

# 市立札幌旭丘高等学校数理データサイエンス科 2023年度(65期)SDS探求研究報告書

## 研究タイトル

パーソナルカラー診断アプリケーションの開発

## 班番号

11

## 班名

JOROCHANS

## 班員名

小橋恵 / Kobashi Megumi

高橋こころ / Takahashi Kokoro

富樫遥 / Togashi Haruka

関根美月 / Sekine Mitsuki

葛巻歩花 / Kuzumaki Ayuka

## 指導教員

新村貴之

## アドバイザー

公立はこだて未来大学 三上貞芳教授

公立はこだて未来大学 奥野拓教授

北海道大学情報科学院 西浦翼さん(TA)

北海道大学情報科学院 森雄斗さん(TA)

## 概要

本研究ではOpenCVを用いて顔の画像からパーソナルカラーを診断するアプリケーションを開発した。先行研究を参考に、顔写真からその人にあったパーソナルカラーを診断できるプログラミングをGoogleColabを用いてPython用語で作成した。複数の学生を対象に、実際に使用し、被験者が実際の専門機関で受けた診断結果とパーソナルカラー診断用コードを用いて得られた結果が一致しているか検証した。その結果、プログラム上で8割以上の正確率でパーソナルカラーを診断することが出来た。

文責:高橋ころ

## Abstract

In this study, we developed an application to diagnose personal color from facial images using OpenCV. Referring to previous research, programming that can diagnose a person's personal color from a facial image was created in Python using GoogleColab. We used the program on several students and verified whether the results obtained using the code for personal color diagnosis matched the results of the diagnosis that the subjects received at an actual professional institution. As a result, the program was able to diagnose personal color with an accuracy rate of more than 80%.

文責:高橋ころ

# 目次

## 第1章 はじめに

1.1 研究背景.....	4
1.2 先行研究.....	4

## 第2章 研究方法

2.1 パーソナルカラーについて.....	5
2.2 従来の診断方法.....	6

## 第3章 研究内容

3.1 アプリケーション作動の流れ.....	7
3.2 OPENCVを用いたパーツ検査.....	7

## 第4章 結果と考察

4.1 結果.....	9
4.2 考察.....	9
4.3 展望.....	9

## 第5章 発表報告

5.1 校内発表会.....	10
5.1.1 SDS成果発表.....	10
5.1.2 ST・SDS発表会.....	10

## 第6章 終わりに

6.1 感想.....	11
6.1.1 高橋ころろ.....	11
6.1.2 富樫遥.....	11
6.1.3 小橋恵.....	11
6.1.4 関根美月.....	12
6.1.4 葛巻歩花.....	12
6.2 謝辞.....	13
6.3 参考.....	13

付録: 今回用いたプログラムコード

## 第1章 はじめに

### 1.1 研究背景

コロナ禍があげ、徐々にマスクを外す機会が増えたことで世間の化粧への関心が高まりつつある。特に、中高生や若い世代の間で「パーソナルカラー診断」という言葉をよく耳にするようになった。パーソナルカラー診断とは、自分自身に似合う色「パーソナルカラー」を知る事で第三者に自分をより魅力的に見せ、自身の輝きをワンステップ上へと導いてくれる美容診断のことである。この診断は、ネット上で自分自身で行う場合や専門店で診断する場合がある。しかし、専門家による診断はかなり高額で、無料でできる診断はアンケート形式のため、同じ人でも違った検証結果が生じてしまう。よって無料で一人一人にあった診断が出来るアプリにニーズを感じ開発を始めた。

文責:小橋恵

### 1.2 先行研究

大阪産業大学 亀尾菜月さん

「パーソナルカラー診断アプリケーションの開発」

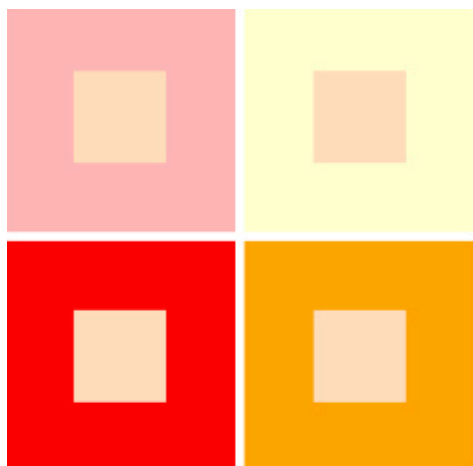
自身のソースを用いたパーソナルカラーのWEBアプリケーション開発

文責:小橋恵

## 第2章 研究方法

### 2.1 パーソナルカラーについて

私たちは始めにパーソナルカラーについて調べた。パーソナルカラーとは、第三者から見て似合う色のことであり、小顔、美白、目を美しく強く見せる、着痩せ、高感度アップなどの効果がある。自分の顔をイエベ春、ブルベ夏、イエベ秋、ブルベ冬の4つに分類し、それぞれの特徴に合わせてメイクや服の色を変えこれらの効果を得ることができることがわかった。イエベ春とは、茶色がかかった地毛や、クリームベージュ肌、明るく透け感のある瞳、血色感のある頬と唇などを持ち、元気で可愛い印象。ブルベ夏とは、ソフトな暗い地毛や、ピンクベージュの肌、黒～こげ茶の瞳、赤みピンクの頬などを持ち、エレガントでおとなしめな印象。イエベ秋とは、艶のある地毛や黄味がかかったベージュ、ダークブラウンの瞳、コーラル系の頬と唇などを持ち、大人っぽくゴージャスな印象。ブルベ冬とは、真っ黒な地毛と目、透けるような白い肌、やや青みがかかったピンクの肌、赤みの強いピンクの唇などを持ち、大胆で派手な印象。



周りの色によって色の見え方が異なる様子を示した図

文責:小橋恵 富樫遥

## 2.2 従来の診断方法

パーソナルカラー診断をする方法は主に2つある。

1つは専門の資格を持ったカラーリストが行う診断である。この診断にはドレープという布を使う。ドレープは大きく4つのグループに別れた色が使用され、例えば4種類の赤色の布でどの赤が一番似合うか、顔色の変化をみていく。客観的な判断を得られるため、パーソナルカラーの中でもその人の雰囲気に合わせて色やどう見せたいかという相談をすることができる。その代わりに、診断料金がかかること、診断場所に行かなければいけないこと、カラーリストとの相性があるという問題がある。

もう1つはWebサイトでいくつかの質問に答える診断である。髪、目、頬、肌、唇の色を答えるだけのものから数十問の質問が設置されているサイトがある。実際にWebサイトで診断をしようとする、質問数にばらつきがあり、どのWebサイトで診断をすればいいのか迷う。質問は、肌は青みがかっているか黄色がかっているかを自分で判断しなくてはならず、初心者には答えることが難しいものや、各パーツに近い色を選ぶ形式だと、近しい色が無い、わからないという問題がある。

文責：関根美月



写真1：公立はこだて未来大学の三上教授とのオンラインミーティングの様子

## 第3章 研究内容

### 3.1 アプリケーション作動の流れ

- ①顔写真を撮ると人間と認識される
- ②目・鼻・口をOPENCVによって切り取る
- ③それぞれのパーツの色から明度・彩度・輝度を抽出する
- ④明度・彩度・輝度の一定の値と検出したデータを比較する
- ⑤4シーズンに分けて表示する

	彩度	明度
イエベ春	50-100	50-100
ブルベ夏	50-100	0-49
イエベ秋	0-49	0-49
ブルベ冬	0-49	50-100

文責: 関根美月・高橋こころ

### 基になる明度と彩度を200にした理由

すでにパーソナルカラーが判明している芸能人の顔画像を用いて、診断結果が一致するように値を調整した。この時、フォーシーズンタイプそれぞれの画像が一致するような値にしたため、正確性は十分と言えるだろう。

文責: 関根美月

### 3.2 OPENCVを用いたパーツ検出

OPENCVとは画像解析・画像処理・機械学習などの機能を持つライブラリのこと、よく画像からの物体の検出に用いられる。今回は顔・鼻・目・口の四つの情報を持つカスケード分類機を用いて検出した

OPENCVを用いたパーツ検出の流れは以下の通りである。

- ①顔の画像にx軸、y軸を設置する
- ②OPENCVのカスケード分類器を用いて顔のパーツ(目・鼻・口)を検出
- ③検出された顔のパーツが画像上のどこにあるのかx座標とy座標を用いて表す
- ④x座標とy座標を基に画像から顔のパーツを切り取る

文責:関根美月



写真2:研究の様子



## 第4章 結果と考察

### 4.1 結果

OPEN CVを用いて顔の画像からパーツを切り取り、各パーツの明度・彩度・輝度を抽出し、抽出した値と一定の値を比較してフォーシーズンタイプに分けるコードを作成する事ができた。だが、当初の目的であるアプリ制作を実現することはできず、自動で診断されるアプリにはならなかった。

### 4.2 考察

ウェブアプリ化できなかった原因は次のように考えられる

- ・見本とした亀尾さんの論文に誤りがあった。
- ・学校のPCに搭載されているブロックソフトの影響でプログラムを実行したときに表示されるリンクにアクセスできなかった。
- ・私たちがプログラムを書き換えた時に誤字が生じてしまった
- ・Google Colabularyの環境

### 4.3 展望

- ・パーソナルカラー診断のプログラムを打つだけでなく、アプリ化に力を入れ、多くの人に使ってもらえるようにしたい。
- ・診断結果は信頼性があるとはいえないため、さらなる信頼性の確証と正確性を上げることが求められる。今後、個人のファッションの参考にできるだけでなく、アパレル業界とヘアメイク業界に貢献することが期待できる。
- ・音声をつけるなどして、障害のある方も同じようにパーソナルカラー診断ができるアプリケーションにしたい。
- ・パーソナルカラーの出る頻度の偏りを無くしたい。

文責: 関根美月

## 第5章 発表報告

### 5.1 校内発表会

#### 5.1.1 SDS探究成果発表会(12/6)

生徒からの質問が少なかったことから内容が難しく伝えきれてないという課題が残った。公立はこだて未来大学の奥野教授から、男女兼用で使えるのか、診断に時間が掛かる理由を質問された。また、プログラムの説明はフローチャートで説明するほうがわかりやすいと助言を貰った。

#### 5.1.2 ST・SDS発表会(2/6)

パーソナルカラー診断を実際にやってみたいという声が多く聞こえた。手軽に無料で使えるパーソナルカラーの需要が高いことを実感した。また、前回の助言を受けてプログラムをフローチャートで説明したり、実際に実行して説明すると分かりやすいと好評だった。しかし、質問はなかった。

文責:富樫遥

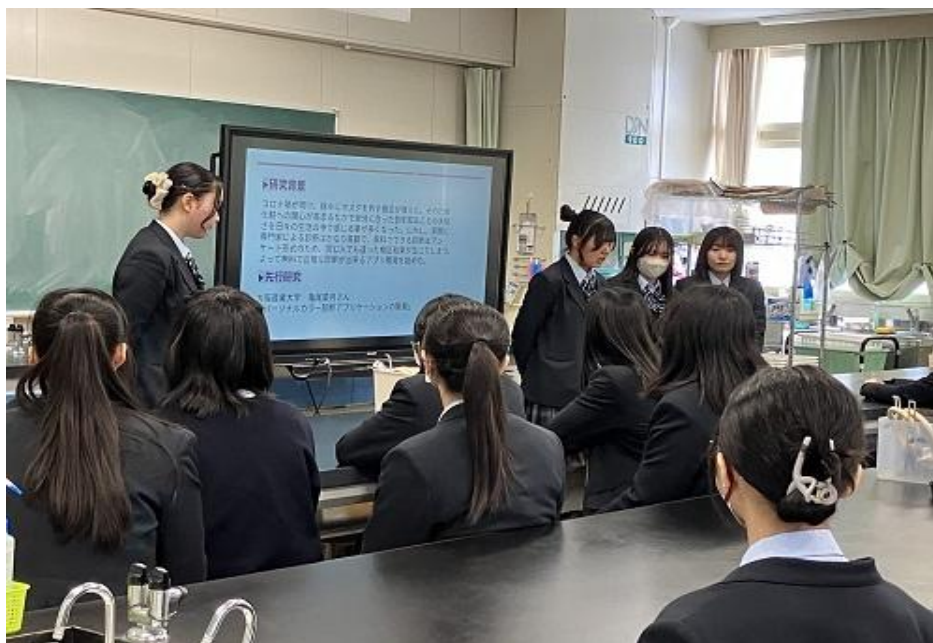


写真3:ST・SDS発表会の様子

## 第6章 終わりに

### 6.1 感想

#### 6.1.1 高橋こころ

アプリケーションを開発する上でさまざまなプログラム用語などを知ることによりプログラムに興味を持つことができた。さらにこのアプリケーションのように、専門家による診断が必要な事柄を技術を用いて解決するのに対応していくことが非常に重要になっていくと実感することができた。この活動を通して、私は今後このような機械と向き合う必要のある世代として、このような変化に対応していくことが大事だと思った。さらに今回の研究を活用して医療や教育などのさまざまな分野に広げ、より良い世の中になっていくよう私たちが発展させる必要があると思った。

#### 6.1.2 富樫遥

パーソナルカラー診断アプリ開発の研究の中で、協調性や創造性を身につけることが出来た。初めは、先行研究がたくさんある中で、オリジナリティを出すことが難しかったが様々な立場の人の視点にたって考えたり、班員と話し合う中で、自分たちが作りたいパーソナルカラー診断アプリの在り方を見つけられた。それを形にすることが大変だったが、いろいろな人から助言をもらい、研究を進めることが出来た。その中で試行錯誤しながら目的のために研究することの面白さに気づくことが出来た。余裕をもって行動出来なかった事が反省点だが、班員と助け合い、意見を出し合う中で自分自身の成長を感じられた。このパーソナルカラー診断アプリ開発の中で培った技術や経験を今後の生活の中で生かしていきたい。

#### 6.1.3 小橋恵

パーソナルカラーは近年若者の間でとても注目されている題材だったので、研究できてよかったと思う。アプリ開発となるとプログラミングの知識も浅く、かなり難しかったが、プログラミングについて学びながら研究を進めることができた。コードを作成し、

パーソナルカラーを診断することはできたが、アプリの完成には届かなかったのもとても残念だ。障害のある方でも診断できるアプリにするという目標も達成できなかったが、完成させたらとても役立つものになると思うので、研究、作成を続けていきたいと思った。校内の発表では、とても多くの生徒から興味関心を持っていただくことが出来たので、改めてパーソナルカラーの研究の意義を感じられた。

#### 6.1.4 関根美月

今回行ったパーソナルカラー診断アプリケーションの開発において、私は非常に自分自身の成長を感じた。元々は遺伝子についての研究を進めていた私たちであるが、他に気になっていたアプリ開発に切り替え、早急に取り組むことができたのが良かったと思う。アプリ開発未経験のメンバーであるだけでなく、プログラミングにおいても基礎しか知らなかったのにここまで完成度の高いアプリを開発できたことは誇りであり、非常に嬉しいことであった。今回の開発にあたり私が心掛けていたことは、この旭丘高校DS科で学んだことを今後の生活に当たる上の基礎と出来る様に習得することである。今回学んだ発表の仕方やまとめ方、論文の作成を今後の研究にも生かしていきたい。

#### 6.1.5 葛巻歩花

パーソナルカラーアプリケーションのプログラムを組むのは初めてで、プログラムの専門用語の知識は、ほぼ無知だったが北海道大学のTAの皆さんの多大なるサポートでなんとか完成することができた。学んだことも多く、プログラミングでエラーが出た際にどのように対応するべきなのか、プログラムの専門用語の意味も知ることができた。また、学んだことを次回以降しっかり生かしてしていきたい。

## 6.2 謝辞

本論文の作成にあたり、多くの方々にご指導ご鞭撻を賜りました。指導教官のはこだて未来大学副学長奥野拓教授には終始適切なご指導を賜りました。ここに深謝の意を表します。北海道大学、西浦翼氏と森雄斗氏には、本論文の作成にあたり、適切なお助言を賜りました。感謝申し上げます。最後に、大阪産業大学の亀尾菜月氏には、多くのデータを提供していただきました。厚く御礼申し上げます。本研究の遂行にあたり、快く実験に参加頂いた札市立札幌旭丘高校の皆様にも、感謝いたします。ありがとうございました。

文責：葛巻歩花

## 6.3 参考文献

亀尾菜月さん大阪産業大学『パーソナルカラー診断アプリケーションの開発』2016年度

<https://www.kikagaku.co.jp/kikagaku-blog/google-colab-drive-mount/>

## 付録: 今回用いたプログラムコード

### 1: パーツ検出

```
from google.colab import drive
drive.mount('/content/drive')

def cut_img(img):
    dir = 'drive/MyDrive/'

    face_cascade = cv2.CascadeClassifier(dir + 'haarcascade_frontalface_default.xml')
    eye_cascade = cv2.CascadeClassifier(dir + 'haarcascade_mcs_ryeeye.xml')
    nose_cascade = cv2.CascadeClassifier(dir + 'haarcascade_mcs_nose.xml')
    mouth_cascade = cv2.CascadeClassifier(dir + 'haarcascade_mcs_mouth.xml')

    #current_dir = os.path.abspath(os.path.dirname("__file__"))
    #img.save( os.path.join(current_dir, img.filename))
    #filename = img.filename
    current_dir = 'drive/MyDrive/Colab_Notebooks'

    img = cv2.imread(img, cv2.IMREAD_COLOR)
    #os.remove(os.path.join(current_dir, filename))

    if(img is None):
        return False

    else:
        gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
        # print('gray', gray)
        face = face_cascade.detectMultiScale(gray)
        print('face', face)

        if type(face) is tuple:
            return False

    i = 0

    for (x,y,w,h) in face:
        roi_gray = gray[y:y+h, x:x+w]
        roi_face = img[y:y+h, x:x+w]

        eyes = eye_cascade.detectMultiScale(roi_gray, 1.3, 5)
```

```
nose = nose_cascade.detectMultiScale(roi_gray,1.3,5)
mouth = mouth_cascade.detectMultiScale(roi_gray,1.3,5)

if type(eyes) is tuple:
    roi_eyes = False
else:
    for(ex,ey,ew,eh) in eyes:
        i+=1
        cv2.imwrite(current_dir+'/static/image/'+str(i)+'.jpg',roi_face[ey:ey+eh,])

if type(nose) is tuple:
    roi_nose = False

else:
    for (nx,ny,nw,nh) in nose:
        i+=1
        cv2.imwrite(current_dir+'/static/image/'+str(i)+'.jpg', roi_face[ny:ny+nh,])

if type(mouth) is tuple:
    roi_mouth = False
else:
    for(mx,my,mw,mh) in mouth:
        i+=1
        cv2.imwrite(current_dir+'/static/image/'+str(i)+'.jpg', roi_face[my:my+mh,])
```

## 2:パーソナルカラー判別

```
import cv2
import numpy as np
import os
import glob

def personal_color(eye, nose, mouth):
    #current_dir = os.path.dirname(__file__)
    current_dir = 'drive/MyDrive/Colab_Notebooks'
    if eye == 0:
        roi_eyes = False
    else:
        roi_eyes = cv2.imread(current_dir+'/static/image/'+str(eye)+'.jpg',cv2.IMREAD_COLOR)

    if nose == 0:
        roi_nose = False
    else:
        roi_nose = cv2.imread(current_dir+'/static/image/'+str(nose)+'.jpg',
cv2.IMREAD_COLOR)
```

```
if mouth ==0:
    roi_mouth = False
else:
    roi_mouth =
cv2.imread(current_dir+'/static/image/'+str(mouth)+'.jpg',cv2.IMREAD_COLOR)

def season(roi_parts):
    hue = 0
    saturation = 0
    value = 0

    if roi_parts is False :
        no_parts = roi_parts

    hsvcat = cv2.cvtColor(roi_parts , cv2.COLOR_BGR2HSV)

    tyuusin = (hsvcat[hsvcat.shape[0]//2, hsvcat.shape[1]//2], (hsvcat[hsvcat.shape[0]//2+1,
hsvcat.shape[1]//2+1]), (hsvcat[hsvcat.shape[0]//2+1, hsvcat.shape[1]//2-1]),
(hsvcat[hsvcat.shape[0]//2-1, hsvcat.shape[1]//2+1]),(hsvcat[hsvcat.shape[0]//2-1,
hsvcat.shape[1]//2-1])

    for h, s, v in tyuusin:
        hue += h. astype(np.float64)
        saturation += s.astype(np.float64)
        value += v.astype(np.float64)

    ave_hue = hue/5
    ave_saturation = saturation/5
    ave_value = value/5

    spring=0
    summer=0
    autumn=0
    winter=0

    if ave_saturation > (200/2) and ave_value > (200/2):
        spring += 1
    elif ave_saturation < (200/2) and ave_value < (200/2):
        autumn += 1
    elif ave_saturation < (200/2) and ave_value > (200/2):
        summer += 1
    elif ave_saturation > (200/2) and ave_value < (200/2):
        winter += 1

    return spring , autumn , summer , winter
```



```
if type(roi_eyes) !=bool:
    spring , autumn , summer , winter =season(roi_eyes)
else:
    print('no')

if type(roi_nose) !=bool:
    spring , autumn , summer , winter =season(roi_nose)
else:
    print('no')

if type(roi_mouth) !=bool:
    spring , autumn , summer , winter = season(roi_mouth)
else:
    print('no')

list = [spring, summer, autumn, winter]
re_season = max(list)

if re_season == spring:
    return 1
elif re_season ==summer:
    return 2
elif re_season == autumn:
    return 3
elif re_season == winter:
    return 4

os.remove(current_dir+'static/image/*.jpg')
```

### 3:実際に判定するコード

```
from google.colab import drive

drive.mount('/content/drive')
cut_img('/content/drive/MyDrive/Colab_Notebooks/ .jpeg')

current_dir = 'drive/MyDrive/'
img_dir = current_dir + 'static/image/'
personal_color( , , ) #eye nose mouth
```

文責: 関根美月・高橋こころ