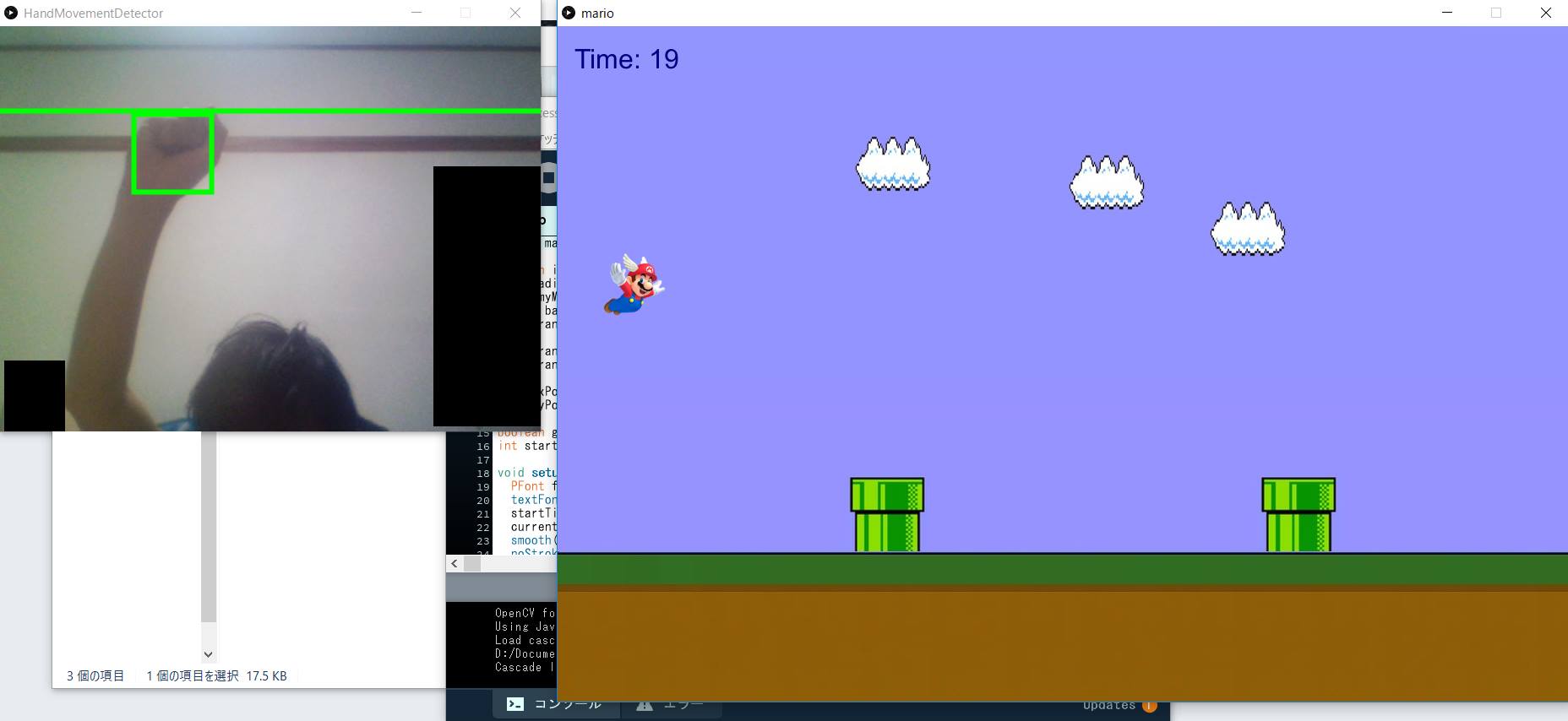
2017

コー　ユーリャン

t14520yk

2017年7月25日

画像処理プログラミング



**はじめに**

OpenCV機械学習ライブラリーを使って、拳認識でキャラクターを動かすゲームを製作しました。認識方法はHaar-Cascadeで、データは参考文章に張ったオープンソースコードを使って、拳認識が出来ました。

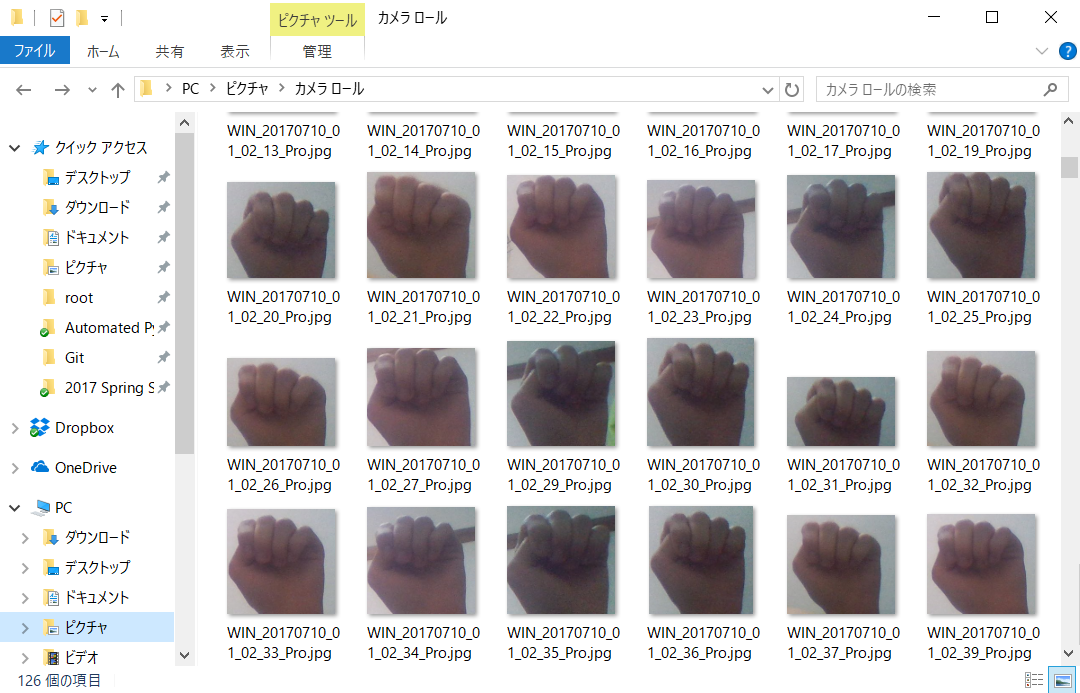
**発表の概要**

筋トレが楽しめるゲームを作ろうとしました。簡単に手を画面に表示されてるの緑の線の上に動かせば、キャラクターが上に向かいます。

手を上がる速度による、キャラクターの速さが変わります。

雲かパイプ（ホール）をぶつかったら、ゲームが終了します。スコアは左上のタイマーとなります。

**データの取り方**

カスケードが必要だったので、色々調べました。最初はポジティブとネガティブイメージサンプルが必要でした。ポジティブイメージは自分で125枚を撮りました。以下にご覧ください。

イメージ処理はpythonのPILでこの風に処理しました：

*# ポジティブイメージ処理*

for i in range(1, len(sys.argv)):

try:

filepath = sys.argv[i]

image = Image.open(filepath)

except IOError, e:

print "Problem opening", filepath, ":", e

continue

image = image.crop( (100,100,500,600) )

image = image.resize((1080, 566), Image.ANTIALIAS)

image.save(name + '\_new' + extension)

そして、ネガティブイメージはHIPIを使いインスターグラムからAWSのEC2でダウンロードして、txtファイルに名前変わって、ポジティブイメージをベックトールで組み合わせて12万枚のポジティブ画像サンプルを作りました。

perl bin/createsamples.pl positives.txt negatives.txt samples 1000

“opencv\_createsamples –bgcolor 0 – bgthresh 0 -maxxangle 1.1\

-maxyangle 1.1 maxzangle 1.1 – maxidev 40 -w 1080 -h 566”

次いで、カスケードをトレーニングし、自分の拳を認識できるようになりました。

opencv\_traincascade -data classifier -vec samples.vec -bg negatives.txt\

-numStages 20 -minHitRate 0.999 -maxFalseAlarmRate 0.5 -numPos 125000\

-numNeg 1000 -w 1080 -h 566 -mode ALL -precalcValBufSize 1024\

-precalcIdxBuffSize 1024

でも、その後にグーグルで私の拳だけではなく、皆の拳を認識できたオープンソースコードを発見して、今回の課題で使われてるカスケードはそれを使うことにしました。

**アピールポイント**

アピールポイントは、ダムベールだけではなく、本や水稲とか、手を拳形できればゲームを楽しめます。

データセットがあれば、コードを一行変わるだけで、拳だけではなく、掌でもコントロールすることが出来ます。

**今後やりたいこと**

もし時間の余裕があったら、checkGameOverの条件をもっと最適化にして、綺麗に書きたいと思います。今のところは、一つ一つの条件でチェックするしか思いつきません。

それと、「S]キーでやり直すより、手の認識にしたいと思います。

**感想**

一年生の時に中西研究会でProcessingとArduinoを触ったことありますが、この授業は色々考えさせてもらいました。例えば、画像処理と言えば、普通にPhotoshopなどでフィルターをかけるくらいと思ったが、二回目の授業にマトリックスのような処理ができることが教えていただけました。それからどんどんレベルアップに行って、以前学部生には難しいと思った認識ができました。そして、憧れたスーパーマリオにもその仕組みで簡単なゲームができました。土曜日の授業に出席できなくて悔しいが、筧先生の授業を受けて良かったです。

**参考文章**

アイデアの構想のコード

<https://forum.processing.org/two/discussion/5554/super-mario-game-help>

オープンソースデータ

<https://github.com/Aravindlivewire/Opencv/tree/master/haarcascade>

PythonのPILでイメージ処理

<http://www.coderholic.com/batch-image-processing-with-python/>

私が前に完成したゲームからのアイデア<https://play.google.com/store/apps/details?id=com.KarmicThief.test>

HIPIで大数の画像をダウンロード

<https://www.oreilly.com/ideas/how-to-analyze-100-million-images-for-624>

Haar-Cascadeをトレーニングする

<http://coding-robin.de/2013/07/22/train-your-own-opencv-haar-classifier.html>

モーション分析とオブジェクトのトレッキング

<http://docs.opencv.org/2.4/modules/video/doc/motion_analysis_and_object_tracking.html>

シンプルなブースト・カスケードでラピッドオブジェクト認識

<http://wearables.cc.gatech.edu/paper_of_week/viola01rapid.pdf>