, 103, 192] ./ 255);
xlim([0 1]); ylim([0 1]); zlim([0 1])
xticks([0 1]); yticks([0 1]); zticks([0 1])
ax = gca; ax.FontSize = 10; box on

title(f_2d_es, '2D Scatter Plot (小学生の絵) ');

```

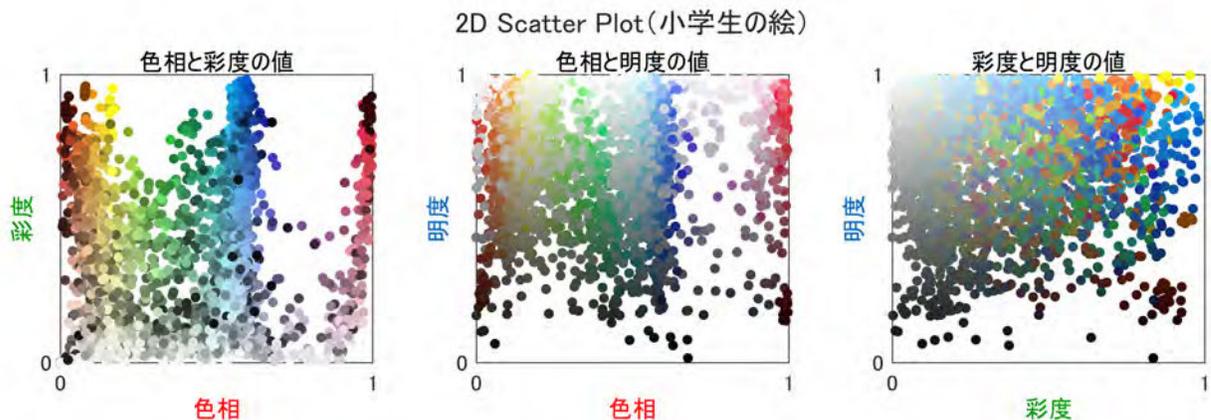


図6 小学生の絵に使われた色を色相、彩度、明度の三要素に分けた時の二次元散布図

```

%%%%%%%%%%
%%%%%%%%%%
% RGB空間での中学生の絵の色の2次元プロット
figure('Position', [100 100 890 270]); % [left bottom width height]
f_2d_js = tiledlayout(1,3);

% 1つ目 (色相と彩度)
nexttile
scatter(plot_data_js_and_to_k(:, 1), plot_data_js_and_to_k(:, 2),...
        plot_size, plot_data_js_rgb, 'filled');

title('色相と彩度の値')

```

```

xlabel('色相','Color','r'); ylabel('彩度',"Color",[0, 0.6, 0]);
xlim([0 1]); ylim([0 1]); zlim([0 1])
xticks([0 1]); yticks([0 1]); zticks([0 1])
ax = gca; ax.FontSize = 10; box on

% 2つ目 (色相と明度)
nexttile
scatter(plot_data_js_and_to_k(:, 1), plot_data_js_and_to_k(:, 3),...
        plot_size, plot_data_js_rgb, 'filled');

title('色相と明度の値');
xlabel('色相',"Color",'r'); ylabel('明度',"Color",[0, 103, 192] ./ 255);
xlim([0 1]); ylim([0 1]); zlim([0 1])
xticks([0 1]); yticks([0 1]); zticks([0 1])
ax = gca; ax.FontSize = 10; box on

% 3つ目 (彩度と明度)
nexttile
scatter(plot_data_js_and_to_k(:, 2), plot_data_js_and_to_k(:, 3),...
        plot_size, plot_data_js_rgb, 'filled');

title('彩度と明度の値');
xlabel('彩度',"Color",[0, 0.6, 0]); ylabel('明度',"Color",[0, 103, 192] ./ 255);
xlim([0 1]); ylim([0 1]); zlim([0 1])
xticks([0 1]); yticks([0 1]); zticks([0 1])
ax = gca; ax.FontSize = 10; box on

title(f_2d_js,'2D Scatter Plot (中学生の絵) ');

```

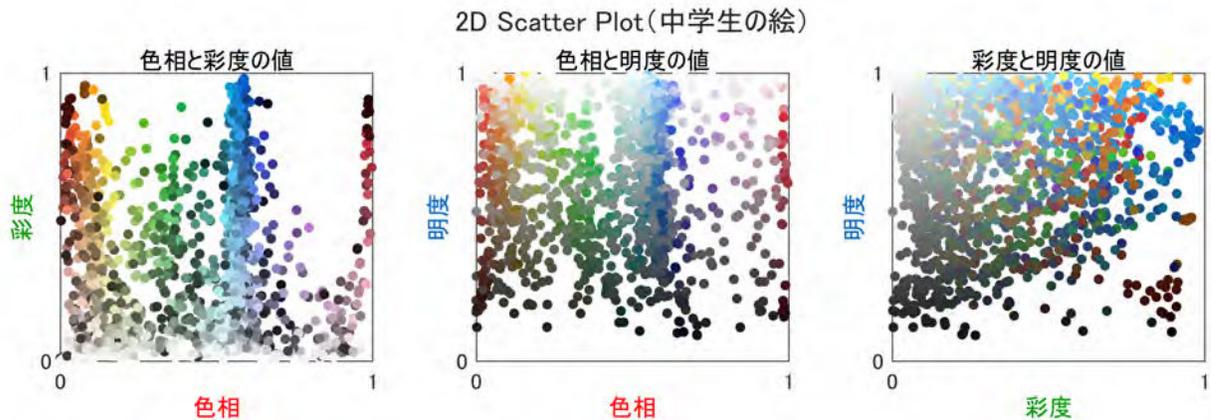


図7 中学生の絵に使われた色を色相、彩度、明度の三要素に分けた時の二次元散布図

```

%%%%%%%%%%
%%%%%%%%%%
% クラスタリングした後の全体のクラスタリング
k_max_es_js_all = 25; % change % 各絵の分類に用いる最大クラスター数の設定。

```

```

% 小学生の各絵の色をクラスタリングした後の全体のクラスタリング
rng('default'); % 乱数発生器をリセット, 同じシードを使用

% 最適なクラスタ数を決める。
%plot_data_es_and_to_k = rgb2hsv(plot_data_es_and_to_k);
eva_es_all = evalclusters(plot_data_es_and_to_k, myfunc,...
    'CalinskiHarabasz', 'KList', 1:k_max_es_js_all);
NumberOfClusters_es_cal = eva_es_all.OptimalK; % 最適なクラスタ数
NumberOfClusters_es = 15; % クラスタ数 : 円グラフに表示させたい色の数

% 円グラフに表示させたい色の数による k-means クラスタリング
[cluster_idx_es, cluster_center_es] = kmeans(plot_data_es_and_to_k,
NumberOfClusters_es,...
    'EmptyAction', 'singleton', 'MaxIter', 1000, 'Replicates', 5);

% 各クラスタの割合を計算
pixel_counts_es = histcounts(cluster_idx_es, 1:(NumberOfClusters_es+1)); % 各クラスタ
一に対応するピクセル数
total_pixels_es = length(cluster_idx_es); % クラスタすべてを合計したピクセル数
cluster_ratios_es = pixel_counts_es / total_pixels_es; % クラスタの割合

% 割合に基づいてクラスタを並べ替え
[sorted_ratios_es, sort_order_es] = sort(cluster_ratios_es);
sorted_colors_es = cluster_center_es(sort_order_es, :); % 割合を小さい順に並べた時の
HSV

exlude_es = zeros(NumberOfClusters_es,1); % 中心がくっついていない円グラフを作成する時に
使用
exlude_es = exlude_es + 1; % 中心がくっついていない円グラフを作成する時に使用
%%%%%%%%%%%%
%%%%%%%%%%%%
% 中学生の各絵の色をクラスタリングした後の全体のクラスタリング
rng('default'); % 乱数発生器をリセット, 同じシードを使用

% 最適なクラスタ数を決める。
%plot_data_js_and_to_k = rgb2hsv(plot_data_js_and_to_k);
eva_js_all = evalclusters(plot_data_js_and_to_k, myfunc,...
    'CalinskiHarabasz', 'KList', 1:k_max_es_js_all);
NumberOfClusters_js_cal = eva_js_all.OptimalK; % 最適なクラスタ数
NumberOfClusters_js = 15; % クラスタ数 : 円グラフに表示させたい色の数

% 最適なクラスタ数による k-means クラスタリング
[cluster_idx_js, cluster_center_js] = kmeans(plot_data_js_and_to_k,
NumberOfClusters_js,...
    'EmptyAction', 'singleton', 'MaxIter', 1000, 'Replicates', 5);

% 各クラスタの割合を計算
pixel_counts_js = histcounts(cluster_idx_js, 1:(NumberOfClusters_js+1)); % 各クラスタ
一に対応するピクセル数

```

```
total_pixels_js = length(cluster_idx_js); % クラスターすべてを合計したピクセル数
cluster_ratios_js = pixel_counts_js / total_pixels_js; % クラスターの割合
```

```
% 割合に基づいてクラスターを並べ替え
```

```
[sorted_ratios_js, sort_order_js] = sort(cluster_ratios_js);
sorted_colors_js = cluster_center_js(sort_order_js, :); % 割合を小さい順に並べた時の
HSV
```

```
explode_js = zeros(NumberOfClusters_js,1); % 中心がくっついていない円グラフを作成する時に
使用
```

```
explode_js = explode_js + 1; % 中心がくっついていない円グラフを作成する時に使用
```

```
%%%%%%%%%%
```

```
% 小学生の絵の色の割合の円グラフの作成
```

```
figure;
```

```
pie(sorted_ratios_es) %,explode_es)
```

```
colormap(hsv2rgb(sorted_colors_es)); % 並べ替えた色を使用
```

```
title("小学生の絵に使われた色をグループ分けした時の色の割合")
```

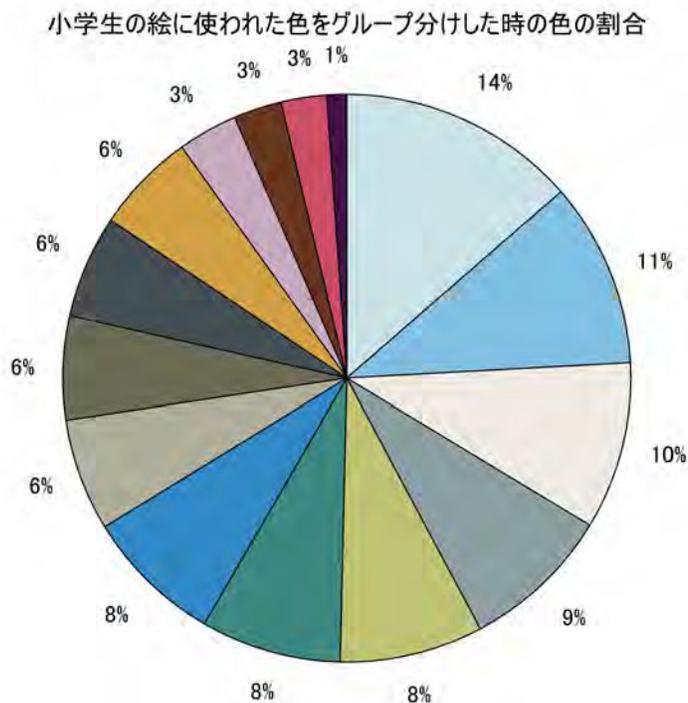


図 8 小学生の各絵で使われた色をクラスタリングし、その結果をまとめてクラスタリングした時の小学生の絵の色の割合

```
% 中学生の絵の色の割合の円グラフの作成
```

```
figure;
```

```
pie(sorted_ratios_js) %,explode_es)
```

```
colormap(hsv2rgb(sorted_colors_js)); % 並べ替えた色を使用  
title("中学生の絵に使われた色をグループ分けした時の色の割合")
```

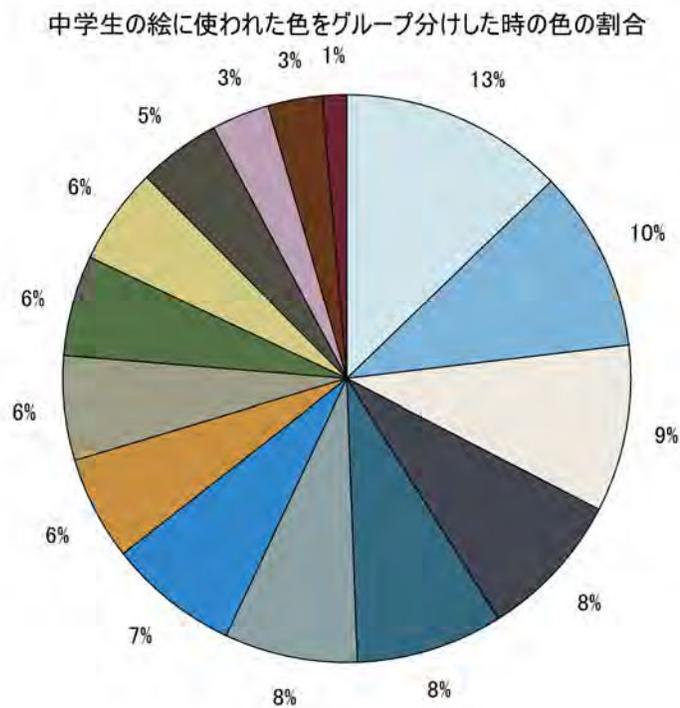


図9 中学生の各絵で使われた色をクラスタリングし、その結果をまとめてクラスタリングした時の中学生の絵の色の割合