卒業論文

ディープラーニング顔画像解析基盤システム

の試作と評価実験

提出者　　田坂征也

提出年月日　平成 31 年 1 月 25 日

指導教員　　金子邦彦 教授

ディープラーニング顔画像解析基盤システム

の試作と評価実験

情報工学科　田坂征也

研究概要

近年の事故の中で多いのはハンドル操作を誤った，ブレーキとアクセルを踏み間違えたなどの運転操作ミスや，脇見運転や同乗者とのおしゃべり，居眠り，単にボーッとしていたなどの漫然運転が原因としてあがり，この 3 つは全てヒューマンエラーによって起こされるものである．このようなヒューマンエラーによる事故を防ぐべくさまざまな安全装置がある中，私は運転者本人を注目するシステムを目標とした．

目次

1. まえがき

近年の交通事故の中で多いのはハンドル操作を誤りや，ブレーキとアクセルを踏み間違えたなどの運転操作ミスや，脇見運転や同乗者とのおしゃべりや，居眠りや，単にボーッとしていたなどの漫然運転である．これらは全てヒューマンエラーである．このようなヒューマンエラーの事故を防ぐべく，車載センシング機器により，生体反応やヒヤリ・ハットなどを検出し，検出した情報は管理者へ通知されるといったシステムがすでに多数存在している．また，運転者本人を注目したシステムの他に，車載のさまざまな安全システムが搭載されている．最新の乗用自動車では，アンチロックブレーキ装置，横滑り防止装置，エアバッグ装置などの安全装備が標準化されている．また近年では，前方を走る自動車との追突事故を予想して防ぐシステムが多くの車種に設定されるようになっている．また，並列する車両があるときにアラームで知らせ，不意のレーンチェンジを抑止する装置もある．

本論文では，カメラ付きのコンピュータとディープラーニング基盤 Keras を使った，人工知能カメラ(AI カメラ)の試作と評価実験について報告する．本論文で示す人工知能カメラについて，種々の実験の取り組みを重ねた．将来は，運転者本人の顔色やその他顔の変化を読み取ることができるディープラーニング顔画像解析に役立つ可能性がある基盤システムである．事故を未然に防ぐという社会的意義がある．

2. 既存技術およびライブラリ

2.1 Chocolatey

Chocolatey とは，パッケージマネージャーというものの 1 つである．Chocolatey は，Windows オペレーティングシステムで動く．その機能には，パッケージの検索，インストール，アップデート，バージョンを指定してのインストールなどがある．

2.2 Anaconda

Anaconda は，Continuum Analytics 社が提供している Python バージョン 3 の言語処理系，開発環境やツール，管理ツールである conda，その他 Python の主要なパッケージを 1 つにまとめたソフトウェアである．次のアプリケーションも同封されている．

- jupyter:Web ベースで動くデータ解析環境．Python，Julia，Ruby，

R, Lua, LuaJIT, Haskel, Scala, Go, JavaScript, node.js, bash などに対応している.

- qtconsole:対話型の実行環境である. コンソール機能, グラフ描画機能がある.
- spyder:Python 用の学習向けに開発された統合開発環境である. エディタ, デバッガ, 対話型実行環境など, 便利なツールがセットになっている. Python プログラムを書くと, 構文を見やすく色をつけたり, プログラムを記述するときに便利な機能がついているだけでなく, プログラムを途中で止めて変数の一覧を確認できるブレークポイント機能などのプログラム開発に便利な機能がある.

2.3 ezgiakcora/Facial-Expression

ezgiakcoro/Facial-Expression-Keras は, GitHub にて公開されているプログラムである. これは顔検知・顔識別のライブラリである Dlib を使った表情認識プログラムで, ディープラーニングを使用して映像から表情を認識することができる. 各フレーム内の顔を検出してから, 表現を 7 つのクラスのうちの 1 つに分類をする. その 7 つは, Angry(怒り), Disgusted(嫌悪), Neutral(中間), Sad(悲しい), Happy(幸せ), Surprised(驚く), Fear(恐怖)である.

2.4 ipazc/MTCNN

ipazc/MTCNN は, GitHub にて公開されているプログラムである. Python3.4 以降で動く, TensorFlow を用いた顔検出ソフトウェアである.

2.5 mpatacchiola/DeepGaze

mpatacchiola/DeepGaze は, GitHub にて公開されているプログラムである. ヒューマンコンピュータインタラクション, 人物検出, および顔検出, 頭部姿勢推定および分類のためのライブラリであり, 畳み込みニューラルネットワーク(CNN)を用いている. 人の注視点に関する情報(注意を向けている方向でもある)は, 頭の向きを見つけることによって推定できる. 目の情報を直接利用しないことは, サングラスなど, 目が何かで覆われているとき, またはユーザーがカメラから離れすぎて目の領域を適切な解像度でつかむことができないときに役立つ. なお, 虹彩が画像として取得できるときは, 眼球の注視方向を推定することが可能である.

DeepGaze には以下の便利なパッケージが含まれている.

- 頭部姿勢推定(Perspective-n-Point)
- 顔検出(Haar Cascades)
- 肌と色の検出(範囲検出, 逆投影)
- ヒストグラムに基づく分類(ヒストグラム交差法)
- 動き検出(フレーム差分, MOG, MOG2)
- モーショントラッキング(Particle filter)
- 顕著性マップ(FASA)

3. 顔画像解析基盤システムのセットアップ

3.1 Chocolatey のインストール

Chocolatey の Web ページの記載の手順に従う.

① Web ブラウザで, 下記の URL の Chocolatey の Web ページを開く.
https://chocolatey.org/

② 「Install Chocolatey Now」をクリックする. 移動したページにて, 「Install with cmd.exe」のコマンドをコピー.

③ Windows のコマンドプロントを管理者として実行した後に, 先ほどコピーしたコマンドを実行.

Chocolatey を用いて, git, cmake, wget, 7zip をインストールするために, Windows のコマンドプロントを管理者として実行し, 下記のコマンドを実行.

choco install -y git cmake.install wget 7zip

3.2 Anaconda,Python2 のインストール

① Anaconda, Python2 をインストールしたいので Windows のコマンドプロントを管理者として実行し, 下記のコマンドを実行.

choco install -y anaconda3 python2

② Anaconda の conda-forge のチャンネルを削除するために下記のコマンドを実行. エラーメッセージが出ることもあるが, このときのエラーメッセージは無視して良い.

C:¥tools¥Anaconda3¥Scripts¥conda config –remove chanenls conda-forge

③ Anaconda の conda パッケージを更新したいので, 下記のコマンドを実行.

C:¥tools¥Anaconda3¥Scripts¥conda upgrade -all

y か n を尋ねる質問が来たときは, 続行したいので「y」.

④ Anaconda の古い conda パッケージファイルの削除するために下記のコマンドを実行.

C:¥tools¥Anaconda3¥Scripts¥conda clean --packages

⑤ Chocolatey でインストール済みのパッケージを一括更新.
下記のコマンドを Windows コマンドプロントにて実行.

choco upgrade -y all

⑥ 先ほどインストールしたソフトウェア類に関する設定にて, Windows のシステム環境変数 Path の先頭部分の設定した.

3.3 Anaconda における隔離された Python 環境の作成

Anaconda を用いて. 必要なソフトウェアをインストールする.

① 今から作成する Python 環境の名前と, Python のバージョンを決めておく.

- Python 環境の名前:ai
- Python のバージョン:3

② Windows のコマンドプロントを実行し, 下記のコマンドを実行.

conda reate -n ai python=3

これにより元からの Python 環境と, 新規作成された Python 環境(Python のバージョン 3, 名前は ai)の共存ができた.
新規作成された Python 環境を使いたいときには, Windows でコマンドプロントを実行し, 下記の通りに実行.

active ai

もとの Python 環境に戻るときは「**exit**」.

③ Keras, TensorFlow, OpenCV, spyder のインストールをするために下記のコマンドを実行.

conda install -y tensorflow keras spyder opencv

④ numpy, scipy, h5py, scikit-learn, scikit-image, matplottlib, seaborn, pandas, pillow, jupyter, pytest, docopt, pyyaml, cython のインストールするために下記のコマンドを実行.

conda install -y numpy scipy h5py scikit-learn scikit-image matplotlib seaborn pandas pillow
conda install -y jupyter pytest docopt pyyaml cython

⑤ imutils を一度削除するために下記のコマンドを実行.

mkdir c:¥pytools

cd c:¥pytools

rmdir ¥s ¥q imutils

⑥ imutils をインストールするために下記のコマンドを実行.

cd c:¥pytools

git clone https://GitHub.com/jrosebr1/imutils

cd imutils

python setup.py build

python setup.py install

3.3.1 Microsoft Build Tools for Visual Studio 2017 のインストール

ここでは, Chocolatey を用いてインストールすることにしている.

① Windows SDK 10.1 と Microsoft Build Tools for Visual Studio 2017 をインストールするために Windows のコマンドプロントを管理者として実行し, 下記のコマンドを実行.

choco install -y windows-skd-10.1

choco install -y visualstudio2017-installer

choco install -y Microsoft-build-tools

② C++について設定をしたいので, Visual Studio Installer を起動する.

③ Visual Studio Build Tools 2017 の画面で「変更」をクリック.

④ 「Visual C++ Build Tools」を選び, 右下の「変更」をクリック.

⑤ インストールが始まるので待つ. 終わると「インストール済み」と表示される.

3.3.2 Dlib のインストール方法

Dlib をインストールするために, Windos のコマンドプロントを管理者として実行し, Dlib Python パッケージをインストールするために下記のコマンドを実行.

cd c:¥pytools

git clone https://GitHub.com/davisking/dlib

cd dlib

python setup.py build

python setup.py install

3.4 ezgiakcora/Facial-Expression-Keras のインストール方法

① 節 2.3 にて説明した ezgiakcora/Facial-Expression-Keras をインストールするために，Windows コマンドプロントを管理者として実行し，下記のコマンドを実行．

cd c;¥pytools

git clone https://GitHub.com/ezgiakcora/Facial-Expression-Keras

cd Facial-Expression-Keras

② 節 3.2.2 にてインストールした Dlib 関連のファイルをコピー．

copy

c:¥pytools¥dlib¥python_examples¥shape_predictor_68_face_landmarks.dat

3.5 ipazc/MTCNN のインストール方法

節 2.4 にて説明した ipazc/MTCNN をインストールするために，Windows コマンドプロントを管理者として実行し，下記のコマンドを実行．

cd c:¥pytools

git clone https://GitHub.com/ipazc/mtcnn

cd mtcnn

python setup.py build

python setup.py install

3.6 mpatacchiola/DeepGaze のインストール方法

節 2.5 にて説明した mpatacchiola/Deepgaze をインストールするために，Windows コマンドプロントを管理者として実行．

cd c:¥pytools

git clone https://GitHub.com/mpatacchiola/deepgaze

cd deepgaze

python setup.py build

python setup.py install

4. 実験

人工知能カメラを作る上で確認したいことは，表情認識するときの性能と

精度評価、顔検知の精度、肌色部分の抽出の精度の確認である. 3 章で説明した顔画像解析基盤システムを用いて, 評価実験を実施した.

4.1 表情認識の実験概要

実験は, 隔離した Pyhon 環境を有効にし, ezgiakcora/Facial-Expression-Keras 中の「demo.py」を Windows のコマンドプロントにて実行し, USB カメラを通して, 表情の変化をつけたり, 複数人いた場合, どの表情を認識するかを評価する.

4.2 表情認識の実験結果

図 1 では, 顔の表情で感情ごとに Windows のコマンドプロント上にて%表記されているためわかりやすいが, 大袈裟な反応をしないと思った通りの感情という結果にならないことが確認できた. コマンドプロントにて顔の表情によって, 感情を数値化するのは複数顔があった場合, 先に顔として認識された方が優先されるということが確認できた.

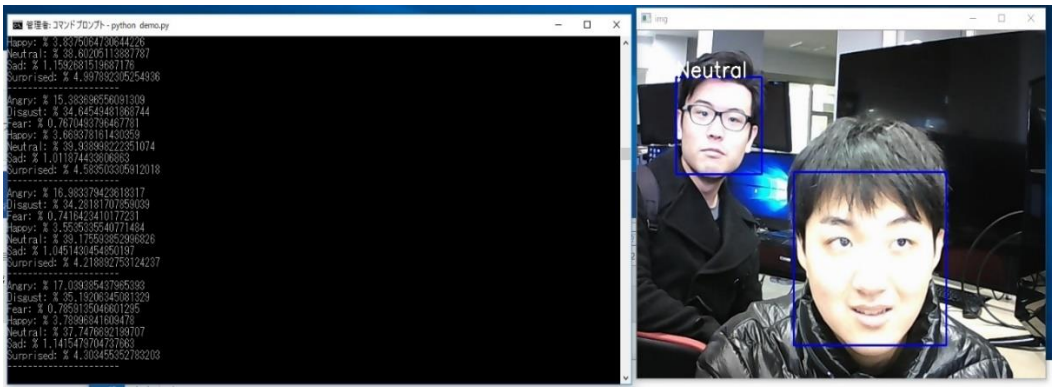

図 1 ezgiakcoro/Facial-Expression-Keras の表情認識の画面

4.3 顔検知の実験概要

実験は, 隔離した Python 環境を有効にし, その Python 環境にて spyder を起動し, USB カメラを通した映像や, 研究室の人が各自スマホで自分の顔を自撮りした動画を使い, ipazc/MTCNN を実行した.

4.4 顔検知の実験結果

図 2 では, ちゃんと顔検知されているが, 図 3 においては顔を検知していると共に, 後ろにあるマウスまで顔として検知してしまうことが確認できた.

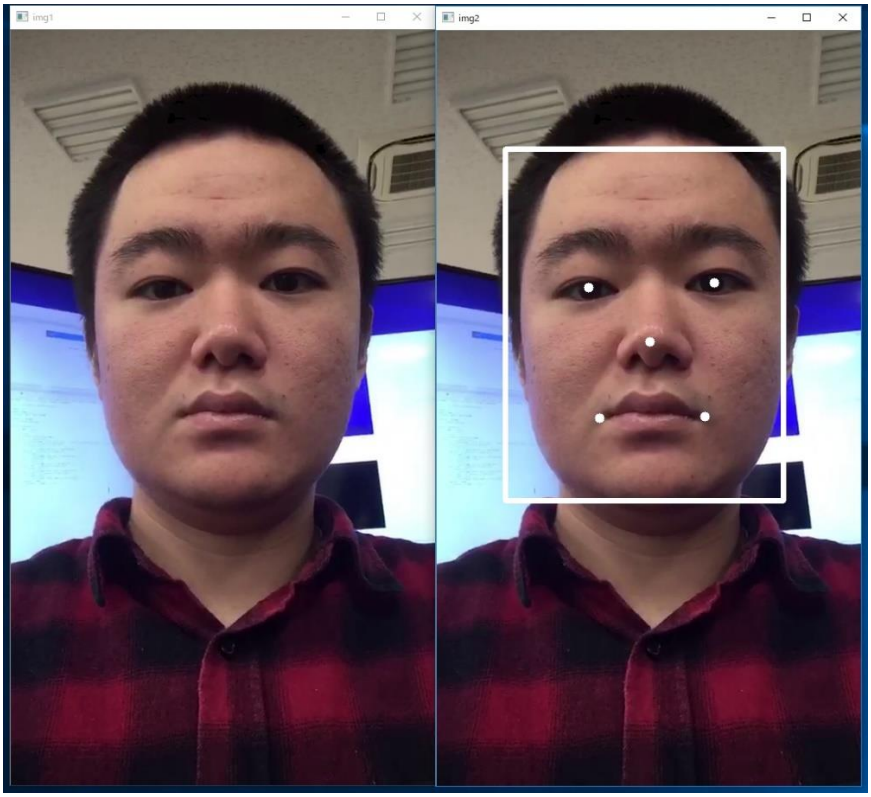

図 2 自撮りした動画を顔検知した画像 1

図 3 USB カメラを通した映像に顔検知をした画像

図 4 と図 5 を比較すると，顔の上に手を持ってくると顔検知が外れることが確認できた．

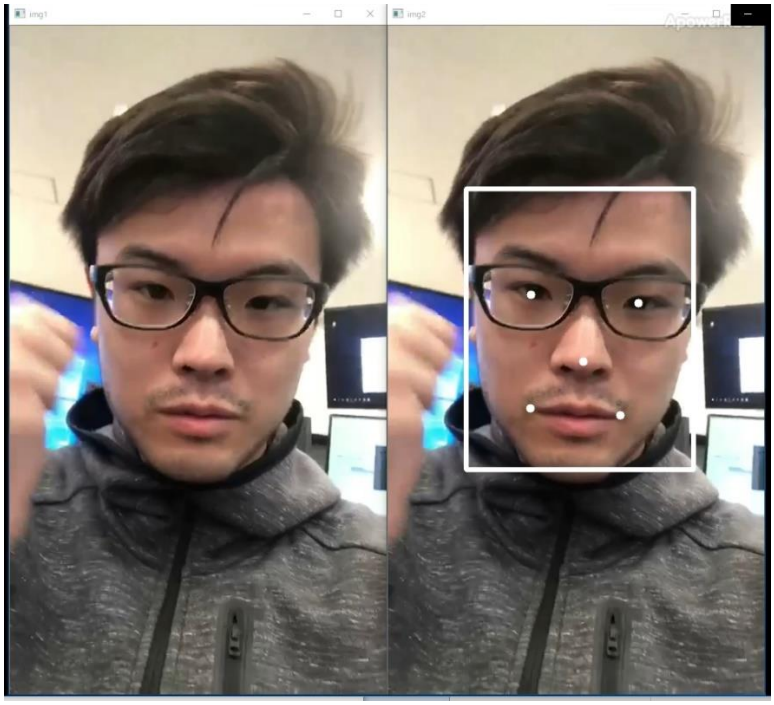

図 4 自撮りした動画を顔検知した画像 2

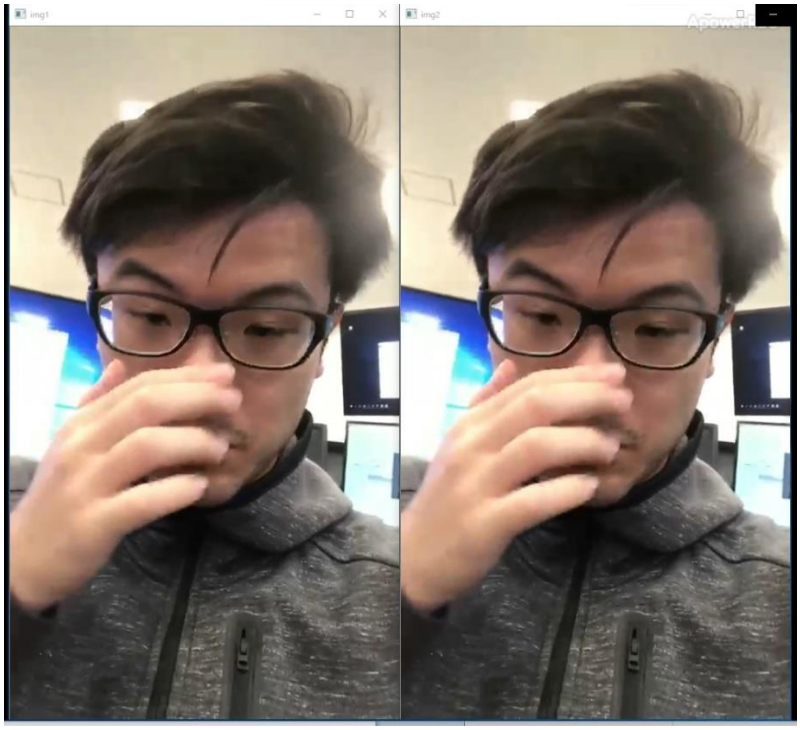

図 5 自撮りした動画を顔検知した画像 3

4.5 肌色抽出の実験概要

実験は，隔離した Python 環境を有効にし，その Python 環境にて Spyderを起動し，研究室の人が各自スマホで自分の顔を自撮りした動画を使い，mpatacchiola/DeepGaze を実行した．

4.6 肌色抽出の実験結果

図 6 では，肌色部分の切り出しを行っているが，一部光の反射によって切り出しから除外されていたり，コンピューターには肌色と思われる木等の部分が肌色として切り出されていることが確認できた．

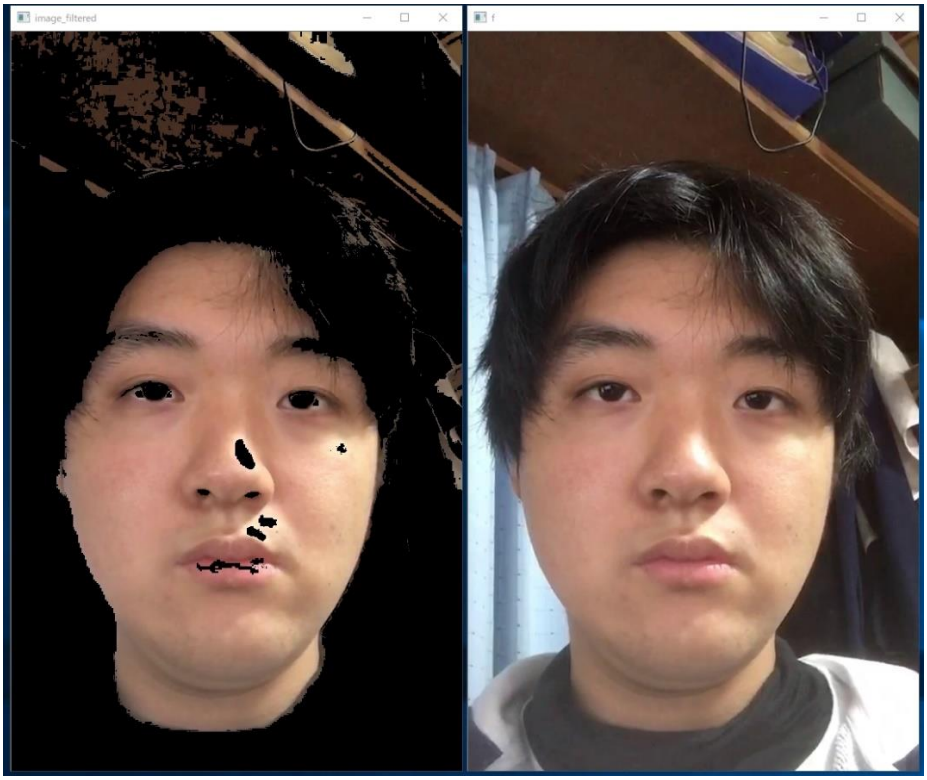

図 6 自撮りした動画を肌色部分の抽出をした画像 1

図 7 では，動画の画質が悪いため，肌色の切り出し方が粗く，逆光の影によって顔の大部分が肌色として抽出されないことが確認できた．

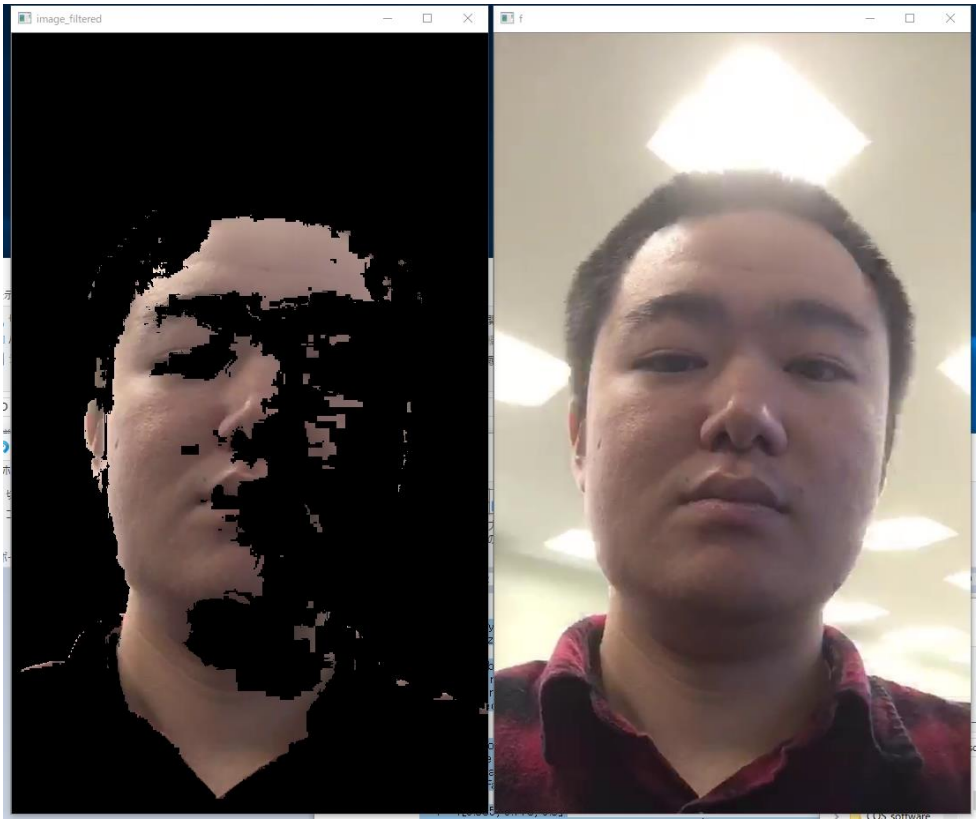

図 7 自撮りした動画を肌色部分の抽出した画像 2

5. 試してみたがうまくいかなかったもの

adithyaselve/face-expression-detect とは，ezgiakcora/Facial-Expression-Keras と似たような表情認識をできるものだが，私の環境にて正常に動作しなかった．原因追及のため，さまざまのことを試したが解決には至らなかった．原因を調べ切ることもできなかった．

6. むすび

本研究では既存の技術を重視し，コンピューターによる人間の顔の表情の読み取りや，顔検知，肌色部分の抽出を行った．人工知能カメラを作る取り組みの上で，新たな課題を発見することができた．また，人間の状態や異常をコンピューターに読み取らせる上で，医学のような自分の専攻外の分

野に触れるよいきっかけとなった．今回，人工知能カメラとして必要だと判断したソフトウエア類を Windows オペレーティングシステムで動作させる手順をまとめることができたこと，それらの検証もできたことも成果である．

謝辞

本研究の実施にあたり、卒業論文指導教員の情報工学科・金子邦彦教授にご指導を賜りました．金子邦彦研究室の飯塚氏，井上氏，半田氏には，実験の協力，研究室や実験の場での議論等を通して，知識や示唆の提供をいただきました．ここに感謝の意を表します．本研究は科研費（16K00163）の助成を受けたものである．

参考文献

(1) Chocolatey – The package manager for windows
https://chocolatey.org/
(2) Home – Anaconda
https://www.anaconda.com/
(3) GitHub - davisking/dlib: A toolkit for making real world machine learning and data analysis applications in C++itHub - davisking/dlib: A toolkit for making real world machine learning and data analysis applications in C++
https://GitHub.com/davisking/dlib
(4) GitHub - ezgiakcora/Facial-Expression-Keras: The aim of this project is to recognize facial expression from a video streaming by using deep learning.
https://GitHub.com/ezgiakcora/Facial-Expression-Keras
(5) GitHub - ipazc/mtcnn: MTCNN face detection implementation for TensorFlow, as a PIP package.
https://GitHub.com/ipazc/mtcnn
(6) GitHub - mpatacchiola/deepgaze: Computer Vision library for human-computer interaction. It implements Head Pose and Gaze Direction Estimation Using Convolutional Neural Networks, Skin Detection through Backprojection, Motion Detection and Tracking, Saliency Map.
https://GitHub.com/mpatacchiola/DeepGaze