# 基于行动装置之眼动指令侦测

**赵贞豪**
2019110070
计算机系
zhaozh19@mails.tsinghua.edu.cn

**宋政钦**
2016012395
计算机系
szq16@mails.tsinghua.edu.cn

**张可真**
2017011432
计算机系
zkz17@mails.tsinghua.edu.cn

## Abstract

正确且精确的眼动侦测在计算机视觉领域中有着重要的地位，在本项目中，我们将针对现实场景应用，实现了五方位的眼动侦测模型。算法包括以下流程：首先，我们对图像进行candidate region proposal，找出眼睛所在位置的可能点并将其送入第一层CNN模型，分别标出左眼、右眼或无等标签。再者，我们将这些带标签的图像送入第二层神经网络，进行五个方位的辨识。

## 1 引言

随计算机科技的发展，现代的行动装置也拥有强大的算力，能支援各式各样的运算，也使得人们使用脸部特征作为输入变得普及，眼动侦测因而成为重要的研究领域之一。本项目致力于眼动侦测对于现实场景的应用，主要想解决缺乏额外设备(红外线侦测)下，以摄像头作为输入设备进行解读。预设场景是做一款眼动控制的手机桌布，人们可以就由眼球移动的方向来控制行动装置。

然而，眼动侦测是一个相当困难的项目，原因来自于眼睛出现在图像中不固定的位置与角度，也因为现实场景中变化多端的光照以及对人脸的遮挡使得此任务难度加剧。在这篇报告当中，我们将提出如何因应多变的现实场景并有效率的侦测人们眼睛动态。以下我们将先由「相关工作」作为引子，讨论现有的相关文献以及其中的算法，并在「方法」中提出一套有效的算法。而在「实验」的章节，我们将藉由数据证明此方法的有效性，最后在「结论」、「讨论」中概括本篇重点并提出进一步的设想。

## 2 相关工作

目前有关侦测眼睛注视方向的方法大致分为两类：第一种为Model-based 的侦测方式，着重于对现实场景的信号的分析。像是[1]、[2]当中提出以接收瞳孔发出的红外线反射去判断眼睛位置，然而这样的方式缺点是需要有多个高解析度的摄像头，以及在有稳定光照的环境下才能完成。而Appearance-based的方式则可直接以图像作为输入，藉由图像直接判别眼睛动作，使得这样的方式在深度学习非常火红的今日，成为了研究的一大热点。但主要的缺点仍然是需要大量的data，而且对于model没有辨识过的图像仍然会有精度上的落差[3]。Appearance-based的模型建立在图像中pixel values跟眼睛注视位置的连结(映射)，可想而知这样的算法很容易会受到光线、角度的变化而使得映射后有极大的变化。这也成为Appearance-based模型在Real-time的实际应用场景之中有许多弱点——非常不稳定且很容易侦测失败。

较早期的机器学习对眼动侦测的算法[4]，以衔接式的Haar wavelet以及SVM去分类。而近期随着神经网络的兴起，大部分对影像的处理都交给了CNN去处理，由于CNN可以快速的抽取到图像特征，因此像是[5]衔接了两层CNN去确认眼睛位置并预测瞳孔位置、[10]使用Viola-Jones algorithm的算法找到眼睛位置后并对其进行九方位的分类，在精度上有显著的成果。然而在有脸部遮挡、光线变化的现实场景中，模型的精度无法达到如实验数据中的完美。因此，在此基础上，我们将提出以下方法，加强模型侦测的Rubustness，解决Appearance-based模型在现实场景中的效果不彰的问题。

## 3 方法

我们要侦测与解读人们的眼球移动的五种方向，分别是往上、往下、往左、往右、不动。输入即是行动装置摄像头捕捉到的画面，可以视为一张张连续、密切相关的人脸图像输入；而输出是上述提到的五种指令。也就是说，$X$=$\{x|x_1, x_2, x_3, \ldots, x_i$ is a 3D tensor with size Width*Height*3 $\}$是藉由设备采集到的相片，$A \in \{a_1, a_2, a_3, a_4, a_5\}$是预测使用者执行特定动作的机率($P_{pred}$)，而我们将取最高的机率作为模型输出($a_{pred}$)。

$$P_{pred} = \{p_1, p_2, p_3, p_4, p_5\} = P(A|x_i) = \{P(a_1|x_i), P(a_2|x_i), P(a_3|x_i), P(a_4|x_i), P(a_5|x_i)\}$$
$$a_{pred} = argmax(P_{pred})$$

算法的workflow如Figure 1所展示，主要分为两个区块：一是从输入图像当中提取眼睛的位置，二是由这些撷取的图像进行分类任务。以撷取眼睛图像来说，我们使用R-CNN[5]的方式，先对输入图像做Region proposal。对于这些撷取过后的图像我们将其送进第一阶段的卷积神经网路并分类为三类：左眼与右眼或无。此后将单一图像中带label的左、右眼图像送入第二层神经网路，最后分类成五种动作。在接下来的几个单元中，我们将更详细的讨论这些功能如何实现。

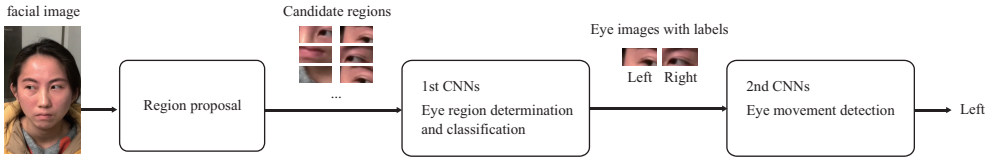


Figure 1: 整体算法workflow图

### 3.1 Region Proposal

若直接对整个脸部图像(e.g. 540*960)做卷积，会需要非常庞大的计算资源，在当今一般的行动装置是难以支持的做到Real-time的。[10]提出使用Viola and Jones' face detector迅速找到脸部特侦，然而由于光线变化、脸部遮挡等原因，会使此算法精度有大幅度的降低。因此，我们不使用以人脸特侦做region proposal的方式，而是参考[7]当中提到的算法进行改良。我们发现在眼睑边缘时常会是周遭最黑的部分之一，因此先对图像做了边缘提取的运算，再去寻找局部极值，并搜集此些点作为region proposal的参照值。具体实现的部分，首先选用三个不同标准差的Gaussian kernel $G_{x,y,\sigma}$ 对脸部图像$I_{x,y}$做卷积(模糊化)，再对此三图像进行Sobel kernel $S$的卷积提取图像边缘，若在图像中的pixel为邻近5x5的三张图像中最大值则取中心点为该点加常数修正。整体运算可以如下表示：

$$G_{x,y,\sigma} = \frac{1}{2\pi\sigma^2} e^{\frac{-(x^2+y^2)}{2\sigma^2}}$$
$$S_x = \begin{pmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{pmatrix} * (I_{x,y} * G_{x,y,\sigma}), \quad S_y = \begin{pmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{pmatrix} * (I_{x,y} * G_{x,y,\sigma}), \quad S = \sqrt{S_x + S_y}$$

经过以上操作后，我们选出N个最大的局部极值点作为feature points，而N的取值主要是要根据眼睛是否能被正确撷取而决定。这里面临的取舍在于精度与速度，N取越大，之后的第一层CNN要进行的运算量就越大，但N取太小则可能造成正确位置的遗漏，我们将在以下章节详细用数据客观筛选N值。

### 3.2 Eye Region Determination and Classification

在选取可能为眼睛的图像的位置(candidate points)后，我们还需要将它们归类，标上左眼、右眼的标签。这部分我们利用卷积神经网路(CNN)去分类，由于现实场景中人物所距离摄像头的位置可远可近，眼睛大小变化多，因此我们依据该cadidate point撷取三种不同出入，分别为12x24 、18x36、24x48。

这三套卷积神经网路的架如下所示。首先，我们第一层卷积filter size=3x3, stride=2，共用了64个filter。第二层卷积filter size=3x3, stride=2，共用了64个filter。第一层卷积filter size=3x3, stride=2，共用了64个filter。在每层中间我们都加入了batch normalization层，以及ReLU层作为activation，以及加入max pooling缩减图像尺寸。最后使用softmax计算各输出的机率，并使用cross-entropy函数作为loss function。

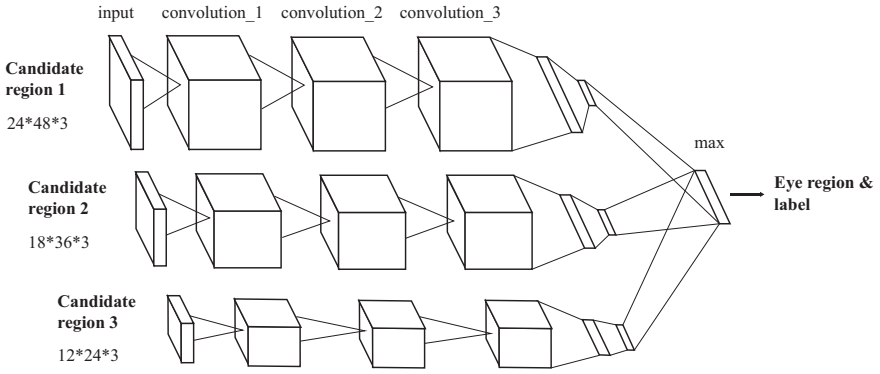


Figure 2: 第一层神经网路架构

### 3.3 Eye Movement Classification

我们从第一层的卷积神经网路当中得到了带标签的眼睛图像，会依据输出的机率值挑出最高的左、右眼图像各一张输入第二层卷积神经网路做眼睛动态的分类。

网络架构总结于下图Figure 3。我们首先对个别输入图像进行三层的卷积，最后将左右眼卷积的输出结果连接一全连接层，映射到五种动作的分类。首先，我们第一层卷积filter size=3x3, stride=2，共用了64个filter。第二层卷积filter size=3x3, stride=2，共用了64个filter。第一层卷积filter size=3x3, stride=2，共用了64个filter。在每层中间我们都加入了batch normalization层，以及ReLU层作为activation，以及加入max pooling缩减图像尺寸。最后使用softmax计算各输出的机率，并使用cross-entropy函数作为loss function。

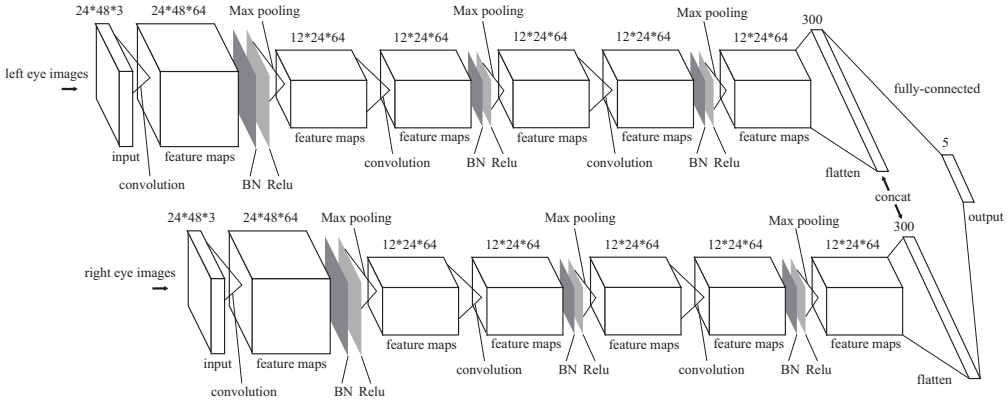


Figure 3: 第二层神经网路架构

# 4 训练方式

## 4.1 Feature Points Selection

在Region Proposal的章节当中提到了选择N值需面临的trade-off。在降低运算量的同时我们会牺牲一部份的精度。因此我们测试了以下几组的N值(N=50,100,150,200,250)并记录「至少有一只眼睛」被选择到的比例。在N=200的时候，94.19%的frame成功找到了眼睛位置，在实验数据中为膝点，也代表了此点以后增加大量计算却无精度提升，因此，我们取其作为feature points的数量依据。

| N | 50 | 100 | 150 | **200** | 250 |
|---|---|---|---|---|---|
| Percentage | 55.69 | 75.30 | 93.70 | **94.19** | 94.19 |

Table 1: 至少一只眼睛被涵盖之机率

## 4.2 Dataset

搜集dataset的主要原因来自于现今许多脸部图像的训练集如BioID[8]、GI4E[9]的图像较为一致，在光线、角度、遮挡上没有太大的变化，并不符合我们所针对的应用场景。因此我们从真实场景使用去搜集dataset，我们由33人总共搜集了249个视频。每个视频约1至3秒，主要是日常以手机等行动装置拍摄，包含不同角度、光线条件、若干脸部遮挡。我们将其分为三类：训练集、测试集(known)、测试集(unknown)，后两者的差别在于测试集(known)当中的人接有出现于训练集当中，而测试集(unkown)当中的人则是与训练集当中无重复的，以模拟现实场景之侦测。

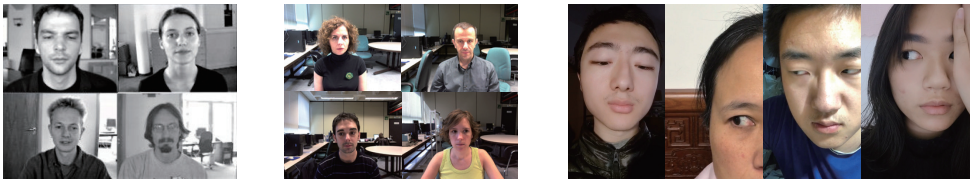

Figure 4: 样例: (left) from BioID (middle) from GI4E (right) form our dataset

对于第一阶段CNN的dataset我们自各视频当中取偶数格frame(i.e. 0,2,4,6...)，并对其进行上述region proposal的算法，得到cadidate regions，从中挑出5至10张图像加入训练集(优先选左右眼图像)，而label($L_i$)的判定则是跟当初标记的位置取欧氏距离的平方，左方眼睛距记为$D_{left,i}(x,y)$、与右方眼睛距离$D_{right,i}(x,y)$则表示关系如下。而对于第二阶段CNN的dataset我们同样取偶数格frame，藉由标注的眼睛点撷取图像，并人工标注上、下、左、右、不动等标签。

$$D_{left,i}(x,y) = d^2_{left,i,x} + d^2_{left,i,y}$$
$$D_{right,i}(x,y) = d^2_{right,i,x} + d^2_{right,i,y}$$
$$L_i = \begin{cases} 1 & \text{if } D_{left,i}(x,y) < 150 \\ 2 & \text{if } D_{right,i}(x,y) < 150 \\ 0 & \text{otherwise} \end{cases}$$

## 4.3 Data Augmentation

为了扩充dataset的大小以及增强模型的rubustness，我们应用了一些data augmentation的技巧。首先是角度的旋转，对于短视频中的每一个frame，分别旋转-8,-5,5,8再撷取，若撷取到部分图像外的区域则以补零处理，借此增强模型对人脸角度的rubostness；再者是距离的偏移，除了在我们标注的眼睛位置撷取作为输入之外，我们分别对上下左右各偏移5单位的距离再次撷取作为训练，增强模型对小幅度偏移的rubostness；最后则是对图像色相(H)、饱

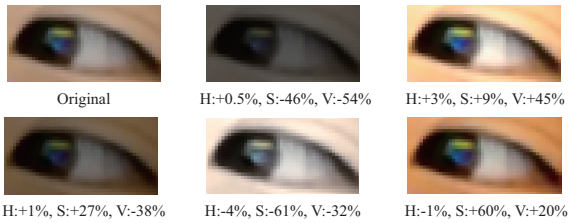


Figure 5: 数据增强效果(HSV调节)

和度(S)、彩度(V)的调整，我们由原图调整了五组不同HSV值，将这些经过转换的图加入训练集，增强模型对于不同光线、环境的rubostness。经过这些转换后，在第一阶段的traing data达到了493902张图像；第二阶段则达到344100张图像。

## 5 实验

实验在Intel(R) Xeon(R) CPU E5-2682 v4 @ 2.50GHz的计算机搭载NVIDIA GeForce GTX 1080 GPU上执行，这套算法使用python 3.6.8、tensorflow 1.12.0实现，以下将分开讨论第一层、第二层网络的性能，两层网络使用我们搜集的dataset进行验证与比较。

### 5.1 Eye Detection and Classification

Table 2比较了我们的模型以及[6]在测试集(known)、测试集(unknown)上的精度。可以看到我们的算法在精度上是有所些微提升，然而更显著的是Data Augmentation的效果，在精度上是十分有帮助的。

| Model | Data Augmentation | Test Accuracy (known) | Test Accuracy (unknown) | Running Time(ms) |
|---|---|---|---|---|
| Model from [6] | Yes | 0.7567 | 0.7615 | 3.91924 |
| Model from [6] | No | 0.6852 | 0.7032 | - |
| **Our Model** | **Yes** | **0.8399** | **0.8220** | **4.28803** |
| Our Model | No | 0.7606 | 0.7730 | - |

Table 2: 比较我们的model以及[6]当中的model在有无数据增强条件下之精度、运行时间

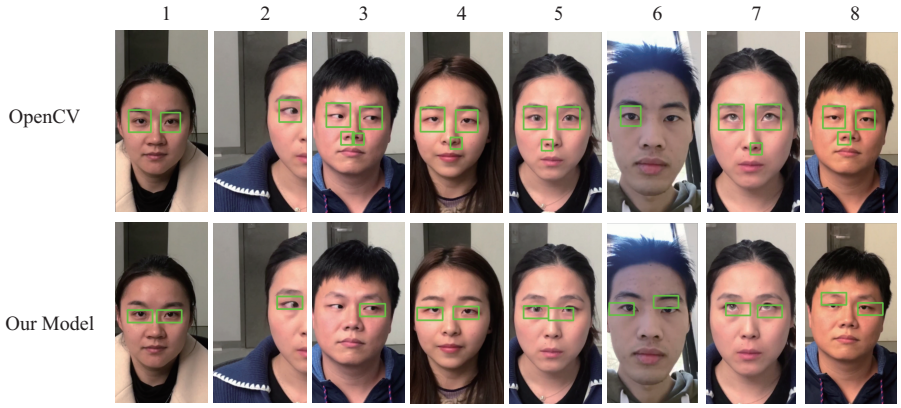


Figure 6: 两者预测结果1、2之预测结果正确；3我们的模型找到一只眼睛，而OpenCV则过度预测(包含鼻孔)；4我们的模型找到两只眼睛，而OpenCV则也过度预测(包含鼻孔)；5、6、7、8我们的model因位置不准而侦测失败，而OpenCV过度预测(包含鼻孔)或过少预测(6只有左眼)

Figure 6展示了一些范例，我们比较与现今被广泛使用的OpenCV的Haar-cascade Detection算法在精度的表现。两者都伴随着精度不理想的问题。我们使模型预测左右眼位置，若能包含且判断预测与ground truth的欧式距离是否小于150且个数正确，则计为成功预测。在测试集(unknown)测试的精度，两者相当左眼皆为0.2939，右眼为0.2727(OpenCV)及0.2797(Our)。

### 5.2 Eye Movement Classification

Table 4展示我们在精度上稍微领先[10]当中所提出的模型。这部分我们一样是在我们所搜集的dataset上比较，主要分为known users和unknown users。我们可以看到有无data augmentation的悬殊差别：model经过Augmented data的训练后，精度提升超过30%，如此说明训练集的数量以及当中光线变化都是重要的训练指标之一。

| Model | Data Augmentation | Test Accuracy (known) | Test Accuracy (unknown) | Running Time(ms) |
|---|---|---|---|---|
| Model from [10] | Yes | 0.8693 | 0.7782 | 2.13550 |
| Model from [10] | No | 0.5561 | 0.5315 | - |
| **Our Model** | **Yes** | **0.8952** | **0.7858** | **2.20166** |
| Our Model | No | 0.5538 | 0.4645 | - |

Table 3: 比较我们的model以及[10]当中的model在有无数据增强条件下之精度、运行时间

### 5.3 Evaluation

最终，我们将两层架构相接，针对测试集(known)进行测试，分类正确的图像在1423帧中占652帧，正确率0.4582；测试集(unknown)分类正确的图像在1619帧中占570帧，正确率0.3521。

我们实验在两层网络中间加入尺寸调节，使得第一层预测的位置能稍作调整避免误判。具体作法是将图像先放大截图20*30、28*42、36*54，再缩回原大小即12*24、18*36、24*48。经过调节后测试集(known)达到接近60%正确率，也突破40%。

| Test set | Resize | Correct frames | All frames | Accuracy |
|---|---|---|---|---|
| known | Yes | 845 | 1423 | 0.5938 |
| known | No | 652 | 1423 | 0.4582 |
| unknown | Yes | 682 | 1619 | 0.4212 |
| unknown | No | 570 | 1619 | 0.3521 |

Table 4: 比较最终精度在有无尺寸调节下的精度

## 6 结论

本项目做出了以下贡献：(1)制作一套完整的dataset，搜集来自33位不同年纪、性别受试者的249个短视频，并且其场景包含多样的光照环境以及若干的脸部遮挡。整理为训练集、测试集(known)、测试集(unknown)，并可供后续相关研究使用。(2)建立了一套以衔接式CNN去分类眼睛动态的算法流程，当中包括对[7]提出region proposal的方式以及对[10]当中眼睛方位分类器进行算法修正，让模型在进行寻找眼睛位置、预测方位时能以提升精度。(3)设计了一系列的data Augmentation方式去增强模型的rubostness，包含角度变化、位置变化、HSV变化，同时以实验数据印证了其效果。

组内分工：赵贞豪：(1)想法提案及paper search、(2)处理dataset工具搭建(用于标label及档案转换)、(3)手工标注一半训练集、一半测试集(known)、全部测试集(unknown)、(4)搭建文中

提及的两层神经网络(将想法以python实现)、(5)第一阶段神经网路的训练(fine-tune)、(6)第二阶段神经网路的训练(fine-tune)、(7)期中与期末报告撰写(文中所有实验数据生成)。

张可真：(1)搜集一半训练集、测试集(known)、(2)手工标注一半训练集、一半测试集(known)、(3)第二阶段神经网路的训练(fine-tune)。

宋政钦：(1)搜集一半训练集、测试集(known)、全部测试集(unknown)。

## 参考文献


[1] P. Majaranta and A. Bulling, “Eye tracking and eye-based human–computer interaction,” in Advances in physiological computing. Springer, 2014, pp. 39–65

[2] C. H. Morimoto, A. Amir, and M. Flickner, “Detecting eye position and gaze from a single camera and 2 light sources,” in Pattern Recognition, 2002. Proceedings. 16th International Conference on, vol. 4. IEEE, 2002, pp. 314–317

[3] X. Zhang, Y. Sugano, M. Fritz, and A. Bulling, “Appearance-based gaze estimation in the wild,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 4511–4520.

[4] S. Chen and C. Liu, “Eye detection using discriminatory Haar features and a new efficient SVM,” Image and Vision Computing, vol. 33, pp. 68–77, 2015.

[5] Girshick, R., Donahue, J., Darrell, T., Malik, J.: Rich feature hierarchies for accurate object detection and semantic segmentation. In: Conference on Computer Vision and Pattern Recognition, pp. 580–587. IEEE (2014)

[6] Bin Li & Hong Fu (2017) Real Time Eye Detector with Cascaded Convolutional Neural Networks Applied Computational Intelligence and Soft Computing, vol. 2018, Article ID 1439312, 8 pages, 2018. https://doi.org/10.1155/2018/1439312.

[7] J. Lemley. & A. Kar, A. Drimbarean & P. Corcoran (2018) Efficient CNN Implementation for Eye-Gaze Estimation on Low-Power/Low-Quality Consumer Imaging Systems (arXiv:1806.10890)

[8] O. Jesorsky, K. J. Kirchberg, and R. W. Frischholz, “Robust face detection using the hausdorff distance,” in Audio- and Video-Based Biometric Person Authentication: Third International Conference, AVBPA 2001 Halmstad, Sweden, June 6–8, 2001 Proceedings, J. Bigun and F. Smeraldi, Eds., vol. 2091 of Lecture Notes in Computer Science, pp. 90–95, Springer, Berlin, Germany, 2001

[9] A. Villanueva, V. Ponz, L. Sesma-Sanchez, M. Ariz, S. Porta, and R. Cabeza, “Hybrid method based on topography for robust detection of iris center and eye corners,” ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM), vol. 9, no. 4, article 25, 2013.

[10] C. Zhang, R. Yao, and J. Cai, “Efficient eye typing with 9-direction gaze estimation,” Multimedia Tools and Applications, Nov 2017. [Online]. Available: https://doi.org/10.1007/s11042-017-5426-y